UNIVERSIDADE ESTADUAL DE CAMPINAS FACULDADE DE ENGENHARIA MECÂNICA

Relatório Final Trabalho de Conclusão de Curso

Comparação e implementação dos métodos de otimização topológica estrutural evolucionária BESO e baseada em programação matemática SIMP

> Autor: Vinicius Lourenço da Silva Daldon Orientador: Prof. Dr. Renato Pavanello

Campinas, agosto de 2020

UNIVERSIDADE ESTADUAL DE CAMPINAS FACULDADE DE ENGENHARIA MECÂNICA

Relatório Final Trabalho de Conclusão de Curso

Comparação e implementação dos métodos de otimização topológica estrutural evolucionária BESO e baseada em programação matemática SIMP

Autor: Vinicius Lourenço da Silva Daldon Orientador: Prof. Dr. Renato Pavanello

Curso: Engenharia Mecânica

Trabalho de Conclusão de Curso apresentado à Comissão de Graduação da Faculdade de Engenharia Mecânica, como requisito para a obtenção do título de Engenheiro Mecânico.

Campinas, 2020 S.P. – Brasil

Dedicatória:

Dedico este trabalho à minha Mãe Luzinete, ao meu Pai Wilson (in memoriam) e ao meu Irmão Marcelo.

Agradecimentos

Este trabalho não poderia ser terminado sem a ajuda de diversas pessoas às quais presto minha homenagem:

A Deus, pela existência e tudo o que possuo.

À minha família, Luzinete, Wilson (in memoriam) e Marcelo, por todo o amor e suporte em minha vida.

Ao Prof. Renato Pavanello, pela orientação, sugestão de tema, apoio e amizade.

Aos professores da Faculdade de Engenharia Mecânica da UNICAMP, por acreditarem no poder de transformação da educação e se empenharem ao máximo na formação dos alunos.

À UNICAMP, por fornecer condições de ensino e permanência estudantil excepcionais.

Aos meus amigos e colegas de curso, por todo apoio e por tornarem as experiências durante a graduação mais leves.

Índice

	Resumo	1
	Lista de Figuras	3
	Lista de Tabelas	4
	Nomenclatura	5
	Abreviações	6
Capítulo 1	Introdução	7
Capítulo 2	Revisão Bibliográfica	10
2.1.	Otimização Estrutural e Método dos Elementos Finitos	10
2.2.	A Aplicação do Método dos Elementos Finitos	11
2.3.	A Formulação dos Elementos Finitos	12
2.4.	Bi-directional Evolutionary Structural Optimization (BESO)	14
2.5.	Solid Isotropic Material with Penalization Method (SIMP)	18
Capítulo 3	Modelo Adotado e Equações constitutivas	21
3.1.	Introdução	21
3.2.	O Estado Geral de Deformação	21
3.3.	O Conceito de Tensão e o Estado de Tensão	26
3.4.	O Modelo Constitutivo	30
3.5	Discretização de Um Sistema Contínuo e Equação de Movimento	31
Capítulo 4	Aplicação do Método dos Elementos Finitos	36
4.1.	Introdução	36
4.2.	Formulação Isoparamétrica do Elemento Plano Quadrilateral	37
Capítulo 5	Otimização Estrutural	44
5.1	Introdução	44
5.2	Bi-directional Evolutionary Structural Optimization (BESO)	44
5.3	Solid Isotropic Material with Penalization Method (SIMP)	50
Capítulo 6	Procedimento Experimental	54
6.1.	Sistemática de Otimização Topológica Estrutural	54
Capítulo 7	Resultados e Discussões	55
7.1.	Aplicação do Método BESO	55
7.1.1.	O Problema da Estrutura de Michell	55

7.1.1.1.	Análise Inicial	55
7.1.1.2.	Variando o Parâmetro ER	60
7.1.2.	A Estrutura em Forma de Mão-Francesa	66
7.1.2.1.	Análise Inicial	66
7.1.2.2.	Variando o Parâmetro ARmax	73
7.2.	Aplicação do Método SIMP	77
7.2.1.	O Problema da Estrutura de Michell	77
7.2.1.1.	Análise Inicial	77
7.2.1.2.	Variando o Parâmetro de Penalidade p	81
7.2.2.	A Estrutura em Forma de Mão-Francesa	82
7.2.2.1.	Análise Inicial	82
7.2.2.2.	Variando a Dimensão do Filtro rmin	85
Capítulo 8	Conclusões	88
	Referências Bibliográficas	91
	Anexos	94

Resumo

DALDON, Vinicius Lourenço da Silva, Comparação e implementação dos métodos de otimização topológica estrutural evolucionária BESO e baseada em programação matemática SIMP, Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, Trabalho de Conclusão de Curso, Relatório Final, (2020), 121 pp.

O presente trabalho tem como objetivo comparar os métodos de otimização topológica de estruturas Bidirectional Evolutionary Structural Optimization (BESO) e Solid Isotropic Material with Penalization method (SIMP), utilizando uma abordagem baseada na aplicação de método dos elementos finitos (MEF) para o estudo de problemas estruturais selecionados. Para isso, serão adaptados dois códigos de Matlab® bastante utilizados pelo Departamento de Mecânica Computacional da Faculdade de Engenharia Mecânica da UNICAMP.

Inicialmente, o projeto de trabalho de graduação será contextualizado, descrevendo os problemas de estruturas mecânicas, discorrendo sobre as diferentes classificações de otimização. Um enfoque maior sobre a otimização topológica, objeto de estudo do trabalho, será dado.

Na sequência, serão destacados as principais aplicações e características da análise de problemas de estruturas mecânica sob a abordagem do método dos elementos finitos, indicando as suas vantagens e desvantagens. A importância desse método reside na possibilidade de se obter soluções de maneira simplificada e de grande precisão, quando aliado a ferramentas computacionais.

A metodologia de análise de problemas estruturais utilizando o método dos elementos finitos será descrita, apresentando os seus fundamentos. Além disso, as questões referentes à escolha do tipo de elemento finito e função de interpolação serão abordadas.

Em seguida, o método BESO será apresentado. A técnica consiste em, a partir de uma malha de elementos finitos, eliminar os elementos que não atinjam um critério estabelecido para a resolução do problema. Ao contrário do Evolutionary Structural Optimization (ESO), o BESO possibilita a inclusão de novos elementos, caso essa adição direcione a estrutura à topologia mais eficiente. A influência da variação de parâmetros de otimização BESO sobre os exemplos de problemas selecionados será discutida.

Finalmente, o método SIMP será discutido, apresentando as suas principais características. O algoritmo é fundamentado na possibilidade de existirem densidades intermediárias para os elementos finitos adotados na discretização da estrutura em estudo, permitindo uma análise matemática mais robusta. Mais uma vez, serão selecionados alguns parâmetros de otimização inerentes a esse método e a variação deles terá a sua influência analisada sobre os resultados nos exemplos escolhidos.

Palavras Chave: Otimização Topológica, Métodos dos Elementos Finitos, BESO, SIMP.

Lista de Figuras

Figura 2.1. Procedimento geral para a formulação de elementos finitos.	14
Figura 2.2. Fluxograma de aplicação do método BESO.	16
Figura 2.3. Fluxograma de aplicação do método SIMP.	19
Figura 3.1. Corpo qualquer submetido a esforços externos e vínculos.	22
Figura 3.2. Plano xy para o elemento infinitesimal adotado.	22
Figura 3.3. Esforços internos em uma seção transversal de um corpo qualquer.	26
Figura 3.4. Componentes do estado de tensão em um ponto O.	27
Figura 3.5. Tetraedro formado em torno do ponto O.	28
Figura 4.1. Elemento plano quadrilateral de formulação isoparamétrica.	37
Figura 5.1. Um exemplo de tabuleiro de xadrez resultante no método ESO.	46
Figura 5.2. Nós contemplados pelo domínio Ω i, aplicado no esquema de filtro	48
utilizado para o elemento i.	
Figura 7.1. Domínio inicial para a Estrutura de Michell.	56
Figura 7.2. Numeração de elementos e eixos no domínio da Estrutura de Michell.	56
Figura 7.3. Estrutura de Michell otimizada minimizando o compliance.	57
Figura 7.4. Mapa de cores da tensão equivalente de Von misses na Estrutura de	58
Michell otimizada	
Figura 7.5. Compliance e fração de volume em função da iteração do programa	59
Figura 7.6. Estrutura de Michell otimizada por Hemp (1973).	60
Figura 7.7. Topologias obtidas para ER = 10%, 8% e 6%.	61
Figura 7.8. Topologias obtidas para ER = 4%, 2% e 1%.	62
Figura 7.9. Topologias obtidas para ER = 0,50%, 0,25% e 0,20%.	62
Figura 7.10. Topologias obtidas para ER = 0,15%, 0,10% e 0,05%.	62
Figura 7.11. Gráfico de dispersão Compliance X ER.	63
Figura 7.12. Gráfico de dispersão Iteração X ER.	63
Figura 7.13. Gráfico de dispersão Compliance X ER (excluindo os pontos em que	64
a estrutura não converge).	
Figura 7.14. Gráfico de dispersão Iteração X ER (excluindo os pontos em que a	65
estrutura não converge).	
Figura 7.15. Domínio inicial para a Estrutura em forma de mão-francesa.	67
Figura 7.16. Estrutura em Forma de Mão-Francesa otimizada minimizando o	68
compliance.	

Figura 7.17. Mapa de cores da tensão equivalente de Von misses na Estrutura	69
em Forma de Mão-Francesa otimizada.	
Figura 7.18. Compliance e fração de volume em função da iteração do programa.	70
Figura 7.19. Mão-Francesa obtida por otimização ESO com restrição volumétrica	71
de 35%.	
Figura 7.20. Mão-Francesa obtida por otimização ESO com restrição volumétrica	72
de 35%, utilizando malha hexagonal.	
Figura 7.21. Topologias obtidas para ARmax = 5% e 4%.	74
Figura 7.22. Topologias obtidas para ARmax = 3% e 2%.	74
Figura 7.23. Topologias obtidas para ARmax = 1% e 0,5%.	75
Figura 7.24. Topologias obtidas para ARmax = 0,1% e 0,05%.	75
Figura 7.25. Gráfico de dispersão Compliance X ARmax.	76
Figura 7.26. Gráfico de dispersão Iteração X ARmax.	76
Figura 7.27. Numeração de elementos e eixos no domínio da Estrutura de	78
Michell para o código SIMP.	
Figura 7.28. Estrutura de Michell otimizada minimizando o compliance utilizando	79
o método SIMP.	
Figura 7.29. Compliance e fração de volume em função da iteração do programa.	80
Figura 7.30. Topologia obtida para a Estrutura de Michell com $p = 1 e 2$.	81
Figura 7.31. Topologia obtida para a Estrutura de Michell com $p = 3 e 4$.	81
Figura 7.32. Estrutura em forma de Mão-Francesa otimizada minimizando o	83
compliance utilizando o método SIMP.	
Figura 7.33. Compliance e fração de volume em função da iteração do programa.	84
Figura 7.34. Topologias obtidas para rmin = 0,1, 0,5 e 1,0.	86
Figura 7.35. Topologias obtidas para rmin = 1,5, 2,0 e 2,5.	86
Figura 7.36. Topologias obtidas para rmin = 3,0, 3,5 e 4,0.	86
Lista de Tabelas	
Tabela 7.1. Resumo de resultados para a otimização inicial da Estrutura de	60
Michell.	

Tabela 7.2. Resumo de resultados para a otimização inicial da Estrutura de61Michell variando o parâmetro ER.

Tabela 7.3. Valores sem convergência para a otimização da Estrutura de Michell64

variando o parâmetro ER.

Tabela 7.4. Resumo de resultados para a otimização inicial da Estrutura em	73
Forma de Mão-Francesa.	
Tabela 7.5. Resumo de resultados para a otimização inicial da Estrutura em	73
Forma de Mão-Francesa variando o parâmetro ARmax.	
Tabela 7.6. Resumo de resultados para a otimização inicial da Estrutura de	80
Michell utilizando o método SIMP.	
Tabela 7.7. Resumo de resultados para a otimização inicial da Estrutura de	82
Michell utilizando o método SIMP e variando p.	
Tabela 7.8. Resumo de resultados para a otimização inicial da Estrutura em	84
forma de Mão-Francesa utilizando o método SIMP.	
Tabela 7.9. Resumo de resultados para a otimização inicial da Estrutura em	85
forma de Mão-Francesa utilizando o método SIMP e variando rmin.	

Nomenclatura

Vetores e Matrizes

[<i>K</i>]	Matriz de rigidez global
[<i>C</i>]	Matriz de flexibilidade
{ <i>U</i> }	Vetor de deslocamentos nodais
{ <i>F</i> }	Vetor global de carregamento externo
$\{T_t\}$	Vetor com as componentes de tensão do ponto O nas direções x,
	y e z segundo o plano p
[<i>T</i> _o]	Tensor de tensões de Cauchy com as componentes de tensão no
	ponto P
{ P _p }	Vetor com os cossenos diretores da normal ao plano p
{σ}	Vetor com as componentes de tensão
[<i>D</i>]	Matriz de elasticidade
{ε}	Vetor com as componentes de deformação
{ <i>u</i> }	Vetor com os deslocamentos do sistema continuo
[<i>N</i>]	Matriz de funções de forma
[<i>B</i>]	Matriz de deformação
{a}	Vetor de aceleração

- *{***\vhetay}** Vetor de forças de superfície
- **{X}** Vetor de forças de corpo
- *{δu}* Vetor com os deslocamentos virtuais
- **{δ***U***}** Vetor com os deslocamentos virtuais nodais
- **{δÜ}** Vetor com as acelerações nodais
- *{P}* Vetor com as cargas concentradas
- [J] Matriz Jacobiana
- [G] Matriz mapeamento geométrico
- [P] Matriz de derivadas parciais das funções de interpolação
- *{δ}* Vetor de deslocamentos nodais do elemento
- [*k*^(e)] Matriz rigidez do elemento

Abreviações

- MEF Método dos elementos finitos;
- **ESO** Evolutionary structural optimization;
- **BESO** Bidirectional evolutionary structural optimization;
- **SIMP** Solid isotropic material with penalization;

Capítulo 1

Introdução

A concepção de estruturas cada vez mais eficientes em suas aplicações é um processo inerente ao desenvolvimento da engenharia. Isso se justifica pela necessidade de se obter produtos com alto grau de qualidade e segurança, associado a custos de produção menores, motivados pela crescente competitividade natural dos setores produtivos. Nesse contexto, está inserido o estudo de técnicas de otimização estrutural, que visam aplicar metodologias capazes de direcionar a solução de um problema para o resultado mais adequado com base em objetivos previamente definidos. Portanto, o interesse pelo estudo da otimização estrutural é justificado pela condição das técnicas de viabilizar o projeto de estruturas com desempenho mais eficiente de forma sistemática.

A modelagem de estruturas mecânicas é acompanhada muitas vezes pela necessidade de analisar equações diferenciais, que regem os fenômenos físicos envolvidos, com elevado grau de complexidade, tornando a obtenção de soluções analíticas inviável. Dada essa condição, o método dos elementos finitos (MEF) surge como uma alternativa capaz de contornar essa dificuldade. Segundo Bittencourt (2010), o MEF pode ser considerado como um método numérico utilizado para a solução de Problemas de Valor de Contorno relacionando equações diferencias parciais e ordinárias com suas respectivas condições de contorno. A essência do MEF consiste em dividir o domínio de um problema em vários subdomínios de geometria e tratamento mais simples, denominados elementos finitos. Esse procedimento é identificado como a discretização de um domínio. Dessa forma, sobrepondo a análise realizada em cada um dos elementos gerados, é possível obter uma solução aproximada do problema inicial mais complexo.

A aplicação do MEF na análise de problemas estruturais mecânicos encontra na capacidade de processamento dos computadores um fator limitante. Tendo em vista que a divisão de um domínio em elementos finitos resulta em uma série de problemas menores, é importante que os sistemas resultantes desse processo sejam solucionados em computadores com elevada capacidade de processamento. Caso contrário, a aplicação do método se torna limitada e com resultados pouco satisfatórios. Conforme Bittencourt (2010), uma série de programas computacionais baseada na análise por elementos finitos

passou a ser implementada ao final da década de 1960 e começo da década de 1970. Contudo, o escopo de aplicação era bastante restrito, possibilitando o estudo de problemas mais simples, motivado pelo baixo desempenho dos computadores da época e alto custo computacional. Por sua vez, o crescente desenvolvimento na indústria eletrônica tem possibilitado o crescimento exponencial da aplicação do MEF para os mais diferentes tipos de problemas, tornando possível solucionar complexos que não eram tratáveis, segundo Xie e Steven (1997).

A otimização estrutural consiste no processo matemático que possui como objetivo determinar configurações de projetos estruturais capazes de realizar o melhor desempenho para parâmetros selecionados, respeitando as cargas e condições de contorno impostas ao problema. Desse modo, é possível determinar configurações que ofereçam ao projeto a maior rigidez, a menor massa ou o menor custo possível, por exemplo. Grosso modo, o objetivo da otimização estrutural é definir uma maneira sistemática de se encontrar configurações capazes de atingir os requisitos do projeto de forma mais econômica e segura possível.

Os métodos de otimização estrutural são classificados em três grandes categorias: otimização dimensional, otimização geométrica e otimização topológica. Os métodos de otimização dimensional visam determinar as dimensões dos componentes de uma estrutura, determinando as seções transversais que minimizem o volume de uma viga, por exemplo. Por sua vez, a otimização geométrica objetiva investigar os contornos mais adequados para que os requisitos sejam cumpridos da forma mais eficiente possível, indicando as posições ideais para a aplicação dos nós na construção de uma torre de transmissão de energia, por exemplo. Finalmente, a otimização topológica, objetivo de estudo do presente trabalho, busca definir o padrão de conectividade dos elementos que compõe o projeto desenvolvido, de forma a atingir o desempenho ótimo com base em critérios definidos. Simplificadamente, a otimização estrutural indica onde, em um domínio definido, deve existir material ou a ausência dele de forma a se obter uma estrutura com desempenho eficiente.

Entre os métodos de otimização topológica, o Bi-directional Evolutionary Structural Optimization (BESO) recebe importante destaque. Assim como o método Evolutionary Structural Optimization (ESO), a técnica BESO se baseia no princípio de remoção gradual do material ineficiente na estrutura, resultando em uma configuração mais próxima do estado com melhor desempenho. Contudo, segundo Huang e Xie (2010), a diferença entre

os dois métodos reside na possibilidade de ocorrer tanto acréscimo quanto remoção de material no processo de otimização.

Outra técnica de otimização topológica muito estudada é a Solid Isotropic Material with Penalization (SIMP). Esse método é aplicado através da atribuição de densidades de material intermediárias para os elementos utilizados na análise do problema, possibilitando o contorno de algumas dificuldades encontradas em outros métodos de otimização, através da aplicação de uma matemática mais robusta.

Diante do grande número de técnicas de otimização presentes na área de otimização topológica, é interessante estabelecer um comparativo entre dois métodos bastante utilizados (BESO e SIMP). Dessa forma, é possível investigar as vantagens e desvantagens de cada método, possibilitando entender melhor os seus escopos satisfatórios de aplicação.

Dessa forma, o presente trabalho visa realizar uma pesquisa bibliográfica com o intuito de proporcionar entendimento sobre as duas técnicas de otimização estrutural, possibilitando adaptar e utilizar códigos de Matlab® para implementar as otimizações em exemplos de problemas estruturais selecionados e analisar os resultados obtidos.

Capítulo 2

Revisão Bibliográfica

2.1. Otimização Estrutural e Método dos Elementos Finitos

Xie e Steven (1997) definem a otimização estrutural como a junção da matemática, engenharia, ciência e tecnologia atuando com o intuito de alcançar o melhor desempenho para uma estrutura, sendo que esta pode desempenhar o papel de uma ponte ou um veículo espacial, por exemplo. Na aplicação do processo de otimização, ainda segundo os autores, todas os aspectos do ambiente em que a estrutura se insere devem ser tratados como variáveis envolvidas no processo. Huang e Xie (2010) defendem que o interesse pelo estudo da otimização estrutural é justificado por uma série de fatores, como o impacto ambiental, a escassez de recursos materiais e a competição tecnológica. Tudo isso, atua como uma mola propulsora na busca por projetos cada vez mais eficientes.

Huang e Xie (2010) apresentam as três categorias em que os processos de otimização estrutural costumam ser divididos. São eles: dimensional, geométrica e topológica. Segundo os autores, a otimização dimensional tem por objetivo encontrar a configuração ideal para uma estrutura alterando as variáveis relacionadas as dimensões, como espessuras de chapas e seções transversais de armações ou treliças. Os autores defendem que essa é a abordagem mais fácil e rápida para melhorar o desempenho de uma estrutura. Por sua vez, a otimização geométrica possui aplicação comum em estruturas contínuas, alterando os contornos predefinidos para obter um melhor desempenho. Finalmente, a otimização topológica atua em estruturas discretas e continuas de maneiras diferentes. Os autores apontam que em estruturas discretas, como quadros e treliças, o processo age indicando a ordem espacial e conectividade ideais de barras. Já para estruturas contínuas, o processo se baseia em determinar as melhores localidades e geometrias para a introdução de cavidades no domínio analisado.

As técnicas de otimização estrutural, por serem fundamentadas em análises de mecânica dos sólidos, encontraram na complexidade algébrica das equações diferencias que fundamentam os fenômenos físicos um fator limitante para o seu desenvolvimento. Nesse sentido, é interessante aplicar alguma abordagem capaz de contornar essa dificuldade. Reddy (2006) defende que a aplicação de métodos numéricos muitas vezes é a única saída possível para a solução desse tipo de problemas. Além disso, essa aplicação

é bastante recomendada graças ao grande desenvolvimento computacional alcançado nos últimos anos, possibilitando obter soluções muito eficazes de maneira rápida. Dentre os principais métodos numéricos, o método dos elementos finitos recebe grande destaque no campo de análise estrutural. Reddy (2006) descreve o método dos elementos finitos como o processo de visualizar um domínio como um conjunto de subdomínios, nos quais são aplicadas aproximações das equações governantes dos fenômenos analisados. Ainda segundo o autor, é mais simples tratar funções complexas como uma série de funções polinomiais, decorrente da aplicação do método.

O conjunto de características estudadas, tornam o método dos elementos finitos como recurso essencial para as ferramentas de otimização topológicas. Dessa forma, os dois métodos de otimização topológica estudados no presente trabalho fundamentam suas aplicações em abordagens baseadas no método dos elementos finitos.

2.2. A Aplicação do Método dos Elementos Finitos

É comum recorrer ao método dos elementos finitos diante de problemas de engenharia que apresentam geometrias complexas e não linearidades dos materiais analisados. Essas condições podem demandar a resolução de equações diferenciais com soluções de obtenção complexa ou impossíveis. Dessa forma, os métodos numéricos são apresentados como importantes alternativas.

Bittencourt (2010) defende que o cerne dos métodos dos elementos finitos consiste em dividir o domínio de análise (o corpo em estudo, no caso da análise estática) em subdomínios (elementos finitos), obtendo a solução aproximada do problema. Para isso, são considerados pontos específicos do domínio (nós). A junção dos elementos finitos e nós constitui a malha de elementos finitos.

Os elementos finitos passam então a representar uma fração do corpo rígido em análise. Portanto, é necessário que cada uma dessas partes represente componentes da rigidez do sistema. Esse papel é desempenhado pela formulação adequada da matriz de rigidez dos elementos. As matrizes são funções das propriedades materiais e geométricas da estrutura, além de serem caracterizadas pelos nós que delimitam os elementos finitos correspondentes.

Através da sobreposição das matrizes de rigidez da totalidade dos elementos que compõe a malha do corpo estudado, é possível obter a matriz de rigidez global [K]. Utilizando uma análise nodal, deve ser representado um vetor global de carregamento

externo {*F*}, posicionando os esforços e reações aplicados em cada um dos graus de liberdade dos nós utilizados na malha. Então, associando o vetor de deslocamentos nodais {*U*}, que correspondem as incógnitas do problema, é obtido o sistema de equações representado pela equação 2.1:

Ainda segundo Bittencourt (2010), impondo as condições de contorno do problema, invertendo a matriz de rigidez [K] obtêm-se a matriz de flexibilidade [C], permitindo obter os deslocamentos nodais da análise quando multiplicando os dois termos da equação 2.1, conforme a equação 2.2:

A solução numérica do sistema gerado possibilita obter os deslocamentos nodais e, consequentemente, as deformações e tensões no domínio do problema. A partir dos fundamentos apresentados, Bittencourt (2010) salienta que o MEF demanda duas aproximações. A primeira reside no fato de que as geometrias das estruturas são representadas pelas faces e superfícies dos elementos finitos situados nos contornos dos corpos. Sendo assim, geometrias complexas necessitam de malhas mais refinadas, isto é, com mais elementos. A segunda aproximação decorre do fato de que as grandezas de interesse na análise são obtidas por uma abordagem nodal. Dessa forma, para os demais pontos das estruturas, é necessário utilizar funções de interpolação, definidas a partir das grandezas obtidas para os nós. É esperado que a solução aproximada obtida pelo MEF se aproxime cada vez mais da solução real conforme se refina a malha de análise.

2.3. A Formulação dos Elementos Finitos

Existe uma vasta biblioteca de elementos finitos disponíveis para a execução de análises por MEF, cada um com suas características e comportamentos. Dessa forma, um tipo de elemento pode ser adequado para a aplicação em um certo tipo de problema enquanto em outras aplicações esse mesmo elemento pode resultar em uma análise pouco precisa. O comportamento interno dos elementos, isto é, dos pontos que não pertencem aos nós das malhas, é governado pela função de interpolação escolhida, como apresenta Alves Filho (2007). O autor destaca a importância da escolha da função interpolação que, de acordo com as suas limitações, determinam o comportamento dos deslocamentos dos pontos que não são nós e, consequentemente, influenciam na precisão do resultado aproximado obtido pela análise utilizando MEF. Alves Filho (2007) aponta que funções polinomiais correspondem a uma das classes de funções de interpolação mais difundidas, considerando a sua fácil manipulação.

O desenvolvimento da formulação, conforme proposto por Alves Filho (2007), reside em, inicialmente, selecionar a função de interpolação mais adequada para o tipo de elementos adotado. Em seguida, é possível calcular o valor dos coeficientes para cada elemento finito relacionando os deslocamentos de seus pontos com os deslocamentos nodais associados a esse elemento. Na sequência, é possível obter as deformações internas dos elementos, com os coeficientes da função de interpolação determinados. Então, determina-se as forças internas em cada elemento e também as suas tensões. Finalmente, é possível obter a matriz de rigidez de cada elemento, efetuando um balanço entre o trabalho interno e externo das forças. Resumidamente, o procedimento é ilustrado pela seguinte figura:

DETERMINE OS COEFICIENTES DESCONHECIDOS DA FUNÇÃO
A partir dos Deslocamentos Nodais são calculados os Deslocamentos dentro do Elemento por meio da Função de Forma $[N(x)]$
$\delta_{\text{DENTRO DO ELEMENTO}} = N(x) \cdot \delta_{\text{NODAL}}$
DETERMINE AS DEFORMAÇÕES DENTRO DO ELEMENTO
A partir dos Deslocamentos Nodais são calculadas as Deformações dentro do Elemento por intermédio da Matriz Deslocamento-Deformação [B(x)]
$\epsilon_{\text{DENTRO DO ELEMENTO}} = [B(x)] \cdot \delta_{\text{NODAL}}$
DETERMINE AS FORÇAS INTERNAS NO ELEMENTO E AS TENSÕES
$\sigma_{\text{DENTRO DO ELEMENTO}} = [S] \cdot \delta_{\text{NODAL}}$
DETERMINE A MATRIZ DE RIGIDEZ DO ELEMENTO, APLICANDO A CONDIÇÃO DE EQUIVALÊNCIA DE ENERGIA, IGUALANDO OS TRABALHOS EXTERNO E INTERNO, RESULTANDO:
$\{k\}^{e} = \int_{vol} [B]^{T} \cdot [D] \cdot [B] \cdot dVol$



2.4. Bi-directional Evolutionary Structural Optimization (BESO)

Como apresentado por Huang e Xie (2010), as primeiras pesquisas utilizando o BESO foram realizadas por Yang et al. (1999), visando obter otimizações para a rigidez de estruturas. Esses estudos eram embasados na premissa de estimar números de

sensibilidade para os elementos vazios de um domínio através da extrapolação linear do campo de deslocamento resultante da análise por métodos numéricos. Na etapa seguinte, os elementos sólidos que apresentavam números de sensibilidade mais baixos eram removidos, enquanto os elementos nulos com número de sensibilidade mais altos eram transformados em elementos sólidos. Os elementos que seriam removidos ou inseridos após cada iteração do algoritmo eram consequência dos parâmetros taxa de rejeição (TR) e taxa de inclusão (TI), respectivamente.

Huang e Xie (2007) definem o BESO como uma extensão do método Evolutionary Structural Optimization (ESO). O ESO, segundo os autores, se baseia no conceito de remoção gradual de material ineficiente na estrutura, fazendo com que ela caminhe para um estado ótimo. Por sua vez, o BESO permite que materiais eficientes também sejam adicionados à estrutura, possibilitando a adição e remoção de material. Os autores ainda elencam vantagens e desvantagens da aplicação dos métodos ESO/BESO. As características mais favoráveis da aplicação dos métodos são a facilidade em suas implementações e a possibilidade de se obter um perfil claro de topologia, sem elementos "cinzas", facilitando a fabricação da estrutura gerada. Por sua vez, Huang e Xie (2007) mencionam o fato de que a aplicação do ESO e o BESO muitas vezes podem resultar em soluções não convergentes. Ou seja, a solução pode caminhar para um resultado cada vez pior com relação à função objetivo, caso o procedimento continue sem um critério de parada bem definido.

Huang e Xie (2010) estabelecem o método de aplicação da otimização BESO da seguinte maneira:

- Discretizar o domínio inicial utilizando malhas de elementos finitos e atribuir valores iniciais para as propriedades (0 ou 1) analisadas, com o objetivo de estabelecer uma configuração inicial.
- Executar a análise sob a abordagem dos elementos finitos e calcular o número de sensibilidade de cada elemento.
- Calcular a média dos números de sensibilidade dos elementos, utilizando o valor da iteração corrente e o valor da iteração anterior. Adotar como sensibilidade do elemento, a média entre os dois valores.
- 4) Determinar o volume buscado para a próxima iteração.
- 5) Remover ou adicionar os elementos que atinjam os critérios estabelecidos pelo método.

 Repetir os passos de 2 a 5, até que o volume esperado e o critério de convergência sejam atingidos.

Huang e Xie (2010) apresentam o seguinte fluxograma com o objetivo de esquematizar o funcionamento do método BESO:



Figura 2.2. Fluxograma de aplicação do método BESO. Fonte: Huang e Xie (2010).

Diante da versatilidade e simplicidade de aplicação do método BESO, uma série de trabalhos que propõe a sua aplicação para otimização de estruturas foi publicada. Além disso, o algoritmo do método é passível de modificações, tornando a sua aplicação mais eficiente para o contexto de cada problema analisado. Pereira, Bono e Bono (2019) lançam mão do método BESO para a otimização de um sistema de contravento para edifícios altos. Além disso, os autores estabelecem um comparativo com o método SIMP, destacando que, embora o SIMP resulte em estruturas bem mais flexíveis, o método BESO se destaca pela sua maior simplicidade e agilidade de processamento. Somado a isso, os autores concluem que a otimização topológica decorrente da aplicação do método BESO possibilita uma interpretação muito mais simples dos resultados, considerando que o método não admite elementos cinzas, isto é, elementos com densidades intermediárias.

Explorando a versatilidade do método, He e Liu (2008) aplicaram o método BESO visando otimizar a topologia de estruturas utilizadas para a dissipação de calor. Na construção do problema, o critério de rejeição utilizado pelos autores foi a sensibilidade de resistência ao calor, indicando que elementos com baixa sensibilidade de resistência térmica são ineficientes e devem ser retirados da estrutura. Por sua vez, o gradiente de temperatura dos elementos foi considerado como critério de adição, o que indica que elementos com alto gradiente de temperatura são considerados como elementos eficientes e devem ser adicionados, com o objetivo de minimizar a máxima temperatura na estrutura. Os resultados encontrados pelos autores mais uma vez corroboram a eficiência do método, demonstrando os resultados obtidos. Além disso, mais uma vez é citada a facilidade na manufatura proporcionada nos projetos desenvolvidos pela otimização BESO frente ao método SIMP.

Rodríguez (2015) fez uso da polivalência do método BESO analisando estruturas bidimensionais sob uma abordagem multi-física e multi-objetivo, levando em consideração comportamentos mecânicos e térmicos. Nesse estudo, o objetivo é minimizar a capacidade térmica e a flexibilidade média das estruturas nos problemas selecionados. Os resultados obtidos pelo autor permitiram validar a aplicação do método na otimização de estruturas de problemas de natureza termo-mecânica, considerando que as formas resultantes são coerentes com topologias existentes na literatura.

Objetivando contornar alguns dos principais problemas do método BESO, Huang e Xie (2007) propuseram um algoritmo adaptado da otimização. A dependência de malha

é uma das dificuldades enfrentadas durante a execução do procedimento. Ela consiste na constatação de que, conforme são utilizadas malhas mais refinadas em volumes estruturais constantes, a introdução de mais orifícios na topologia aumenta a eficiência do projeto. Esse comportamento é enxergado como uma instabilidade numérica. Além disso, os autores citam o problema de o método resultar em soluções não convergentes. Para contornar esses problemas, os autores aplicaram um filtro independente de malha e melhoraram a sensibilidade dos elementos com base em informações históricas de outros trabalhos. Com isso, os resultados alcançados demostram o sucesso do algoritmo em superar as dificuldades inerentes ao método BESO.

2.5. Solid Isotropic Material with Penalization Method (SIMP)

As dificuldades encontradas pelos pesquisadores nos métodos de otimização topológica utilizando variáveis binárias, resultaram na procura por métodos capazes de tratar o problema da otimização de forma continua. O método Solid Isotropic Material with Penalization Method (SIMP), por tratar as densidades de material de forma continua, representa uma alternativa para os métodos de otimização topológica. Segundo Huang e Xie (2010), os conceitos originais para o método foram propostos por Bendsøe (1989). Segundo os autores, o método SIMP pode ser entendido como uma abordagem baseada em assumir que cada elemento finito é formado por um material isotrópico com densidade variável. Dessa forma, as variáveis analisadas na otimização são as densidades relativas dos elementos. Na construção do método, é realizada uma interpolação por lei de potencia que possui como objetivo penalizar as densidades intermediárias obtidas para os elementos, induzindo a obtenção de uma solução próxima ao esquema 0-1. Huang e Xie (2010) ainda defendem que o método SIMP já demonstrou a sua eficiência através da aplicação em uma ampla gama de exemplos, além de ter sido bem aceito graças a eficiência computacional e simplicidade conceitual de seu algoritmo.

Aremu et al. (1999) definem um fluxograma que apresenta o funcionamento de um algoritmo baseado no método SIMP:



Figura 2.3. Fluxograma de aplicação do método SIMP. Fonte: Aremu et al. (2010).

Muitos trabalhos explorando a aplicação do método SIMP foram publicados, envolvendo a otimização topológica de diferentes tipos estruturas. Aremu et al. (1999) utilizaram o método para gerar topologias otimizadas para uma placa metálica retangular plana e uma estrutura aeroespacial tridimensional, que podem ser fabricadas por manufatura aditiva. Nos resultados obtidos, os autores discorrem sobre a necessidade de se adaptar os algoritmos para o projeto de manufatura aditiva, considerando o surgimento de furos que, grosso modo, representa uma estrutura otimizada. Contudo, dada a complexidade de fabricação de estruturas desse tipo, o surgimento desse resultado é inibido pela penalidade imposta no método SIMP. Os autores defendem que essa dificuldade é facilmente contornada pela versatilidade inerente à fabricação aditiva, possibilitando fabricar os modelos mais complexos obtidos por um algoritmo adaptado.

Wang, Lin e Xia (2013) aplicaram o método SIMP para a construção de uma topologia otimizada para o projeto estrutural de uma asa dobrável. Segundo os autores, o projeto desse tipo de componente é concebido, de modo geral, pela experiência do

engenheiro projetista. Contudo, esse método requer muito tempo e experiência do responsável pelo projeto. Os resultados obtidos, segundo os autores, indicam que a otimização topológica pelo método SIMP representa um método de projeto mais eficiente e assertivo.

Huang e Xie (2010) estabeleceram um comparativo para a otimização topológica entre o método BESO e o método SIMP. Essa comparação, segundo os autores, permite melhores conclusões quando separada em dois casos: utilizando filtro independente de malha e não utilizando um filtro independente de malha. Na ausência do filtro, as topologias alcançadas diferem bastante, sendo praticamente incomparáveis, segundo os autores. Nesse cenário, o BESO alcança o estado otimizado em muito menos iterações do algoritmo. Por sua vez, o estado obtido pela otimização SIMP apresenta um compliance médio muito maior. Inserindo o filtro independente de malha, as topologias obtidas são muito próximas, para os dois métodos. Contudo, o método SIMP gerar uma topologia com regiões de densidade intermediária (cinza). Além disso, os números de interações necessárias para o estado otimizado dessa vez são muito próximos, enquanto o compliance médio obtido para o método SIMP é maior, mais uma vez.

Capítulo 3

Modelo Adotado e Equações constitutivas

3.1. Introdução

Na presente etapa do trabalho, será apresentado o desenvolvimento teórico utilizado como base para a aplicação do método dos elementos finitos. Dessa forma, serão apresentadas as definições de grandezas físicas pertinentes e as equações constitutivas utilizadas em problemas de mecânica dos sólidos.

O estudo realizado consiste em um problema de elasticidade plana, utilizando uma abordagem baseada em algumas hipóteses. O meio material analisado é sólido, homogêneo, isotrópico e com comportamento linear elástico, isto é, regido pela Lei de Hooke. Além disso, será considerado o caso bidimensional para o estado plano de tensão.

3.2. O Estado Geral de Deformação

Para que um corpo se deforme, é necessário que exista um número adequado de restrições capazes de impedir o movimento desse corpo, conforme Bittencourt (2010). Supondo um regime de pequenos deslocamentos, aqueles realizados por partículas do corpo em estudo nos eixos coordenados x, $y \in z$ dependem das posições dessas partículas e podem ser descritos, respectivamente por u, $v \in w$:

$$u = u(x, y, z)$$

 $v = v(x, y, z)$ Equação 3.1
 $w = w(x, y, z)$

Como consequência dos pequenos deslocamentos assumidos, as componentes de deslocamento serão muito pequenas e variam continuamente ao longo de seus domínios, isto é, o volume do corpo.



Figura 3.1. Corpo qualquer submetido a esforços externos e vínculos. Fonte: Mesquita (2016).

Selecionando um elemento infinitesimal que compõe o corpo sólido, é possível descrever matematicamente as deformações normais ϵ e distorções (ou deformações angulares) γ impostas aos pontos desse elemento. Mesquita (2016) menciona a possibilidade de linearizar as distorções, no sentido de expansões de séries de Taylor, quando a abordagem se dá de forma a considerar pequenos deslocamentos e deformações. Somado ao fato de o problema estudado ser um caso de elasticidade plana, a figura 3.2, que apresenta o plano *xy*, é capaz representar de forma eficiente as configurações normal e deformada do elemento selecionado:



Figura 3.2. Plano *xy* para o elemento infinitesimal adotado. Fonte: Mesquita (2016) (adaptado).

A deformação específica no ponto A é definida pela seguinte expressão:

$$\epsilon_{xx} = \lim_{\Delta x \to 0} \frac{\Delta x' - \Delta x}{\Delta x}$$
 Equação 3.2

Da figura 3.2, é possível concluir que:

$$\Delta x' = \Delta x + u + \Delta u - u = \Delta x + \Delta u$$

Por sua vez, a variação do deslocamento pode ser descrita por:

$$\Delta u = \frac{\partial u}{\partial x} \Delta x$$

Finalmente, substituindo as duas expressões em (3.2), é possível concluir que:

$$\epsilon_{xx} = \frac{\partial u}{\partial x}$$

Analogamente, para a direção y, é possível concluir que:

$$\epsilon_{yy} = \frac{\partial v}{\partial y}$$

Consequentemente, equações de deformações normais em ponto do elemento podem ser representadas por:

$$\epsilon_{xxx} = \frac{\partial u}{\partial x}$$

$$\epsilon_{yy} = \frac{\partial v}{\partial y}$$

Equação 3.3

$$\epsilon_{zz} = \frac{\partial w}{\partial z}$$

Retornando à figura 3.2, é possível concluir que a deformação angular no plano *xy* é dada pela soma dos ângulos γ_1 e γ_2 . Portanto:

$$\gamma_{xy} = \gamma_1 + \gamma_2$$
 Equação 3.4

Por sua vez, os diferenciais de *u* e *v* em relação *y* e *x* são dados por, respectivamente:

$$du_{y} = \frac{\partial u}{\partial y} \Delta y$$
$$dv_{x} = \frac{\partial v}{\partial x} \Delta x$$

Analisando a geometria apresentada na figura 3.2, é possível determinar as tangentes dos ângulos γ_1 e γ_2 :

$$tan \gamma_{1} = \frac{\frac{\partial v}{\partial x} \Delta x}{\Delta x'} = \frac{\frac{\partial v}{\partial x} \Delta x}{\Delta x + \frac{\partial u}{\partial x} \Delta x} = \frac{\frac{\partial v}{\partial x}}{1 + \epsilon_{xx}}$$
Equação 3.5
$$tan \gamma_{2} = \frac{\frac{\partial u}{\partial y} \Delta y}{\Delta y'} = \frac{\frac{\partial u}{\partial y} \Delta y}{\Delta y + \frac{\partial v}{\partial y} \Delta y} = \frac{\frac{\partial u}{\partial y}}{1 + \epsilon_{yy}}$$
Equação 3.6

Considerando que as deformações são pequenas, é possível aproximar os valores das tangentes de γ_1 e γ_2 aos próprios ângulos. Além disso, as deformações específicas ϵ_{xx} e ϵ_{yy} são muito pequenas quando comparadas à unidade. Dessa forma, é possível concluir que:

$$\gamma_1 = \frac{\partial v}{\partial x}$$
 Equação 3.7

24

$$\gamma_2 = \frac{\partial u}{\partial y}$$
 Equação 3.8

Voltando para a equação 3.4 e aplicando as duas equações anteriores, é possível obter a deformação angular total como:

$$\gamma_{xy} = \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}$$
 Equação 3.9

Dessa forma, analogamente, é possível determinar as deformações angulares em todos os planos da seguinte forma:

$$\gamma_{xy} = \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}$$

$$\gamma_{xz} = \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z}$$
Equação 3.10
$$\gamma_{yz} = \frac{\partial w}{\partial y} + \frac{\partial v}{\partial z}$$

Finalmente, é possível escrever as equações 3.3 e 3.10 de forma matricial, relacionando as deformações com os deslocamentos de um ponto:

$$\begin{cases} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{zz} \\ \gamma_{xy} \\ \gamma_{xz} \\ \gamma_{yz} \end{cases} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \end{bmatrix}$$

Equação 3.11

3.3. O Conceito de Tensão e o Estado de Tensão

Supondo que um corpo qualquer de um material que atenda as hipóteses consideradas para o presente estudo (perfeitamente elástico, homogêneo e isotrópico) seja submetido à ação de um conjunto de forças externas, é possível verificar que ele se deforma até que a força resultante desse conjunto seja equilibrada pelas forças internas existentes no interior do corpo, segundo Bittencourt (2010). Por isso, no estudo de mecânica dos sólidos, é importante caracterizar essas forças internas. Com esse objetivo, o corpo é seccionado ao longo de um plano qualquer *mm*, resultando na figura 3.3, que retrata uma das partes do corpo original:



Figura 3.3. Esforços internos em uma seção transversal de um corpo qualquer. Fonte: Bittencourt (2010).

As forças internas atuantes na parte isolada do corpo têm o papel de equilibrar as forças externas aplicadas, organizadas em uma distribuição não uniforme e continua. Dada essa natureza, a grandeza das forças internas é atribuída ao conceito de tensão, isto é, força interna resultante por unidade de área da superfície analisada. Dessa forma, selecionando o ponto *O* pertencente a superfície resultante do corte *mm*, é possível determinar a tensão atuante nele através da seguinte expressão, que relaciona a resultante das forças internas atuantes δF e a área elementar δA :

$$\overrightarrow{T_t} = \lim_{\delta A \to 0} \frac{\delta \overrightarrow{P}}{\delta A}$$

Equação 3.12

De acordo com a expressão, o vetor tensão T_t atua na mesma direção da resultante de forças internas *P*. Além disso, conforme mostrado na figura 3.3, é possível decompor o vetor em componentes normal (tensão normal) e tangente a superfície (tensão de cisalhamento). Em último caso, também representado na figura 3.3, é possível associar um sistema de coordenadas à análise e decompor o vetor tensão T_t em três componentes: σ_{zz} (tensão normal atuante no plano *z* e na direção de *z*), σ_{zx} (Tensão de cisalhamento atuante no plano *z* na direção *x*) e σ_{zy} (Tensão de cisalhamento atuante no plano *z* na direção *y*).

Finalmente, no caso geral de solicitação, é gerado uma tensão que pode ser decomposta em nove componentes, sendo distribuídas com três em cada um dos planos. A seguinte figura ilustra a situação:





Bittencourt (2010) reforça que o teorema de Cauchy mostra que as componentes de tensão de cisalhamento atuantes em planos perpendiculares entre si são iguais. Isso pode ser demonstrado através de um balanço de momentos atuantes. Portanto:

$$au_{xy} = au_{yx}$$

 $au_{xz} = au_{zx}$ Equação 3.13
 $au_{yz} = au_{zy}$

Retornando ao corpo genérico que atende a todas as hipóteses do problema e esteja submetido ao conjunto de forças externas, segundo Bittencourt (2010), é possível verificar que tensão que atua em um ponto não depende apenas do ponto selecionado, mas também do plano que secciona o corpo. Dessa forma, para um mesmo ponto, é possível obter tantos vetores de tensão quantos são os planos que podem conter o ponto analisado. Contudo, ainda segundo Bittencourt (2010), é possível caracterizar o estado de tensão de um ponto determinando apenas as componentes de tensão com relação aos três planos coordenados. Com esse objetivo, conhecendo as componentes de tensão em um ponto *O* com relação aos eixos coordenados, é representado um tetraedro formado pelos três planos ortogonais e um plano *p*, caracterizado pelo vetor normal *P*. A situação é representada na figura 3.5:



Figura 3.5. Tetraedro formado em torno do ponto O. Fonte: Bittencourt (2010).

Os cossenos diretores do vetor P em relação aos eixos coordenados x, y e z são, respectivamente, I, m e n. Portanto, é possível projetar a área da face BCD, A, nos três planos coordenados, obtendo as áreas das outras faces do tetraedro. Dessa forma, as

áreas das faces OCD, OBC e OBD são dadas, respectivamente por *Al*, *Am* e *An*. Então, realizando um balanço de forças atuantes no tetraedro, é possível obter:

$$\Sigma F_{x} = 0; \quad \tilde{T}_{x}A - \tilde{\sigma}_{xx}Al - \tilde{\tau}_{xy}Am - \tilde{\tau}_{xz}An - \chi_{x}\frac{1}{3}Ah = 0$$

$$\Sigma F_{y} = 0; \quad \tilde{T}_{y}A - \tilde{\tau}_{yx}Al - \tilde{\sigma}_{yy}Am - \tilde{\tau}_{yz}An - \chi_{y}\frac{1}{3}Ah = 0$$

Equação 3.14

$$\Sigma F_z = 0: \quad \tilde{T}_z A - \tilde{\tau}_{zx} A l - \tilde{\tau}_{zy} A m - \tilde{\sigma}_{zz} A n - \chi_z \frac{1}{3} A h = 0$$

Na equação 3.14, T_x , T_y e T_z são as tensões médias na face BCD e χ_x , χ_x e χ_x são as forças de volume aplicadas no tetraedro. Quando a altura *h* do tetraedro tendo a zero, as tensões médias existentes nas faces do sólido tendem às componentes de tensão no ponto *O*. Aplicando esse procedimento e considerando a equação 3.13, o balanço de forças pode ser apresentado da seguinte maneira:

$$T_{x} = \sigma_{xx}l + \tau_{xy}m + \tau_{xz}n$$

$$T_{y} = \tau_{xy}l + \sigma_{yy}m + \tau_{yz}n$$
Equação 3.15
$$T_{z} = \tau_{xz}l + \tau_{yz}m + \sigma_{zz}n$$

Organizando de forma matricial, as equações podem ser expressas por:

$$\begin{cases} T_x \\ T_y \\ T_z \end{cases} = \begin{bmatrix} \sigma_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{xy} & \sigma_{yy} & \tau_{yz} \\ \tau_{xz} & \tau_{yz} & \sigma_{zz} \end{bmatrix} \begin{pmatrix} l \\ m \\ n \end{pmatrix}$$
Equação 3.16

Alternativamente:

$$\{T_t\} = [T_0^{\sigma}]\{P_p\}$$
Equação 3.17

Onde T_t representa o vetor com as componentes de tensão do ponto O nas direções x, y e z segundo o plano p, T_0 representa o tensor de tensões de Cauchy com as

componentes de tensão no ponto *P* (com relação a *x*, *y* e *z*) e P_p consiste no vetor com os cossenos diretores da normal ao plano *p*.

3.4. O Modelo Constitutivo

Uma das hipóteses adotadas para o desenvolvimento do presente trabalho é que os comportamentos das estruturas analisadas estão de acordo com o conceito de estado plano de tensões. De acordo com Alves Filho (2007), esse tipo de comportamento ocorre principalmente em chapas finas, estruturas em que a espessura é bem menor que o comprimento e largura, e quando as forças atuantes nessas estruturas ocorrem paralelamente aos seus planos. Isso tudo resultado em deslocamentos também no próprio plano da estrutura, possibilitando representar o estado de tensão por apenas duas dimensões. Representando uma chapa no plano *xy*, por exemplo, as tensões atuantes são representadas apenas por σ_{xx} , σ_{yy} e τ_{xy} .

Bittencourt (2010) apresenta a lei de Hooke como a equação que relaciona as componentes de deformação e tensão que ocorrem nos pontos de uma estrutura. Descartando a influência de uma distribuição de temperatura, que não é objetivo do presente estudo, as relações são representadas por:

$$\epsilon_{xx} = \frac{1}{E} \left[\sigma_{xx} - \nu \left(\sigma_{yy} + \sigma_{zz} \right) \right]$$

$$\epsilon_{yy} = \frac{1}{E} \left[\sigma_{yy} - \nu \left(\sigma_{xx} + \sigma_{zz} \right) \right]$$

$$\gamma_{xy} = \frac{\tau_{xy}}{G}$$

Equação 3.18

Na equação 3.18, v representa o coeficiente de Poisson, E corresponde ao módulo de elasticidade longitudinal do material e G equivale ao módulo de elasticidade transversal desse material. Nesse sentido, é possível relacionar G e E da seguinte forma:

$$G = \frac{E}{2(1+\nu)}$$
 Equação 3.19
Então, aplicando a equação 3.19 na equação 3.18 e considerando que os problemas são de estado plano de tensões, é possível apresentar a lei de Hooke de forma matricial:

$$\begin{cases} \epsilon_{xx} \\ \epsilon_{yy} \\ \gamma_{xy} \end{cases} = \frac{1}{E} \begin{bmatrix} 1 & -\nu & 0 \\ -\nu & 1 & 0 \\ 0 & 0 & 2(1+\nu) \end{bmatrix} \begin{cases} \sigma_{xx} \\ \sigma_{yy} \\ \tau_{xy} \end{cases}$$
 Equação 3.20

Alternativamente, é possível apresentar as componentes de tensão em função das componentes de deformação da seguinte forma:

$$\begin{cases} \sigma_{xx} \\ \sigma_{yy} \\ \tau_{xy} \end{cases} = \frac{E}{1 - \nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1 - \nu}{2} \end{bmatrix} \begin{cases} \epsilon_{xx} \\ \epsilon_{yy} \\ \gamma_{xy} \end{cases}$$
 Equação 3.21

Compactamente, a equação 3.21 pode ser escrita como:

$$\{\sigma\} = [D]\{\varepsilon\}$$
 Equação 3.22

Onde D é chamada de matriz de elasticidade.

3.5. Discretização de Um Sistema Contínuo e Equação de Movimento

A solução de problemas estruturais muitas vezes é acompanhada de uma série de dificuldades que inviabiliza a aplicação direta das equações de elasticidade. Esses entraves são decorrentes das geometrias dos problemas, das condições de contorno, dos carregamentos aplicados e propriedades do material, por exemplo. Diante desse cenário, que inviabiliza a obtenção da solução analítica do problema, uma das alternativas é buscar a solução em pontos específicos no domínio do problema (nós) e, de posse desses valores, expandir os resultados para as outras regiões através de funções de interpolação. Esse tipo de sistema é denominado discreto. Nessa abordagem, as estruturas são compostas por elementos discretos, determinados pelas conexões dos nós (ou pontos nodais) que, por sua vez, devem respeitar compatibilidades de deslocamentos e forças, por exemplo. Dessa forma, é possível definir vários subdomínios para a estrutura

convergir para a solução real do problema. Bittencourt (2010) apresenta um desenvolvimento para a equação de movimento de uma estrutura discretizada, que será utilizada como base para o presente trabalho.

É possível relacionar os deslocamentos do sistema continuo $\{u\}$ e discreto $\{U\}$ da seguinte maneira:

$$\{u\} = [N]\{U\}$$
Equação 3.23

Onde [*N*] representa uma matriz compostas por funções N_{ij} (funções de forma) que fazem o mapeamento entre os modelos discreto e contínuo.

No modelo discreto, as deformações podem ser descritas substituindo a equação 3.23 na equação 3.11, resultando em:

$$\{\epsilon\} = [B]\{U\}$$
 Equação 3.24

Onde [*B*] corresponde a matriz de deformação, que contém as derivadas parciais das funções de forma adotadas. Analogamente, é possível descrever as tensões no modelo discreto substituindo a equação 3.24 na equação 3.22, resultando em:

$$\{\sigma\} = [D][B]\{U\}$$
 Equação 3.25

Desse modo, como descrito, obtendo os valores de deslocamentos nodais, é possível obter os valores de tensões e deformações.

Quando um corpo sólido e continuo de massa *m* é submetido a um sistema de forças externas {*F*}, de acordo com a Segunda Lei de Newton, ele deve se movimentar com uma aceleração {*a*}. Ou seja:

$$\{F\} = m\{a\} = m\{\ddot{u}\}$$
Equação 3.26

Onde $\{a\}$ representa o vetor de aceleração, que equivale a segunda derivada do vetor de deslocamentos $\{u\}$ dos pontos do corpo analisado.

Escrevendo a equação 3.26 de forma alternativa:

$$\{F\} - m\{\ddot{u}\} = 0 \qquad \qquad \text{Equação 3.27}$$

O segundo termo da equação, $-m{\ddot{u}}$, corresponde a força de inércia do corpo. Apresentando a equação no formato de 3.27, o chamado Princípio de D'Alembert, é possível aplicar o Princípio do Trabalho Virtual para os problemas dinâmicos. Com esse objetivo em mente, é possível expressar a massa de um elemento infinitesimal *dm*, com superfície *dS* e volume *dV*, da seguinte maneira:

$$dm = \rho dV$$
 Equação 3.28

Onde ρ densidade do material que compõe o corpo. Assumindo ainda a aplicação de forças de superfície { φ } e de corpo { χ }, as forças externas que atuam no elemento podem ser descritas por:

$$\{F\} = \{\chi\} dV + \{\phi\} dS$$
Equação 3.29

Assim, a equação 3.27 pode ser rescrita, para o elemento infinitesimal, da seguinte forma:

$$\{\chi\}dV + \{\phi\}dS - \rho\{\ddot{u}\}dV = 0$$
 Equação 3.30

Dessa forma, é possível aplicar um deslocamento virtual { δu } na equação anterior. Então, integrando ao longo do volume e da superfície do corpo, é possível chegar ao seguinte resultado:

$$\int_{V} \{\chi\}^{T} \{\delta u\} dV + \int_{S} \{\phi\}^{T} \{\delta u\} dS - \int_{V} \rho\{\ddot{u}\}^{T} \{\delta u\} dV = \int_{V} \{\sigma\}^{T} \{\delta \epsilon\} dV \qquad \text{Equação 3.31}$$

Discretizando a equação conforme o procedimento apresentado anteriormente:

$$\{\delta u\} = [N]\{\delta U\} e \{\delta \ddot{u}\} = [N]\{\delta \ddot{U}\}$$
Equação 3.32

Onde { δU } representa os deslocamentos virtuais nodais e { δU } as acelerações nodais. Da equação 3.24, é possível escrever as deformações decorrentes dos deslocamentos virtuais nodais:

$$\{\delta\epsilon\} = [B]\{\delta U\}$$
 Equação 3.33

De posse dessas equações, é possível substituir 3.25, 3.32 e 3.33 em 3.31, obtendo a formulação equivalente para o sistema discretizado:

$$\{P\}^{T}\{\delta U\} + \int_{V} \{\chi\}^{T}[N]\{\delta U\}dV + \int_{S} \{\phi\}^{T}[N]\{\delta U\}dS - \int_{V} \rho\{\ddot{U}\}^{T}[N]^{T}[N]\{\delta U\}dV \qquad \text{Equação 3.34}$$
$$= \int_{V} (\{U\}^{T}[B]^{T}[D] - \alpha \Delta T[D_{T}])^{T}[B]\{\delta U\}dV$$

Onde {*P*} corresponde ao vetor com as cargas concentradas aplicadas nos pontos que que sofrem os deslocamentos { δU }. Na equação 3.34, os vetores {U}, { \ddot{U} }, { δU } e { $\delta \ddot{U}$ } possuem valores constantes e, portanto, podem ser retirados dos integrandos da equação. Desse modo:

$$\{P\}^{T}\{\delta U\} + \left(\int_{V} \{\chi\}^{T}[N]dV\right)\{\delta U\} + \left(\int_{S} \{\phi\}^{T}[N]dS\right)\{\delta U\} - \{\ddot{U}\}^{T}\left(\int_{V} \rho[N]^{T}[N]dV\right)\{\delta U\}$$

$$= \{U\}^{T}\left(\int_{V} [B]^{T}[D][B]dV\right)\{\delta U\} - \left(\int_{V} \alpha \Delta T[D_{T}]^{T}[B]dV\right)\{\delta U\}$$
Equação 3.35

Simplificando e rearranjando a equação, é possível obter:

$$\begin{pmatrix} \int_{V} \rho[N]^{T}[N]dV \\ V \end{pmatrix} \{ \ddot{U} \} + \begin{pmatrix} \int_{V} [B]^{T}[D][B]dV \\ V \end{pmatrix} \{ U \}$$

$$= \{ P \} - \int_{V} \alpha \Delta T [D_{T}]^{T}[B]dV + \int_{V} \{ \chi \}^{T}[N]dV + \int_{S} \{ \phi \}^{T}[N]dS$$
Equação 3.36

De forma reduzida, é possível apresentar a equação do seguinte modo:

$$[M]{\ddot{U}} + [K]{U} = {P} - {Q} + {F_{\chi}} + {F_{\phi}}$$
 Equação 3.37

Onde:

Matriz de massa do corpo:

Matriz de rigidez do corpo:

Vetor de carregamento nodal térmico equivalente:

 $[M] = \int \rho[N]^T [N] dV$

 $[K] = \int_{U} [B]^T [D] [B] dV$

 $\{F_{\chi}\} = \int_{\Omega} \{\chi\}^T [N] dV$

 $\left\{F_{\phi}\right\} = \int \{\phi\}^{T} [N] dS$

 $\{Q\} = \int_{V} \alpha \Delta T [D_T]^T [B] dV$

Vetor de forças nodais equivalentes devido a $\{\chi\}$:

Vetor de forças nodais equivalentes devido a $\{\varphi\}$:

 $E \{P\}$ corresponde ao vetor de forças concentradas.

Considerando que o presente trabalho se restringe ao caso de análise estática, o vetor de aceleração é nulo. Portanto, a equação 3.37, correspondente a equação de equilíbrio estático, pode ser simplificada na seguinte forma:

$$[K]{U} = {F} Equação 3.43$$

Onde {F} ao carregamento nodal de forças aplicado à estrutura do problema analisado.

Equação 3.39

Equação 3.38

Equação 3.40

Equação 3.42

Equação 3.41

Capítulo 4

Aplicação do Método dos Elementos Finitos

4.1. Introdução

Como já mencionado, a complexidade na solução de problemas de engenharia fundamentados em equações diferenciais motivou a procura por alternativas para esse procedimento. Nesse sentido, o método dos elementos finitos (MEF) associado a ferramentas computacionais surge como uma eficiente alternativa na obtenção de soluções eficientes. Segundo Hutton (2004), o MEF consiste em uma técnica computacional utilizada para solucionar problemas de valor do contorno, obtendo soluções aproximadas. O autor ainda esclarece que, grosso modo, um problema de valor do contorno consiste em um problema matemático onde uma ou mais variável dependente deve satisfazer uma equação diferencial em todo o seu domínio, satisfazendo condições de contorno dadas. Hutton (2004) ainda estabelece as ações executadas durante a etapa de pré-processamento de uma análise de problema baseado no MEF:

- Definir a domínio do problema.
- Selecionar o tipo de elemento finito a ser utilizado.
- Definir as propriedades materiais dos elementos.
- Definir as propriedades geométricas dos elementos, como suas dimensões.
- Definir a malha de elementos do problema.
- Definir as condições de contorno do problema.
- Definir o carregamento aplicado.

Dentre essas ações, a seleção do tipo de elemento finito a ser utilizado merece ser destacada, considerando que exerce influência sobre todo o resultado obtido ao longo da análise. Como já mencionado anteriormente, a aplicação do MEF é baseada em uma análise nodal, isto é, primeiramente são obtidos os valores das grandezas procuradas nos nós dos elementos finitos. Em seguida, essas grandezas são calculadas para os outros pontos do domínio do problema através de interpolações. Nesse sentido, cada tipo de elemento finito possuí características quem implicam em diferentes funções de

interpolação adotadas. Consequentemente, todos os resultados dependentes dessas funções obtidos na análise do problema serão diferentes. Para uma chapa sujeita a flexão em seu próprio plano, Alves Filho (2007) apresenta um comparativo entre elementos triangular e retangular para o calculo da deformação em que o segundo apresenta um resultado muito mais próximo da realidade, graças ao comportamento das funções de interpolação aplicadas a esse elemento. Tendo isso em mente, o tipo de elemento selecionado para as análises realizadas no presente trabalho é elemento quadrilateral de formulação isoparamétrica.

4.2. Formulação Isoparamétrica do Elemento Plano Quadrilateral

Hutton (2004) apresenta as vantagens e desvantagens dos elementos quadrangular e triangular, muito utilizados em análises aplicando o método dos elementos finitos. O elemento triangular, embora a sua geometria seja propícia a aplicação em geometrias de estruturas irregulares, apresenta as deformações em seu interior como constante. Como consequência, um estudo eficaz demanda a aplicação de uma malha com muitos elementos. Por sua vez, o elemento retangular é capaz de contornar a questão da deformação constante em seu interior, visto que essa grandeza apresenta um comportamento linear em seu interior. Como desvantagem, o elemento quadrilateral, dada a sua geometria, não é passível de aplicação em geometrias complexas. Diante desse cenário, Hutton (2004) apresenta o elemento quadrilateral de quatro nós como capaz de representar a variação de deformação e tensão de forma eficaz, além cobrir geometrias de problemas complexas. O elemento é representado a seguir:



Figura 4.1. Elemento plano quadrilateral de formulação isoparamétrica. Fonte: Hutton (2004)

A formulação isoparamétrica estabelece que as funções de mapeamento geométrico são idênticas às funções de interpolação. O elemento mostrado na figura 4.1 (a) é representado no sistema de coordenadas globais, com suas coordenadas nodais. Esse elemento é obtido aplicando as funções de mapeamento no elemento quadrangular representado na figura 4.1 (b), definido no plano *rs*. Esse mapeamento é definido pelas seguintes equações:

$$x = \sum_{i=1}^{4} N_i(r,s) x_i$$
Equação 4.1
$$y = \sum_{i=1}^{4} N_i(r,s) y_i$$
Equação 4.2

Por definição, as funções de interpolação são as mesmas que a de interpolação geométrica. Portanto:

$$u(x,y) = \sum_{i=1}^{4} N_i(r,s)u_i$$
Equação 4.3
$$v(x,y) = \sum_{i=1}^{4} N_i(r,s)v_i$$
Equação 4.4

Nesse caso, a equação 3.11, que apresenta as componentes de deformação, pode ser escrita da seguinte para o estado plano de tensão:

$$\begin{cases} \epsilon_{xx} \\ \epsilon_{yy} \\ \gamma_{xy} \\ \gamma_{xy} \end{cases} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} { u \\ v }$$
 Equação 4.5

As derivadas de u em relação e r e s podem ser obtidas aplicando a regra da cadeia. Dessa forma, é possível obter:

39

Equação 4.6

 $\frac{\partial u}{\partial r} = \frac{\partial u}{\partial x}\frac{\partial x}{\partial r} + \frac{\partial u}{\partial y}\frac{\partial y}{\partial r}$ $\frac{\partial u}{\partial s} = \frac{\partial u}{\partial x}\frac{\partial x}{\partial s} + \frac{\partial u}{\partial y}\frac{\partial y}{\partial s}$

O procedimento é análogo para a variável v. A equação 4.6 pode ser apresentada de forma matricial:

> Equação 4.7 $\begin{cases} \frac{\partial u}{\partial r} \\ \frac{\partial u}{\partial u} \end{cases} = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial y}{\partial r} \\ \frac{\partial x}{\partial x} & \frac{\partial y}{\partial v} \end{bmatrix} \begin{pmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial u} \end{pmatrix} = \begin{bmatrix} J \end{bmatrix} \begin{cases} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial u} \\ \frac{\partial u}{\partial u} \end{cases}$

Onde a matriz Jacobiana é definida como:

 $[J] = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial y}{\partial r} \\ \frac{\partial x}{\partial r} & \frac{\partial y}{\partial r} \end{bmatrix}$

É possível resolver o sistema representado pela equação 4.7 para as derivadas de u em relação a x e y. Contudo, o processo de inversão da matriz Jacobiana não é uma tarefa muito simples. Dessa forma, é possível recorrer a Regra de Cramer:

> $\frac{\partial u}{\partial x} = \frac{\begin{vmatrix} \frac{\partial u}{\partial r} & J_{12} \\ \frac{\partial u}{\partial s} & J_{22} \end{vmatrix}}{|J|} = \frac{1}{|J|} [J_{22} - J_{12}] \begin{cases} \frac{\partial u}{\partial r} \\ \frac{\partial u}{\partial r} \\ \frac{\partial u}{\partial s} \end{cases}$ $\frac{\partial u}{\partial y} = \frac{\begin{vmatrix} J_{11} & \frac{\partial u}{\partial r} \\ J_{21} & \frac{\partial u}{\partial s} \end{vmatrix}}{|J|} = \frac{1}{|J|} \left[-J_{21} + J_{11} \right] \begin{cases} \frac{\partial u}{\partial r} \\ \frac{\partial u}{\partial s} \end{cases}$

Equação 4.9

Equação 4.8

De forma matricial, é possível expressar a equação 4.9 da seguinte maneira:

$$\frac{\partial u}{\partial x} \\
\frac{\partial u}{\partial y} = \frac{1}{|J|} \begin{bmatrix} J_{22} & -J_{12} \\ -J_{21} & J_{11} \end{bmatrix} \begin{cases} \frac{\partial u}{\partial r} \\ \frac{\partial u}{\partial s} \end{cases}$$
Equação 4.10

O determinante da matriz Jacobiana J é conhecido como Jacobiano. Além disso, analogamente para v é possível obter:

$$\begin{cases} \frac{\partial v}{\partial x} \\ \frac{\partial v}{\partial y} \end{cases} = \frac{1}{|J|} \begin{bmatrix} J_{22} & -J_{12} \\ -J_{21} & J_{11} \end{bmatrix} \begin{pmatrix} \frac{\partial v}{\partial r} \\ \frac{\partial v}{\partial s} \end{pmatrix}$$
Equação 4.11

Então, é possível retornar ao problema da determinação dos componentes de deformação, representado pela equação 4.5. Substituindo os resultados obtidos nas equações 4.10 e 4.11, é possível obter:

$$\varepsilon = \begin{cases} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{cases} = \begin{cases} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \end{cases} = \frac{1}{|J|} \begin{bmatrix} J_{22} & -J_{12} & 0 & 0 \\ 0 & 0 & -J_{21} & J_{11} \\ -J_{21} & J_{11} & J_{22} & -J_{12} \end{bmatrix} \begin{cases} \frac{\partial u}{\partial r} \\ \frac{\partial u}{\partial s} \\ \frac{\partial v}{\partial r} \\ \frac{\partial v}{\partial s} \\ \frac$$

Na equação 4.12, a matriz *G* é chamada de mapeamento geométrico. Ela é definida como:

$$[G] = \frac{1}{|J|} \begin{bmatrix} J_{22} & -J_{12} & 0 & 0\\ 0 & 0 & -J_{21} & J_{11}\\ -J_{21} & J_{11} & J_{22} & -J_{12} \end{bmatrix}$$
Equação 4.13

É possível apresentar as equações 4.3 e 4.4 de forma matricial, expressa para os dois graus de liberdade de cada um dos quatro nós presentes em um elemento:

$$\begin{cases} \frac{\partial u}{\partial r} \\ \frac{\partial u}{\partial s} \\ \frac{\partial v}{\partial s} \\ \frac{\partial v}{\partial s} \\ \frac{\partial v}{\partial s} \\ \frac{\partial v}{\partial s} \end{cases} = \begin{bmatrix} \frac{\partial N_1}{\partial r} & \frac{\partial N_2}{\partial r} & \frac{\partial N_3}{\partial r} & \frac{\partial N_4}{\partial r} & 0 & 0 & 0 & 0 \\ \frac{\partial N_1}{\partial s} & \frac{\partial N_2}{\partial s} & \frac{\partial N_3}{\partial s} & \frac{\partial N_4}{\partial s} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{\partial N_1}{\partial r} & \frac{\partial N_2}{\partial r} & \frac{\partial N_3}{\partial r} & \frac{\partial N_4}{\partial r} \\ 0 & 0 & 0 & 0 & \frac{\partial N_1}{\partial s} & \frac{\partial N_2}{\partial s} & \frac{\partial N_3}{\partial s} & \frac{\partial N_4}{\partial s} \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{pmatrix}$$
Equação 4.14

Alternativamente, é possível representar a equação 4.14 da seguinte forma:

$$\begin{cases} \frac{\partial u}{\partial r} \\ \frac{\partial u}{\partial s} \\ \frac{\partial v}{\partial r} \\ \frac{\partial v}{\partial s} \\ \frac{\partial v}{\partial s} \\ \end{cases} = [P]\{\delta\}$$

Equação 4.15

Onde *P* é a matriz de derivadas parciais das funções de interpolação e δ corresponde ao vetor de deslocamentos nodais do elemento.

É possível associar a equação 4.15 à equação 4.12 para obter a equação que relaciona a deformação aos deslocamentos nodais. Desse modo:

$${\varepsilon} = [G][P]{\delta}$$
 Equação 4.16

Por analogia, é possível representar a equação 4.16 na forma de desenvolvimentos comuns em elementos finitos, definindo [B] = [G][P]:

$$\{\varepsilon\} = [B]\{\delta\}$$
 Equação 4.17

A matriz rigidez do elemento é definida por:

$$\left[k^{(e)}\right] = t \int_{A} [B]^{T}[D][B] dA$$
 Equação 4.18

Onde *t* representa a espessura do elemento, e *D* é a matriz elasticidade, apresentada, para o estado plano de tensões, na equação 3.22.

A equação 4.18 apresenta uma integração que deve ser realizada no sistema global *xy*. Contudo, a matriz *B* é definida no sistema local *rs*. No espaço físico, *dA* é definida como *dxdy*. Todavia, é necessário que se integra de -1 a 1 nas coordenadas locais. No caso da formulação isoparamétrica, esse procedimento não é tão simples e a sua dedução não é inerente ao objetivo do presente trabalho. Portanto, o ajuste de coordenadas é simplesmente apresentado em seguida:

$$dA = dxdy = |J|drds$$
 Equação 4.19

Dessa forma, a equação 4.18 pode ser reescrita da seguinte forma:

$$\left[k^{(e)}\right] = t \int_{A} [B]^{T}[D][B]|J|drds = t \int_{-1}^{1} \int_{-1}^{1} [B]^{T}[D][B]|J|drds$$
Equação 4.20

As integrações que devem ser realizadas na equação 4.20 são de elevada complexidade ou até mesmo impossíveis de obter soluções exatas, segundo Hutton (2004). Portanto, usa solução numérica baseada na quadratura de Gauss será adotada, realizando a integração com 2x2 pontos de Gauss. Dessa forma, a equação 4.20 pode ser apresentada como:

$$\left[k^{(e)}\right] = t \sum_{i=1}^{p} \sum_{j=1}^{q} W_{i} W_{j} \left[B(r_{i}, s_{j})\right]^{T} \left[D\right] \left[B(r_{i}, s_{j})\right] |J(r_{i}, s_{j})| dr ds$$
Equação 4.21

Onde W_i e W_j são os pesos e r_i e s_j são os pontos da quadratura de Gauss.

As funções de forma utilizadas nas deduções são as mesmas adotadas por Hutton (2004):

$$N_{1}(r,s) = \frac{1}{4}(1-r)(1-s)$$
Equação 4.22

$$N_{2}(r,s) = \frac{1}{4}(1+r)(1-s)$$
Equação 4.23

$$N_{3}(r,s) = \frac{1}{4}(1+r)(1+s)$$
Equação 4.24

$$N_4(r,s) = \frac{1}{4}(1-r)(1+s)$$
 Equação 4.25

Capítulo 5

Otimização Estrutural

5.1. Introdução

Existem diversos métodos de otimização topológica baseados em diferentes abordagens e dinâmicas. Dentre esses, os objetos de estudo do presente trabalho são os métodos Bi-directional Evolutionary Structural Optimization (BESO) e Solid Isotropic Material with Penalization Method (SIMP). Neste capítulo, o desenvolvimento teórico de cada um dos métodos e suas respectivas formulações para o problema da minimização do compliance das estruturas serão apresentados e detalhados.

5.2. Bi-directional Evolutionary Structural Optimization (BESO)

Segundo Huang e Xie (2010), o método Bi-directional Evolutionary Structural Optimization (BESO) se diferencia do método Evolutionary Structural Optimization (ESO) devido a possiblidade de se acrescentar ou retirar elementos no processo de otimização da estrutura no método bidirecional. Enquanto o ESO consiste em remover material ineficiente do projeto visando a aproximação da estrutura ideal, o BESO considera a possibilidade de acréscimo de material na concepção da estrutura ideal.

O método BESO estima números de sensibilidade para os elementos vazios da estrutura através de uma extrapolação linear aplicada ao campo de deslocamentos resultante da análise de elementos finitos. Em seguida, os elementos vazios com números de sensibilidade mais altos são transformados em elementos sólidos, enquanto os elementos sólidos com baixos números de sensibilidade tornam-se elementos vazios. Essas transformações são governadas por dois parâmetros independentes: a taxa de inclusão (IR) e a taxa de rejeição (RR).

Huang e Xie (2010) aplicam o método BESO visando obter a topologia otimizada que apresenta a maior rigidez com o menor volume de material possível. Como o método é baseado no acréscimo ou remoção de elementos, nesse problema, os próprios elementos finitos são as variáveis de projeto. Dessa forma, os autores descrevem esse problema de otimização matematicamente da seguinte forma:

Minimizar:

$$C = \frac{1}{2} \boldsymbol{f}^{\mathrm{T}} \boldsymbol{u}$$
 Equação 5.1

Sujeito a:

$$V^* - \sum_{i=1}^{N} V_i x_i = 0$$
Equação 5.2
$$x_i = 0 \text{ ou } 1$$
Equação 5.3

Onde C corresponde ao compliance, **f** ao vetor de carga aplicado e **u** ao vetor de deslocamento. Por sua vez, *V*^{*i*} corresponde ao volume de um elemento *i* e *V*^{*} é o volume predefinido da estrutura. Finalmente, *N* representa o número total de elementos que compõem a malha e x_i representa a presença (1) ou ausência (0) de elemento na estrutura.

No processo de otimização, quando um elemento é removido da estrutura a alteração no compliance médio ou na energia de deformação total da estrutura é equivalente à energia de deformação elementar, segundo (Chu et al. 1996). Essa relação pode ser apresentada da seguinte forma:

$$\alpha_i^e = \Delta C_i = \frac{1}{2} \boldsymbol{u}_i^T \boldsymbol{K}_i \boldsymbol{u}_i$$
 Equação 5.4

Onde u_i representa o vetor de deslocamento nodal do elemento *i* e K_i corresponde a matriz rigidez do elemento. Em situações que a malha utilizada não é uniforme, é necessário que o número de sensibilidade compute o efeito do volume do elemento. Essa situação pode ser representada permutando a sensibilidade pela densidade de energia de tensão, que pode ser representada por:

$$\boldsymbol{\alpha}_{i}^{\boldsymbol{e}} = \boldsymbol{e}_{i} = \left(\frac{1}{2}\boldsymbol{u}_{i}^{T}\boldsymbol{K}_{i}\boldsymbol{u}_{i}\right)/V_{i}$$
 Equação 5.5

No método BESO, um esquema de filtro é utilizado, visando suavizar o número de sensibilidade dos elementos em todo o domínio do problema e evitar os problemas de dependência de malha e tabuleiro de xadrez. Segundo Huang e Xie (2010), uma análise de elementos finitos bidimensional ou tridimensional, utilizando uma ordem baixa de elementos, pode resultar em descontinuidades nos números de sensibilidade dos elementos. (Jog e Harber 1996) apontam que essa condição resulta no problema do tabuleiro de xadrez. Por sua vez, Huang e Xie (2010) descrevem o problema da dependência da malha como a condição em que as topologias ideias obtidas são dependentes das malhas de elementos finitos utilizados. (Bendsøe e Sigmund 2003) defendem que uma malha mais refinada deve implicar em uma melhor modelagem de elementos finitos e melhor descrição dos limites, para uma estrutura otimizada.



Figura 5.1. Um exemplo de tabuleiro de xadrez resultante no método ESO. Fonte: Huang e Xie (2010).

O esquema de filtro proposto por Huang e Xie (2010), inicialmente, define os números de sensibilidade nodal que não possuem significado físico com a média dos números de sensibilidade elementar da seguinte forma:

$$\alpha_j^n = \sum_{i=1}^M w_i \alpha_i^e$$
Equação

Onde *M* corresponde ao número de elementos ligados pelo *j*-ésimo nó, *w*_i representa o fator de peso atribuído ao i-ésimo elemento, em que:

5.6

$$\sum_{i=1}^{M} w_i = 1$$

Equação 5.7

Dessa forma, *w_i* pode ser definido como:

$$w_i = \frac{1}{M-1} \left(1 - \frac{r_{ij}}{\sum_{i=1}^{M} r_{ij}} \right)$$
 Equação 5.8

Onde *r_{ij}* corresponde à distância entre o centro do *i*-ésimo elemento e o *j*-ésimo nó. O filtro aplicado é atrelado a uma escala de comprimento *r_{min}*, que é independente do refinamento da malha. A sua principal contribuição no processo é identificar quais nós possuem influência sobre a sensibilidade do elemento *i* e atuam para aprimorar o cálculo desse número. Segundo Huang e Xie (2010), *r_{min}* deve ser capaz de encobrir mais de um elemento. Desse modo, o número de sensibilidade aprimorado para o elemento *i* é dado por:

$$\alpha_i = \frac{\sum_{j=1}^{K} w(r_{ij}) \alpha_j^n}{\sum_{j=1}^{K} w(r_{ij})}$$
Equação 5.9

K representa a quantidade de nós compreendidos pelo subdomínio $\Omega_i e w(r_{ij})$ é o fator de peso linear, definido por:

$$w(r_{ij}) = r_{min} - r_{ij}$$
 (j = 1, 2, ..., K) Equação 5.10

De acordo com Huang e Xie (2010), os esquemas de filtro são puramente heurísticos. Entretanto, ele é capaz de contornar problemas numéricos enfrentados durante a resolução de um problema de otimização estrutural, como o tabuleiro de xadrez e a dependência de malhas. Ainda segundo os autores, os esquemas demandam poucos recursos computacionais e são simples de serem implementados. Tudo isso, justifica as suas aplicações no método BESO.



Figura 5.2. Nós contemplados pelo domínio Ω_i, aplicado no esquema de filtro utilizado para o elemento *i*. Fonte: Huang e Xie (2010).

Outro ponto de atenção no processo de otimização estrutural apontado por Huang e Xie (2010) é a convergência da função objetivo e a topologia otimizada correspondente. É possível observar em muitos casos grandes oscilações no comportamento da função objetivo durante a sua evolução no processo de otimização. Essa condição é uma consequência do fato de que os números de sensibilidade dos elementos sólidos e vazios são fundamentados em variáveis discretas de projeto, que podem estar presentes (1) ou ausentes (0). Esse comportamento representa uma dificuldade para a convergência da função objetivo. Diante desse cenário, Huang e Xie (2007) descobriram que utilizar a média do número de sensibilidade com suas informações históricas é uma técnica capaz de contornar essa situação. Desse modo, o número de sensibilidade pode ser calculado da seguinte maneira:

$$\alpha_i = \frac{\alpha_i^k + \alpha_i^{k-1}}{2}$$
 Equação 5.11

Onde *k* corresponde a atual iteração no processo de otimização. Na próxima iteração, $\alpha_i^k = \alpha_i$. Desse modo, o calculo do número de sensibilidade sempre considera o histórico no calculo dessa grandeza e a função objetivo torna-se mais estável em seu processo de convergência.

Contornadas as dificuldades encontradas durante o processo de aplicação do método BESO, a obtenção da topologia otimizada torna-se viável. Nesse sentido, a cada

iteração do processo, o volume que a estrutura deve atingir é atualizado até que o volume de restrição definido seja atingido. O parâmetro que coordena a evolução do volume da estrutura é a *evolutionary volume ratio* (*ER*). O comportamento descrito é representado matematicamente por:

$$V_{k+1} = V_k(1 \pm ER)$$
 (k = 1, 2, 3 ...) Equação 5.12

Na evolução do processo, quando o volume de restrição (V) for alcançado, ele será mantido constante pelas próximas iterações do processo. Ou seja:

$$V_{k+1} = V^*$$
 Equação 5.13

Ainda segundo Huang e Xie (2010), os números de sensibilidade dos elementos (vazios ou sólidos) são calculados conforme descrito anteriormente, e então os elementos são listados de forma decrescente de acordo com seus números de sensibilidade. Os elementos vazios tornam-se sólidos quando satisfazem o seguinte critério:

$$\alpha_i > \alpha_{add}^{th}$$
 Equação 5.14

Enquanto os elementos sólidos são removidos quando:

$$a_i \leq a_{del}^{th}$$
 Equação 5.15

Onde α_{add}^{th} e α_{del}^{th} correspondem aos números de sensibilidade limites. É importante notar que α_{del}^{th} deve sempre ser menor ou igual a α_{add}^{th} .

A análise de elementos finitos, remoção e adição de elementos se repete até que o volume de restrição (V) seja atingido e que o critério de convergência seja atingido. Esse critério é baseado no comportamento da função objeto, e pode ser descrito por:

$$error = \frac{\left|\sum_{i=1}^{N} C_{k-i+1} - \sum_{i=1}^{N} C_{k-i+1}\right|}{\sum_{i=1}^{N} C_{k-i+1}} \le \tau$$
Equação 5.16

Onde *k* corresponde à iteração em vigência, τ é valor de tolerância permitido para a convergência e *N*, geralmente adotado como 5, representa um parâmetro que define o número de iterações para o cálculo do erro.

Huang e Xie (2010) resumem o método BESO através da seguinte lista:

- 1. Discretizar o domínio do problema e definir quais elementos são sólidos ou vazios no início do processo.
- Realizar a análise de elementos finitos e calcular o número de sensibilidade de cada elemento.
- Calcular os números de sensibilidade considerando seus históricos nas iterações, como descrito anteriormente.
- 4. Determinar o volume esperado para a próxima iteração.
- 5. Adicionar e remover elementos conforme o procedimento descrito.
- Repetir os passos de 2 a 5, até que o volume de restrição V^{*} seja atingido e o critério de convergência seja satisfeito.

5.3. Solid Isotropic Material with Penalization Method (SIMP)

O método Solid Isotropic Material with Penalization Method (SIMP) utilizada uma abordagem sustentada no principio de que os elementos finitos são formados por materiais isotrópicos, podendo assumir densidades variáveis, segundo Huang e Xie (2010). Os primeiros estudos envolvendo a otimização topológica utilizando uma abordagem de distribuição do material por parametrização do domínio do problema foram realizados por Bendsøe, no final dos anos de 1980, segundo Krishnaa et al. (2017). Desde então, o método SIMP, fundamentado nessa abordagem, vem recebendo grande atenção dos pesquisadores. Ainda segundo Krishnaa et al. (2017), o método SIMP, também conhecido como método das densidades, opera atribuindo um número de densidade para cada um dos elementos finitos que compõem o domínio discretizado. Esses números variam entre 0 (vazio) e 1 (sólido), possibilitando a existência de densidades intermediárias. Desse modo, é possível descrever o módulo de Young ao longo do domínio da seguinte forma:

$$E_i = x_i^p E_0$$
 Equação 5.17

50

Onde E_0 corresponde ao módulo de Young do material, x_i representa a densidade do elemento i e p equivale ao parâmetro de penalidade para as densidades intermediárias (geralmente, \ge 3). Este último parâmetro possui a função de penalizar os elementos com densidades intermediarias, considerando que essa condição pode ser enxergada como um "material artificial", impossível de ser obtido na pratica, segundo Krishnaa et al. (2017). Dessa forma, é possível obter um resultado mais próximo da distribuição sólido-vazio.

Como consequência do modelo material, a matriz de rigidez penalizada dos elementos pode ser apresentada como:

$$K_i = x_i^p K_i^0$$
 Equação 5.18

Onde K_i^{ρ} representa a matriz de rigidez elementar. De acordo com Huang e Xie (2010), supondo que o coeficiente de Poisson não dependa das variáveis de projeto, a matriz de rigidez global da estrutura pode ser expressa por uma soma das matrizes elementares:

$$\boldsymbol{K} = \sum_{i} x_{i}^{p} \boldsymbol{K}_{i}^{0}$$
 Equação 5.19

Analogamente ao apresentado para o método BESO, é possível apresentar o problema da otimização topológica utilizando o método SIMP de forma matemática. Para o caso de otimizar a topologia com relação a rigidez e considerando uma restrição de volume, o problema pode ser proposto da seguinte forma:

Minimizar:

 $c = f^T u$ Equação 5.20

Sujeito a:

$$V^* - \sum_{i=1}^{N} V_i x_i = 0$$

Equação 5.21
$$0 < x_{min} \le x_i \le 1$$

Equação 5.22

Onde *c* representa o compliance, que corresponde ao dobro da grandeza apresentada para o método BESO (*C*). Por sua vez, *f* equivale ao vetor de carga aplicado e *u* ao vetor de deslocamento. Mais uma vez, o problema de otimização topológica está sujeito a uma restrição de volume definida por V^{*} , equivalente a soma dos volumes elementares *Vi*. Finalmente, as densidades dos elementos são restritas a um valor mínimo *x*_{min} muito pequeno e maior que 0, evitando o uso de 0 para elementos vazios e garantindo a estabilidade numérica da análise de elementos finitos, e 1, que corresponde aos elementos sólidos. Dessa forma, os elementos com densidades intermedirias podem ser representados fisicamente por materiais porosos, por exemplo.

Derivando a equação do compliance, Huang e Xie (2010) apresentam a expressão dos números de sensibilidade da seguinte forma:

$$\frac{\partial c}{\partial x_i} = -p x_i^{p-1} \boldsymbol{u}_i^T \boldsymbol{K}_i^0 \boldsymbol{u}_i$$
 Equação 5.23

Na literatura, existem algumas abordagens utilizadas para solucionar o problema de otimização proposto. Sigmund (2001) apresentam como exemplos o método de critério de otimalidade (OC), métodos de programação linear sequencial (SLP) e método de movimentação de assíntotas (MMA por Svanberg 1987). Por simplicidade, o método adotado é o de critério de otimalidade (OC), que possui um esquema heurístico de atualização para as variáveis de projeto formulado por Bendsøe (1995), formulado como:

$$x_{i}^{K+1} = \begin{cases} \max(x_{\min}, x_{i}^{K} - m) \text{ se } x_{i}^{K} B_{i}^{\eta} \leq \max(x_{\min}, x_{i}^{K} - m) & \text{Equação 5.24} \\ \min(1, x_{i}^{K} + m) \text{ se } \min(1, x_{i}^{K} + m) \leq x_{i}^{k} B_{i}^{\eta} \\ x_{i}^{k} B_{i}^{\eta} \text{ de algum outro modo} \end{cases}$$

Onde x_i^K corresponde a variável de projeto na iteração K, η equivale a um coeficiente numérico de amortecimento (geralmente, 0,5), m representa um limite de movimento positivo e B_i é obtido pela seguinte condição de otimização:

$$B_i = \lambda^{-1} p x_i^{p-1} \boldsymbol{u}_i^T \boldsymbol{K}_i^0 \boldsymbol{u}_i$$
 Equação 5.25

Onde λ representa um Multiplicador de Lagrange, que pode ser obtido utilizando o método da bissecção ou um método de Newton, segundo Bendsøe and Sigmund (2003).

Mais uma vez, os problemas de dependência da malha e tabuleiro de xadrez podem ocorrer, assim como no método BESO. Portanto, a aplicação de um filtro pode viabilizar um resultado satisfatório. (Sigmund 1994, 1997) apresentou uma técnica de filtro que, com base em várias aplicações testadas pelo autor, produziu projetos que contornaram esses problemas. O esquema de filtro apresenta uma modificação no cálculo dos números de sensibilidade dos elementos:

$$\frac{\partial c}{\partial x_i} = \frac{1}{x_i \sum_{j=1}^N H_{ij}} \sum_{j=1}^N H_{ij} x_j \frac{\partial c}{\partial x_j}$$
 Equação 5.26

Onde *N* corresponde ao número de elementos na malha de elementos finitos adotada e H_{ij} equivale ao fator de peso, que é definido como:

$$H_{ij} = r_{min} - r_{ij}, \ \left\{ i \in N \mid r_{ij} \le r_{min} \right\}$$
Equação 5.27

Onde r_{ij} corresponde a distancia entre os centros dos elementos *i* e *j*. Além disso, o fator de peso H_{ij} é nulo em posições exteriores ao circulo representado na figura 5.2.

Capítulo 6

Procedimento Experimental

O procedimento experimental do presente trabalho consiste em resolver numericamente problemas selecionados de análise estrutural através de uma abordagem utilizando o método dos elementos finitos, apresentar topologias otimizadas através dos métodos de otimização topológica estrutural evolucionária Bi-directional Evolutionary Structural Optimization (BESO) e Solid Isotropic Material with Penalization method (SIMP) e analisar as particularidades dos resultados obtidos. Para isso, serão desenvolvidos códigos de Matlab® capazes de realizar a análise por MEF e aplicar os algoritmos para a otimização das topologias desses problemas. Além disso, alguns parâmetros de otimização associados a cada um dos métodos terão as suas influências sobre os resultados obtidos avaliadas. Todos os problemas analisados consistem em problemas de minimização de compliance de estruturas mecânicas.

Os códigos utilizados neste trabalho são adaptações de códigos difundidos e amplamente utilizados pelo Departamento de Mecânica Computacional da Faculdade de Engenharia Mecânica da UNICAMP. O código BESO original foi desenvolvido por Tainan Calixto e Renato Pavanello. Enquanto o código SIMP corresponde ao trabalho realizado por Sigmund (2001).

Em seguida, é apresentada a sistemática básica que representa o procedimento experimental deste trabalho.

6.1. Sistemática de Otimização Topológica Estrutural

- 1) Discretizar o domínio inicial utilizando malhas de elementos finitos.
- Resolver o problema estrutural aplicando seus carregamentos e condições de contorno.
- 3) Definir as funções de interpolação ideais para o estudo em andamento.
- Analisar o comportamento dos elementos finitos com relação às grandezas físicas correspondentes às funções objetivos.
- 5) Alterar a topologia da estrutura, de forma que ela se aproxime cada vez mais de uma solução atinjam os critérios definidos como ideais para o problema.

Capítulo 7

Resultados e Discussões

Utilizando os códigos de Matlab® adaptados, os problemas de minimizar o compliance da Estrutura de Michell e da Estrutura em Forma de Mão-Francesa serão resolvidos utilizando o método BESO e SIMP. Na sequência, foram selecionados parâmetros de otimização inerentes a cada um dos métodos e analisadas as influências das variações de cada um sobre as soluções obtidas. Para o método BESO, na Estrutura de Michell foi variado o parâmetro *ER* e na Estrutura em Forma de Mão-Francesa foi variado o parâmetro *AR_{max}*. Já para o método SIMP, na Estrutura de Michell foi variado o parâmetro *A* e na Estrutura em Forma de Mão-Francesa foi variado o parâmetro de penalidade p e na Estrutura em Forma de Mão-Francesa foi variada a dimensão do filtro *rmin*. A escolha desses exemplos em especifico se justifica por serem problemas clássicos da otimização topológica, possibilitando com que os resultados obtidos sejam comparados com informações existentes na literatura.

7.1. Aplicação do Método BESO

7.1.1. O Problema da Estrutura de Michell

7.1.1.1. Análise Inicial

O domínio inicial adotado para o problema da estrutura de Michell consiste em uma chapa de largura 10 m, altura 5 m e espessura 0,001 m. As condições de contorno são representadas por uma força de intensidade 10 kN aplicada no ponto médio do lado inferior da chapa para baixo e a restrição de deslocamentos nas duas extremidades desse mesmo lado. O problema é representado na figura 7.1:



Figura 7.1. Domínio inicial para a Estrutura de Michell.

O sistema é então discretizado em uma malha de elementos finitos composta por 1250 elementos quadrilaterais com quatro nós. As funções de interpolação utilizadas foram apresentadas da equação 4.22 a equação 4.25.

Os elementos finitos são quadrados de lado 0,2 m. A lógica da numeração dos elementos e eixos adotada é representada pela figura 7.2:



Figura 7.2. Numeração de elementos e eixos no domínio da Estrutura de Michell.

As características adotadas para o material são módulo de elasticidade E = 100GPa e coeficiente de Poisson v = 0,3. Dessa forma, compondo a matriz rigidez global da estrutura, aplicando as condições de contorno e resolvendo o problema representado pelas equações 2.1 e 2.2 é possível obter os deslocamentos e forças nodais que compõe os elementos finitos.

Inicialmente, para a otimização estrutural que minimiza o compliance da estrutura representada na figura 7.1, foram definidas uma fração de volume final de 50%, porcentagem de remoção de material após cada iteração de 1% e porcentagem de adição de material máxima igual a 5%. Além disso, o raio do filtro adotado corresponde a 0,6 metro e a precisão de convergência equivale a 1 x 10⁻⁴. A estrutura otimizada obtida é:



Figura 7.3. Estrutura de Michell otimizada minimizando o compliance.

Após calcular os deslocamentos nodais, é possível obter as grandezas físicas relacionadas para os elementos finitos que compõe a malha do problema. Para isso, aplicando as definições decorrentes da teoria da elasticidade, é possível obter as tensões atuantes nos elementos. Finalmente, com as componentes de tensões de cada elemento finito, foi calculada a tensão equivalente de Von Misses atuante em cada um deles, de acordo com a equação 6.3:

$$\sigma_{\text{EVM}} = \sqrt{(\sigma_{XX}^2 + \sigma_{YY}^2 + \sigma_{XX} \cdot \sigma_{YY} + 3 \cdot \sigma_{XY}^2)}$$
Equação 7.1

Onde σ_{EVM} é a tensão equivalente de Von Misses, σ_{XX} e σ_{YY} são as componentes normais do tensor de tensões nas direções X e Y, e σ_{XY} representa a componente de cisalhamento. O mapa de cores de tensão equivalente de Von Misses obtido para a estrutura otimizada é apresentado a seguir:



Figura 7.4. Mapa de cores da tensão equivalente de Von misses na Estrutura de Michell otimizada.

Finalmente, o código desenvolvido gera um curvas de compliance e volume da estrutura em função da iteração do programa, que permite comprovar a convergência dos resultados e corrobora a relação esperada entre volume da estrutura e compliance. O gráfico obtido é:



Figura 7.5. Compliance e fração de volume em função da iteração do programa.

Essa análise foi realizada com base na formulação isoparamétrica do elemento plano quadrilateral proposta por Hutton (2004).

Os resultados obtidos nessa análise inicial condizem com o que é esperado pela literatura. Primeiramente, é possível notar a simetria no comportamento das tensões equivalentes de Von Misses e deslocamentos dos elementos (apresentados no programa de Matlab®), o que condiz com o esperado, considerando que o problema apresenta simetria entre suas dimensões, características materiais e condições de contorno. Além disso, os pontos de restrições, onde ocorrem as reações para o equilíbrio da estrutura, e aplicação da força de 10 kN apresentam as maiores tensões equivalente de Von Misses, assim como era esperado. A figura 7.3 apresenta uma estrutura que condiz com o resultado obtido por Silva (2001) utilizando um método evolucionário de otimização e com a solução analítica proposta por Hemp (1973). A curva representa na figura 7.5 apresenta informações importantes para a análise da estrutura proposta. Dada a definição do compliance como sendo a energia de deformação, é de se esperar que a redução no volume da estrutura, isto é, a remoção de material no processo de otimização, implique em um decréscimo na rigidez da estrutura e, consequentemente, um acréscimo no

compliance. Esse comportamento é exatamente o encontrado na curva. Além disso, é perceptível que a partir da iteração de número 70, os valores de compliance e fração de volume convergem para 4,59 Nm e 50%, respectivamente. Desse modo, é possível obter a solução, uma estrutura otimizada, para o problema de otimização estrutural de rigidez proposto, respeitando as condições de restrição estabelecidas.





Os resultados obtidos para a otimização da estrutura são resumidos na seguinte tabela:

Tabela 7.1. Resumo de resultados para a otimização inicial da Estrutura de Michell.

Iterações	Compliance	Fração de Volume
79	4,59 Nm	50%

7.1.1.2. Variando o Parâmetro ER

O parâmetro Evolutionary Remotion Ratio (*ER*) consiste na porcentagem de evolução de material no domínio da estrutura analisada em cada iteração. Huang e Xie (2007) alegam que, grosso modo, *ER*'s menores resultam em soluções mais eficientes. Por sua vez, Yang (1999) descreve o parâmetro como a velocidade de evolução da estrutura, agindo de forma semelhante ao limite de movimento utilizado na programação matemática, definindo o tamanho das etapas da otimização. Yang (1999) ainda destaca que *ER*'s de valores menores, embora resultem em evoluções de projeto mais suaves, em conformidade com Huang e Xie (2007), implicam em maiores custos computacionais.

Desse modo, esta sessão tem como objetivo impor uma faixa de diferentes valores de *ER*'s e investigar o resultado dessa variação nas topologias obtidas. Para tanto, os outros parâmetros utilizados na Estrutura de Michell da seção anterior foram mantidos fixos.

Inicialmente, foram selecionados doze valores diferentes próximo ao 1% do problema original e geradas as topologias otimizadas para o compliance. Os valores selecionados e os resultados obtidos estão resumidos na tabela 7.2 e da figura 7.7 a figura 7.10:

ER	Iterações	Compliance
10,00%	34	4,6027 Nm
8,00%	36	4,6052 Nm
6,00%	30	4,6033 Nm
4,00%	28	4,5704 Nm
2,00%	44	4,5739 Nm
1,00%	79	4,5911 Nm
0,50%	149	4,5711 Nm
0,25%	288	4,5805 Nm
0,20%	360	4,5805 Nm
0,15%	476	4,5805 Nm
0,10%	704	4,5805 Nm
0,05%	1396	4,5805 Nm

Tabela 7.2. Resumo de resultados para a otimização inicial da Estrutura de Michell variando o parâmetro *ER*.



Figura 7.7. Topologias obtidas para ER = 10%, 8% e 6%.



. Figura 7.8. Topologias obtidas para ER = 4%, 2% e 1%.



Figura 7.9. Topologias obtidas para ER = 0,50%, 0,25% e 0,20%.



Figura 7.10. Topologias obtidas para ER = 0,15%, 0,10% e 0,05%.

Além disso, com o objetivo de fornecer um embasamento maior para a análise do comportamento da função objetivo, foram calculados seus valores para cem pontos igualmente espaçados variando *ER* de 0,05% a 5,00%. Essa escolha se justifica pelo fato de 0,05% ser o menor valor aceito pelo código utilizado e pela necessidade de se obter uma quantidade considerável de valores para que seja possível analisar o comportamento da estrutura. As relações de compliance e iterações realizadas em função do *ER* são representadas pelos seguintes gráficos de dispersão:



Figura 7.11. Gráfico de dispersão Compliance X ER.



Figura 7.12. Gráfico de dispersão Iteração X ER.

Do gráfico de dispersão anterior, que relaciona o número de iterações necessárias para que a estrutura otimizada atinja a convergência com o parâmetro *ER*, é possível destacar a existência de pontos que destoam do comportamento esperado (*outliers*) que impactam negativamente na análise dos dados. Esses pontos, na realidade, representam valores de *ER* para os quais a estrutura otimizada não obteve convergência. Portanto, a rotina de Matlab® efetua por volta de 25.000 iterações e atribui o valor de 4,6069 Nm para o compliance da estrutura nesses casos. Ou seja, são resultados de instabilidades numéricas. Desse modo, esses pontos serão excluídos da análise realizada. Os valores de

ER com o número de iterações que o programa realiza antes de parar e o compliance atribuído são representados pela tabela abaixo:

ER	Iterações	Compliance
1,25%	25700	4,6069
1,50%	25697	4,6069
1,80%	25702	4,6069
2,75%	25701	4,6069
2,95%	25704	4,6069
3,10%	25699	4,6069
3,15%	25702	4,6069
3,30%	25702	4,6069
3,60%	25699	4,6069
4,30%	25699	4,6069
4,45%	25698	4,6069

Tabela 7.3. Valores sem convergência para a otimização da Estrutura de Michell variando o parâmetro ER.

Os gráficos de dispersão apresentados anteriormente, excluindo os valores de *ER* em que a estrutura não converge, são apresentados em seguida:



Figura 7.13. Gráfico de dispersão Compliance X *ER* (excluindo os pontos em que a estrutura não converge).



Figura 7.14. Gráfico de dispersão Iteração X *ER* (excluindo os pontos em que a estrutura não converge).

Em uma das análises de exemplos de aplicação do BESO realizadas por Huang e Xie (2007), é mencionado o fato de que a extinção abrupta de subdomínios que compõe a estrutura resulta em instabilidades no valor do compliance dela. Desse modo, em conformidade com as topologias apresentadas e os valores de *ER* destacados como os que não possibilitam a convergência da estrutura na faixa de valores selecionada, é possível inferir que essa condição é resultado da situação em que o algoritmo atinge o volume de restrição definido (50%) muito próximo das etapas evolutivas de transição entre o surgimento ou desaparecimento dos furos presentes na estrutura otimizada. Conforme figura 7.10, a topologia apresenta de 1 a 3 furos no volume de restrição, dependo do *ER* selecionado. Daí, a não-convergência nesses valores de *ER*.

A figura 7.13, embora apresente considerável dispersão nos valores nos valores selecionados, apresenta três patamares de compliance bem definidos, com valores de, aproximadamente, 4,5808 Nm, 4,5873 Nm e 4,5960 Nm. De modo geral, é possível observar também que os valores de compliance diminuem com *ER* menores. Como o objetivo da solução do problema de otimização é justamente minimizar essa grandeza, os resultados obtidos estão de acordo com o que é defendido por Huang e Xie (2007). Um valor de *ER* menor, permite que o algoritmo explore mais opções de topologia ao longo do

processo evolutivo. Além disso, da figura 7.14, é possível comprovar a colocação de Yang (1999), uma vez que é possível observar um comportamento onde os menores valores de *ER* demandam muito mais iterações para que a otimização atinja os critérios estabelecidos do que demandam valores de *ER* maiores.

7.1.2. A Estrutura em Forma de Mão-Francesa

7.1.2.1. Análise Inicial

Um outro exemplo selecionado para análise do problema de otimização estrutural visando minimizar o compliance é o problema a estrutura em forma de mão-francesa. O domínio inicial adota consiste em uma chapa quadrada com 100 mm de lado e 1 mm de espessura. Essa chapa possui engastes de 6 mm em um cada um de seus lados, a partir de suas extremidades. Além disso, uma força com intensidade de 100 N é aplicada em um dos cantos do lado oposto da chapa. Essa condição é representada na figura abaixo:


Figura 7.15. Domínio inicial para a Estrutura em forma de mão-francesa.

O domínio do problema é então discretizado em uma malha de elementos finitos formada por 10000 elementos quadrilaterais com quatro nós. As funções de interpolação utilizadas são as mesmas utilizadas no problema anterior. Os elementos finitos são quadrados de lado 1,0 mm. A lógica da numeração dos elementos e eixos utilizada também é a mesma utilizada no exemplo da Estrutura de Michell. Novamente, a análise foi executada utilizando como base a formulação isoparamétrica do elemento plano quadrilateral proposta por Hutton (2004).

As propriedades materiais utilizadas no exemplo são o módulo de elasticidade E = 100 GPa e coeficiente de Poisson v = 0,3. Por sua vez, os parâmetros de otimização correspondem à porcentagem de volume final de 35%, porcentagem de remoção de material a cada iteração de 1% e porcentagem de adição de material máxima equivalente a 5%. Finalmente, o raio do filtro corresponde a 6 mm. Desse modo, o código implementado gerou a seguinte topologia da estrutura:



Figura 7.16. Estrutura em Forma de Mão-Francesa otimizada minimizando o compliance.

De maneira semelhante ao exemplo anterior, após a análise por métodos dos elementos finitos e cálculo dos deslocamentos de cada elemento que compõe a malha, o código utilizado é capaz de calcular as tensões equivalentes de Von Misses, que é representando no seguinte mapa de cores da estrutura otimizada:



Figura 7.17. Mapa de cores da tensão equivalente de Von misses na Estrutura em Forma de Mão-Francesa otimizada.

Mais uma vez, o código também gera os gráficos que relacionam as iterações vigentes com a fração de volume e o compliance da estrutura ao longo do processo evolutivo de otimização. O gráfico resultante para a otimização da estrutura em forma de mão-francesa, para os parâmetros apresentados é o seguinte:



Figura 7.18. Compliance e fração de volume em função da iteração do programa.

Os resultados obtidos para a estrutura analisada estão em conformidade com trabalhos anteriores, validando os resultados obtidos. Labanowski (2004) propôs uma otimização ESO baseada no critério de rigidez e impondo restrições de volumes. Embora o método de otimização e os parâmetros sejam diferentes, aplicando uma mesma restrição de volume e um domínio inicial proporcional, é de se esperar que as estruturas otimizadas sejam semelhantes. Além disso, Sanches (2011) também desenvolveu um trabalho semelhante, utilizando uma otimização ESO para o critério de rigidez, utilizando uma malha de elementos finitos hexagonais. Em ambos os estudos, foram impostas restrições de volume também equivalentes a 35% e as topologias são bem próximas da obtida no presente trabalho.



Figura 7.19. Mão-Francesa obtida por otimização ESO com restrição volumétrica de 35%. Fonte: Labanowski (2004).



Figura 7.20. Mão-Francesa obtida por otimização ESO com restrição volumétrica de 35%, utilizando malha hexagonal. Fonte: Sanches (2011).

O mapa de tensões obtido condiz com o que é esperado. Os maiores valores para a tensão de Von Misses ocorrem nas regiões próximas ao ponto de aplicação da força e nos engastes, decorrente da atuação das reações de equilíbrio. Além disso, os gráficos apresentados na figura 7.18 sugerem um processo de otimização suave, exceto pela variação abrupta no compliance da estrutura por volta da iteração de número 100. Essa situação é consequência da ruptura de umas das barras que se formam ao longo da otimização da estrutura. Mais uma vez, esse comportamento está de acordo com o que é dito por Huang e Xie (2007), onde uma extinção de alguma parte da estrutura ao longo do processo de otimização pode resultar em uma variação abrupta do comportamento da função obtivo. Nesse caso, do compliance. Finalmente, essa grandeza converge para o valor de 1,4215 X 10^-3 Nm e a estrutura atinge a fração de volume de 35% prevista. Os resultados obtidos são resumidos na seguinte tabela:

Tabela 7.4. Resumo de resultados para a otimização inicial da Estrutura em Forma de Mão-Francesa.

Iterações	Compliance	Fração de Volume	
120	1,4215 X 10^-3 Nm	35%	

7.1.2.2. Variando o Parâmetro AR_{max}

Huang e Xie (2007) apresentam a definição do parâmetro Volume Addition Ratio (AR) como sendo a razão entre o número de elementos adicionadas e o número total de elementos presentes no domínio de projeto de uma estrutura. No método BESO, essa razão possui um limite superior, chamado de Prescribed Maximum Volume Addition Ratio (ARmax), que deve ser respeitado em todas as iterações do processo. Desse modo, essa restrição garante que não seja adicionado um número muito alto de elementos por iteração. Huang e Xie (2007) ressaltam que, normalmente, o ARmax deve ser superior a 5%. Isso decorre da necessidade de se preservar o caráter da otimização BESO, isto é, não suprimir a capacidade de adicionar elementos e suas consequentes vantagens. Em conformidade com os autores, Madrid (2016) verificou em seu trabalho que valores pequenos de ARmax exercer influência significativa sobre estruturas que possuem domínio inicial com volume inicial inferior a 100%. Ou seja, para domínios completos, como os utilizados no presente trabalho, a influência na escolha de valores de ARmax deve ser baixa. Desse modo, o objetivo desta seção é verificar a influência desse parâmetro sobre os resultados obtidos no problema. Para isso, no primeiro momento, mantendo todos os outros parâmetros da análise inicial fixos, foram selecionados alguns valores de ARmax e geradas as topologias otimizadas correspondentes:

AR _{max}	Iterações	Compliance x	
		10^-3	
5,0%	120	1,4215	
4,0%	120	1,4215	
3,0%	120	1,4215	

Tabela 7.5. Resumo de resultados para a otimização inicial da Estrutura em Forma de Mão-Francesa variando o parâmetro *AR_{max}*.

2,0%	130	1,4221
1,0%	128	1,4212
0,5%	120	1,4211
0,1%	137	1,4279
0,05%	156	1.4215













Figura 7.22. Topologias obtidas para $AR_{max} = 3\%$ e 2%.



Figura 7.23. Topologias obtidas para $AR_{max} = 1\% e 0.5\%$.







Figura 7.24. Topologias obtidas para $AR_{max} = 0,1\% = 0,05\%$.

Somado a isso, mais uma vez o algoritmo foi utilizado para uma faixa de cem valores de ARmax, tendo seus limites equivalentes a 0,05% e 5,25%. Essa escola se justifica pelo fato de 0,05% ser o mínimo valor que a estrutura do código utilizado suporta e 5,25% ser um valor um pouco maior que o utilizado para a análise inicial. As relações de compliance e iterações realizadas para a otimização completa em função do ARmax são representadas pelos seguintes gráficos de dispersão:



Figura 7.25. Gráfico de dispersão Compliance X ARmax.



Figura 7.26. Gráfico de dispersão Iteração X ARmax.

As topologias obtidas pelos valores de AR_{max} selecionados e ilustradas nas figuras de 7.21 a 7.24 não apresentam diferenças significativas. Grosso modo, são muito parecidas, com a presença ou ausência de alguns elementos em contornos específicos, como a barra superior que delimita a região vazia no meio da estrutura, por exemplo.

Os resultados na figura 7.25 mostram uma estabilidade nos valores de AR_{max} superiores a 2,20%. A partir desse ponto, o compliance obtido para os AR_{max} analisados corresponde a 1,4215 X 10^-3 Nm. Por outro lado, é possível identificar uma pequena

instabilidade nos valores obtidos para a função objetivo quando AR_{max} é menor que 2,20%, com exceção de AR_{max} igual a 0,1%, que resulta no compliance de 1,4279 X 10^-3 Nm. Contudo, a dispersão desses valores é pouco significativa, uma vez que o maior valor de compliance obtido nessa região de AR_{max} é 1,4279 X 10^-3 Nm (AR_{max} equivalente a 0,1%) e o menor valor corresponde a 1,4194 X 10^-3 Nm (AR_{max} igual a 0,6%). A figura 7.26 ilustra um comportamento semelhante para o número de iterações necessárias para completar a otimização. Dessa vez, o AR_{max} que implica em um número de iterações maiores é 0,05%, que demanda 156 iterações.

Os comportamentos obtidos nos dois gráficos de dispersão corroboram o que foi afirmado por Huang e Xie (2007) e Madrid (2016). A escolha de um valor para *AR_{max}* próximo de 5% é adequada, uma vez que se afasta da região de comportamento pouco mais instável, evitando que se possa obter uma estrutura com compliance levemente mais alto. E como observado, mesmo essa região em que os valores de compliance e número de iterações variam, essa variação é pouco significativa em termos práticos. Pelo menos no caso adotado. Possivelmente, valores de *AR_{max}* pequenos teriam uma influência maior para um domínio inicial diferente do volume de 100%, como mencionado pela autora. Contudo, esse caso não é objeto de estudo do presente trabalho.

7.2. Aplicação do Método SIMP

7.2.1. O Problema da Estrutura de Michell

7.2.1.1. Análise Inicial

Na presente seção, o Problema da Estrutura de Michell será resolvido novamente, utilizando um algoritmo base no método SIMP dessa vez. Desse modo, as mesmas condições representadas pela figura 7.1 são impostas. Além disso, a força aplicada possua a mesma intensidade de 10 kN e a espessura também é a de 0,001 m.

O domínio do problema é mais uma vez discretizado em uma malha composta por 1250 elementos quadrilaterais de lado 0,2 m. Contudo, o esquema de numeração dos elementos e graus de liberdade é diferente. Como o programa SIMP utilizado é baseado no desenvolvido por Sigmund (2001), o esquema de numeração original será preservado. Os elementos e nós são numerados em colunas, de cima para baixo e da esquerda para a direita. A seguinte figura ilustra o esquema adotado:



Figura 7.27. Numeração de elementos e eixos no domínio da Estrutura de Michell para o código SIMP.

As propriedades materiais adotadas também são as mesmas para o caso da solução pelo método BESO. Ou seja, o módulo de elasticidade E = 100 GPa e coeficiente de Poisson v = 0,3.

Os parâmetros adotados para a otimização na análise inicial foram uma fração de volume final de 50%, parâmetro de penalidade igual a 3 e a dimensão do filtro (divido pela dimensão de um elemento) equivale a 1,5. Desse modo, a estrutura otimizada obtida pelo método SIMP com esses parâmetros é representada na seguinte figura:



Figura 7.28. Estrutura de Michell otimizada minimizando o compliance utilizando o método SIMP.

A topologia obtida para o método SIMP é muito semelhante àquela obtida para o método BESO, como era de se esperar. Esse resultado, mais uma vez valida da eficácia dos algoritmos utilizados no trabalho. Além disso, foi implementado no código um gráfico para ilustrar a evolução da otimização, isto é, a função objetivo (o compliance da estrutura) e a fração de volume em função da iteração vigente do programa. Embora a dinâmica de funcionamento do algoritmo seja diferente da do BESO, uma vez que no caso do SIMP já na primeira iteração a estrutura possui a fração final de volume e os elementos vão tendo suas densidades alteradas ao longo do processo, o gráfico torna a evolução da otimização mais clara. Sendo assim, foi gerado o seguinte gráfico:



Figura 7.29. Compliance e fração de volume em função da iteração do programa.

O gráfico gerado é bastante ilustrativo para o método SIMP. No programa utilizado, a partir da primeira iteração a estrutura já está configurada na fração de volume final e, de acordo com as analises de elementos finitos realizadas em cada iteração, as densidades dos elementos se ajustam de forma a minimizar o compliance até que o critério de convergência seja atingido.

Após a execução do código, os resultados obtidos são a fração de volume de 50% e o compliance de 9,41 Nm, alcançados na iteração 34. Naturalmente, os valores seriam diferentes, uma vez que a dinâmica dos métodos, modelos numéricos e parâmetros adotados são diferentes. Contudo, esse valor para o compliance da estrutura otimizada é relativamente próximo ao obtido para o método BESO (4,59 Nm), o que reforça a consistência dos algoritmos. Além disso, os resultados obtidos por Huang e Xie (2007) em seus exemplos também apresentam um valor de compliance maior para os algoritmos SIMP. Os autores mencionam a possibilidade de isso ocorrer graças à energia de deformação superestima para os elementos com densidades intermediarias presentes na topologia SIMP. Os resultados são apresentados na tabela a seguir, de forma resumida:

Tabela 7.6. Resumo de resultados para a otimização inicial da Estrutura de Michell utilizando o método SIMP.

lterações	Compliance	Fração de Volume	
34	9,41 Nm	50%	

7.2.1.2. Variando o Parâmetro de Penalidade p

O parâmetro de penalidade p consiste em um expoente utilizado na equação que representa o modelo de material utilizado no método SIMP (e em outros métodos que envolvam interpolação de material). Segundo Aremu et al. (2010), o parâmetro é o responsável por suprimir a contribuição de elementos com densidades intermedirias (cinzas) na topologia otimizada. Sendo assim, a presente seção tem como objetivo investigar a influencia da variação dos valores de p na solução do problema da Estrutura de Michell. Para isso, foram geradas as topologias resultantes para valores de p equivalentes a 1, 2, 3 (valor do problema inicial) e 4. Elas são apresentadas na seguinte figura:



P = 1

P = 2

Figura 7.30. Topologia obtida para a Estrutura de Michell com p = 1 e 2.



Figura 7.31. Topologia obtida para a Estrutura de Michell com p = 3 e 4.

Os números de iterações necessárias e valores de compliance para os valores de p são apresentados na tabela a seguir:

р	Iterações	Compliance
1	14	8,5316 Nm
2	25	9,3188 Nm
3	34	9,4100 Nm
4	66	9,5085 Nm

Tabela 7.7. Resumo de resultados para a otimização inicial da Estrutura de Michell utilizando o método SIMP e variando *p*.

De acordo com as topologias obtidas na figura 7.31, é perceptível que quanto maior o valor de *p*, mais suprimida é a presença de elemento com densidades intermediárias (cinzas) nas estruturas. Portanto, essa condição reforça o que foi dito por Aremu et al. (2010). Além disso, a tabela 7.7 apresenta uma condição em que quanto maior o valor de *p*, mais iterações são necessárias para se alcançar a topologia otimizada. Essa situação pode ser um reflexo da supressão da influencia de elementos cinzas na estrutura, obrigando o algoritmo SIMP a adaptar a topologia de uma forma mais próxima de métodos de otimização binários, como o BESO. Finalmente, é possível observar também que o compliance cresce com o aumento do valor de *p*. Esse comportamento pode ser relacionado com o módulo de elasticidade dos elementos que, de acordo com a equação 5.17, quanto maior o parâmetro de penalização, menor o valor dessa propriedade. Consequentemente, maior será o compliance obtido para a estrutura.

7.2.2. A Estrutura em Forma de Mão-Francesa

7.2.2.1. Análise Inicial

Novamente, o problema de otimização topológica da Estrutura em forma de Mão-Francesa visando minimizar o compliance da estrutura. Dessa vez, ele será resolvido utilizando o código baseado no método SIMP. Mais uma vez, as condições geométricas e as condições de contorno representadas pela figura 7.15 serão adotadas. A força aplicada e a espessura da chapa que corresponde ao domínio inicial correspondem a, novamente, 100 N e 1 mm de espessura.

A discretização da malha adotada para o método BESO é também repetida aqui. Ou seja, o domínio é composto 10000 elementos quadrilaterais de lado 1,0 mm. A lógica de

numeração de elementos e eixos é a mesma adotada para o problema da Estrutura de Michell resolvido pelo método SIMP (figura 7.27), uma vez que o algoritmo utilizado é o mesmo.

Com relação as propriedades materiais, foram adotados o e coeficiente de Poisson v = 0,3 e o módulo de elasticidade E = 1 GPa. A escolha de um valor diferente para o módulo de elasticidade com relação ao problema resolvido com o método BESO é motivado pelo fato de o código não ter reagido muito bem a combinação de parâmetros adotada.

Finalmente, os parâmetros selecionados para a otimização nessa análise inicial são parâmetro de penalidade igual a 3, fração de volume final igual a 35% e dimensão do filtro (divido pela dimensão de um elemento) equivale a 1,5. Com isso, a topologia otimizada pelo método SIMP é a seguinte:



Figura 7.32. Estrutura em forma de Mão-Francesa otimizada minimizando o compliance utilizando o método SIMP.

A topologia obtida é semelhante a estrutura gerada pela otimização BESO, com exceção de duas novas regiões vazias que surgem na composição da estrutura. Contudo,

o resultado valida mais uma vez a eficiência dos algoritmos adotados no presente trabalho. Novamente, foi gerado um gráfico que ilustra a evolução do processo de otimização da topologia. Nele é possível observar os valores da fração de volume e compliance alcançados em cada iteração do programa:



Figura 7.33. Compliance e fração de volume em função da iteração do programa.

O gráfico ilustra a dificuldade mencionada anteriormente enfrentada pelo algoritmo para otimizar a estrutura nas condições impostas. O número de iterações em que a fração volumétrica o compliance variaram muito pouco foi alto. Além disso, a fração de volume de 35% não foi alcançada. Ao final da otimização, o compliance obtido foi 0,2720 Nm, a fração de volume foi equivalente a 38,3% e foram necessárias 172 iterações para atingir a convergência. Essas informações são resumidas na seguinte tabela:

Tabela 7.8. Resumo de resultados para a otimização inicial da Estrutura em forma de Mão-Francesa utilizando o método SIMP.

Iterações	Compliance	Fração de Volume	
172	0,2720 Nm	38,3%	

7.2.2.2. Variando a Dimensão do Filtro rmin

Como já mencionado ao longo do trabalho, os esquemas de filtros aplicados nos algoritmos são ferramentas importantes para contornar problemas presentes nos métodos BESO e SIMP, como a dependência de malha e o problema do tabuleiro de xadrez. O esquema utilizado no algoritmo SIMP utiliza é o proposto por (Sigmund 1994, 1997). Nesse código desenvolvido por Sigmund (2001), o parâmetro *rmin* representa a razão entre a dimensão do filtro e a dimensão dos elementos finitos. Desse modo, a presente seção se propõe a selecionar alguns valores para *rmin*, mantendo fixo todos os outros parâmetros do problema de otimização, e analisar os resultados dessa variação nas topologias e seus resultados obtidos. Os valores de *rmin* com os respectivos valores de compliance, iterações necessárias e fração de volume final para as estruturas são listados na tabela a seguir:

rmin	Iterações	Compliance	Fração de Volume
0,1	75	0,3266	29,0%
0,5	75	0,3266	29,0%
1,0	75	0,3266	29,0%
1,5	172	0,2720	38,3%
2,0	151	0,2703	39,0%
2,5	54	0,2699	39,7%
3,0	50	0,2698	40,1%
3,5	44	0,2696	40,7%
4,0	43	0,2693	41,2%

Tabela 7.9. Resumo de resultados para a otimização inicial da Estrutura em forma de Mão-Francesa utilizando o método SIMP e variando *rmin*.

Além disso, as topologias otimizadas obtidas para os valores de *rmin* selecionados são ilustrados nas seguintes imagens:



Figura 7.34. Topologias obtidas para rmin = 0,1, 0,5 e 1,0.



Figura 7.35. Topologias obtidas para *rmin* = 1,5, 2,0 e 2,5.



Figura 7.36. Topologias obtidas para *rmin* = 3,0, 3,5 e 4,0.

Inicialmente, é possível observar que quando *rmin* é equivalente a 0,1, 0,5 ou 1,0, os resultados e as topologias obtidas são idênticos. Isso é consequência do fato de o esquema de filtro utilizado ser seu efeito anulado para *rmin* menores ou igual a 1. Essa condição significa que significa que a dimensão do filtro é igual ou menor a dimensão de um elemento finito, de acordo com a definição de *rmin*. De acordo com Sigmund (2001), um valor de *rmin* menor que 1 implica que as sensibilidades filtradas passam a ser equivalentes as não-filtradas. Dessa forma, valor de *rmin* entre 0 e 1 deve resultar em topologias e propriedades idênticas, mantidos os outros parâmetros fixos. Além disso, é possível verificar a ineficiência do filtro nessa condição uma vez que as topologias otimizadas passam a apresentar o problema do tabuleiro de xadrez, observado nas topologias registradas na figura 7.34.

O esquema de filtro atua suavizando o problema numérico enfrentado durante a otimização do problema. Dessa forma, selecionando um *rmin* maior que 1, as propriedades e topologias obtidas passam a variar em função do parâmetro. Isso pode ser observado nos resultados apresentados na tabela 7.9 e nas figuras 7.35 e 7.36. A presença do problema do tabuleiro de xadrez desaparece nessas estruturas. É possível observar que, de acordo com os resultados obtidos, a partir do valor de rmin que garante a funcionalidade do filtro, o número de iterações para se obter a convergência da estrutura diminui com o aumento do valor de *rmin*. Além disso, é possível observar uma redução do compliance da estrutura com o aumento da dimensão do filtro. Entretanto, é notável que, com o aumento de rmin, a fração de volume final da estrutura aumenta, afastando-se dos 35% selecionado na definição do problema. De modo geral, um acréscimo de volume implica em maior rigidez e, consequentemente, menor compliance. Dessa forma, embora o filtro atue suavizando os gradientes nos cálculos de sensibilidade, não é possível concluir qual é o valor de rmin mais indicado para a solução do problema. Contudo, a análise corrobora a necessidade da aplicação do filtro no intuito de se obter topologias que não suscetíveis a rigidez artificial imposta pelo problema do tabuleiro de xadrez.

Capítulo 8

Conclusões

O trabalho realizado possuiu como objetivo estabelecer um comparativo entre dois métodos de otimização topológica muito utilizados na área de mecânica estrutural, analisando as suas particularidades através da aplicação deles em problemas clássicos selecionados. Os métodos são o Bi-directional Evolutionary Structural Optimization (BESO) e o Solid Isotropic Material with Penalization Method (SIMP). Grosso modo, a partir da revisão bibliográfica realizada, é possível verificar que os métodos BESO e SIMP possuem suas vantagens e desvantagens entre eles, o que envolve principalmente variáveis relacionada a custos computacionais e complexidade de implementação dos métodos.

Diante das informações presentes nos trabalhos revisados, o método BESO se apresenta como o de mais simples aplicação e interpretação de resultados, considerando a ausência de elementos finitos com densidade intermediária (elementos cinzas). Em contrapartida, o BESO nem sempre converge para a configuração de topologia ótima, sendo necessário definir claramente os critérios de parada do método. Sem contar com a dependência de malha enfrentada pelo método, o que pode representar uma instabilidade numérica para o método.

Por sua vez, ainda de acordo com os trabalhos pesquisados, o método SIMP possui maior robustez matemática por tratar de variáveis continuas e possibilita a obtenção da topologia otimizada de forma mais intuitiva e clara. Todavia, alguns resultados apresentados indicam que o método demanda um maior número de interações para alcançar o estado ótimo em comparação a outros métodos. Além disso, a presença de elementos com densidade intermediria torna um pouco mais complicada a interpretação dos resultados obtidos com a aplicação do SIMP.

As produções bibliográficas utilizadas também mostram a necessidade da aplicação dos métodos dos elementos finitos da forma mais adequada para cada problema analisado. Os tipos de elementos e o grau de refinamento da malha representam fatores importantes para o sucesso da análise realizada.

Os desenvolvimentos teóricos apresentados para os dois métodos demonstram as diferentes abordagens adotadas nos métodos para as variáveis. Enquanto o método BESO utiliza variáveis binárias para a densidade dos elementos, o método SIMP relaxa esse

problema possibilitando a presença de elementos com densidades intermedirias na topologia final. Além disso, o processo de atualização das variáveis no método BESO é heurístico, enquanto o processo adotado para o método SIMP é sustentando por otimizadores numéricos. Considerando essas condições, o método SIMP apresenta uma gama maior de possibilidades de solução para um problema especifico, quando comparado com o método BESO. Por sua vez, o método BESO apresenta soluções possivelmente mais simples.

As topologias obtidas para os exemplos selecionados estão de acordo com o esperado de acordo com outros trabalhos realizados. Com a aplicação dos filtros, essas soluções se tornam muito semelhantes. Contudo, não é viável declarar um método melhor que o outro. No caso do problema da Estrutura de Michell, que tanto as propriedades dimensionais quanto as geométricas são iguais, o compliance obtido para o método BESO é menor que a do método SIMP. Entretanto, o método BESO demanda um maior número de iterações para atingir a convergência do que o SIMP. A preferência pelo uso de um método ou outro deve ser avaliada de acordo com o que se espera no âmbito do valor da função objetivo e velocidade do algoritmo.

Com relação a etapa de variação de parâmetros de otimização para os dois métodos, foi possível comprovar algumas informações apresentadas por outros autores. No caso da otimização BESO, a escolha de um parâmetro *ER* de valor pequeno mostra-se interessante por permitir que o algoritmo de otimização explore diferente arranjos de topologias durante o processo de evolução e a escolha valor de *AR_{max}* mostrou pouco influência sobre os resultados finais para um domínio inicial de volume completo. No caso do método SIMP, foi possível comprovar a influencia da escolha do fator de penalidade na supressão do efeito de elementos com densidades intermediarias e foi possível testar a eficiência do esquema de filtro presente no algoritmo utilizado.

Finalmente, é importante salientar a dificuldade em se declarar um método mais eficiente que o outro com os resultados obtidos neste trabalho. Existe um grande número de fatores que influenciam nos resultados obtidos de um estudo de otimização topológica, como os modelos materiais e matemáticos. Além disso, em um projeto, nem sempre apenas os valores de função objetivo alcançados devem ser considerados, sendo necessário também avaliar a viabilidade prática da obtenção das estruturas obtidas. Portanto, é necessário sempre garantir uma boa colocação do problema a ser analisado e explorar as mais diversas possibilidades de métodos de otimização topológica existentes, que não se restringem aos métodos BESO e SIMP.

Referências Bibliográficas

- Bittencourt, Marco Lúcio, Análise Computacional de Estruturas: com Aplicação do Método dos Elementos Finitos / Marco Lúcio Bittencourt. – Campinas, SP: Editora Unicamp, 2010.
- Alves Filho, Avelino, Elementos Finitos a Base da Tecnologia CAE, Editora Érica, 5º edição, 2007.
- Xie, Y.M. and Steven, G.P., Evolutionary Structural Optimization. Springer, 1997.

Lopez, R.H. e L. F.; Miguel. Introdução a Otimização Estrutural, 7 p. 2013.

- Evolutionary Topology Optimization of Continuum Structures: Methods and Applications X. Huang and Y.M. Xie 2010.
- QUISPE RODRÍGUEZ, Sergio. Otimização Topológica Multiobjetivo de Estruturas submetidas a carregamentos Termo-Mecânicos. 2015. 117p. Tese (Mestrado). Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, Campinas.

X. Huang, Y.M. Xie / Finite Elements in Analysis and Design 43 (2007) 1039–1049

- Rozvany, G.I.N. Struct Multidisc Optim (2009) 37: 217. https://doi.org/10.1007/s00158-007-0217-0
- SILVA, Julio Antonio Beltrami da. Investigação de um metodo evolucionario de otimização estrutural. 2001. 107p. Dissertação (mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Mecanica, Campinas, SP. Disponível em: <http://www.repositorio.unicamp.br/handle/REPOSIP/265191>. Acesso em: 16 ago. 2019.
- Sigmund, Ole. (2001). Sigmund, O.: A 99 Line Topology Optimization Code Written in MATLAB. Structural and Multidisciplinary Optimization 21, 120-127. Structural and Multidisciplinary Optimization. 21. 120-127. 10.1007/s001580050176.

- Zhou, Xiangyang & Chen, Liping & Huang, Zhengdong. (2009). The SIMP-SRV Method for Stiffness Topology Optimization of Continuum Structures. 7.
- Xia, Liang & Xia, Qi & Huang, X. & Xie, Yi. (2016). Bi-directional Evolutionary Structural Optimization on Advanced Structures and Materials: A Comprehensive Review. Archives of Computational Methods in Engineering. 10.1007/s11831-016-9203-2.
- Huang, X. & Xie, Yi. (2007). Convergent and mesh-independent solutions for the bidirectional evolutionary structural optimization method. Finite Elements in Analysis and Design. 1039-1049. 10.1016/j.finel.2007.06.006.
- Pereira, E. L. P., Bono, G., Bono, G. F. F., Otimização Topológica de Sistema de Contraventamento em Edificios Altos. Revista IPT | Tecnologia e Inovação, v.2, n.10, abr., 2019.
- PAULINO, Daniele Melo Santos. Otimização topológica de estruturas planas considerando comportamento não linear geométrico. 2019. Dissertação (Mestrado em Estruturas) Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2019. Disponível em: http://www.teses.usp.br/teses/disponiveis/18/18134/tde-06092019-125605/>. Acesso em: 2019-09-01.
- He, Dan & Liu, Shutian. (2008). BESO method for topology optimization of structures with high efficiency of heat dissipation. International Journal for Simulation and Multidisciplinary Design Optimization. 2. 43-48. 10.1051/smdo:2008005.

Hemp, W. S.: 1973, Optimum Structures. Claredon Press.

- Aremu, Adedeji & Ashcroft, Ian & Hague, Richard & Wildman, Ricky & Tuck, Christopher. (2010). Suitability of SIMP and BESO topology optimization algorithms for additive manufacture. 21st Annual International Solid Freeform Fabrication Symposium - An Additive Manufacturing Conference, SFF 2010. 679-692.
- X. Wang, Z. Lin and R. Xia, "SIMP based topology optimization of a folding wing with mixed design variables," Proceedings of the 2013 IEEE 17th International Conference on

Computer Supported Cooperative Work in Design (CSCWD), Whistler, BC, 2013, pp. 417-421. doi: 10.1109/CSCWD.2013.6580999

- Reddy, J. N., Introduction To The Finite Element Method, An 3rd Ed. McGraw-Hill Higher Education, 2005.
- NETO, EUCLIDES DE MESQUITA, Resistência dos Materiais Notas de Aula Cap. 4 Deformação, 2019.
- Hutton, D.V., Fundamental of Finite Element Analysis. McGraw Hill Companies, New York, 2004.
- Chu, D.N., Xie, Y.M., Hira, A. and Steven, G.P. Evolutionary structural optimization for problems with stiffness constraints. Finite Elements in Analysis and Design 21: 239– 51, 1996.
- Jog, C.S. and Harber, R.B. Stability of finite element models for distributed-parameter optimization and topology design. Comput. Meth. Appl. Mech. Engng. 130: 1951–65, 1996.
- SANCHES, Renato Picelli. Otimização estrutural evolucionária usando malhas hexagonais. 2011. 121 p. Dissertação (mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Mecânica, Campinas, SP. Disponível em: http://www.repositorio.unicamp.br/handle/REPOSIP/265152>. Acesso em: 19 maio. 2020.
- Bendsøe, M.P. and Sigmund, O. Topology Optimization: Theory, Method and Application. Berlin: Springer, 2003.
- Siva Rama Krishna L., Mahesh N., Sateesh N. (2017). Topology optimization using solid isotropic material with penalization technique for additive manufacturing. Materials Today: Proceedings, 4 (2), pp. 1414-1422.

- Bendsøe, M.P. (1995). Optimization of Structural Topology, Shape and Material. Berlin: Springer.
- Yang, Xiaoying (1999) Bi-directional evolutionary method for stiffness and displacement optimisation. Research Master thesis, Victoria University of Technology.
- Labanowski, A., Análise comparativa de métodos de otimização topológica em elasticidade 2D e 3D. Dissertação (mestrado), Universidade Federal de Santa Catarina, 2004.
- PEREZ MADRID, Claudia Marcela. Bi-directional evolutionary topology optimization of compliant mechanisms design using a multi-criteria approach = Otimização topológica bidiecional evolucionária para o projeto de mecanismos flexíveis usando um enfoque multi-critério. 2016. 1 recurso online (98 p.). Dissertação (mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Mecânica, Campinas, SP. Disponível em: http://www.repositorio.unicamp.br/handle/REPOSIP/320788>. Acesso em: 17 maio. 2020.

Anexos

Rotina desenvolvida em Matlab® para a solução do problema da Estrutura de Michell, utilizando o método dos elementos finitos e aplicando o algoritmo de otimização BESO (adaptação do código desenvolvido por Tainan Calixto e Renato Pavanello).

clear all; close all; clc;

% Vinicius Lourenço da Silva Daldon

% RA: 178275

% EM 919 - Trabalho de graduação II

- % Título: Comparação e implementação dos métodos de otimização topológica estrutural evolucionária BESO e baseada em programação matemática SIMP
- % O presente programa consiste em uma adaptação do código desenvolvido por

% Tainan Calixto e Renato Pavanello (DMC FEM - UNICAMP)

%% Dimensionando a malha:

Lx = 10; %% Dimensão em x da malha [m] Ly = 5; %% Dimensão em y da malha [m] Lz = 1e-3; %% Espessura [m]

Nx = 50; %% Número de elementos em x Ny = 25; %% Número de elementos em y

 $r_{ij} = 20,$ r_{ij} $r_{$

%% Propriedades do material:

```
E = 100e9; %% Módulo de elasticidade [Pa]
v = 0.3; %% Coeficiente de Poisson
```

%% Módulo da força aplicada:

F = 10000; %% Módulo da força aplicada [N]

%% Parametros do programa original:

% Optimization parameter

Vfinal = 0.5; % Final Volume Fraction / Porcentagem de volume final

ER = 0.010; % Evolutionary Remotion parameter / Porcentagem de remoção de material a cada iteração

```
AR_max = 0.05; % Evolutionary Addition parameter/ Porcentagem de adição de material máximo (Addition ratio)
```

r_min = 6e-1; % Filter radios / Raio mínimo de ponderação do filtro

tau = 0.01/100; % Convergence parameter / Precisão do critério de convergência (Mean Compliance)

N = 5; % Number of iteration for convergence / Numero de iterações para o cálculo do erro

plot_ite=1; % plot index / Plot da estrutura em cada iteração [0-n,1-s]

emat = [E v Lz]; nel = Nx*Ny; nnode = (Nx+1)*(Ny+1); elem_list = 1:nel; node_list = 1:nnode; ndof = 2; sdof = nnode*ndof;

% Initial values variables / Inicializando as variáveis errorS = 0.9999; V = 1.0d0; mat = ones((Nx*Ny),1); alpha_elem = zeros((Nx*Ny),1); ite = 0; cond_volume = 0;

%% Construção da malha:

 nos_linha = Nx + 1;
 %% Número de nós por linha

 nos_coluna = Ny + 1;
 %% Número de nós por coluna

 nos_X_Y = [nos_linha nos_coluna];
 %% Vetor de relação nó-linha e nó coluna

 nos_total = nos_linha * nos_coluna;
 %% Número de nós na malha

 a = Lx / Nx;
 %% Dimensão em x do elemento [m]

 b = Ly / Ny;
 %% Dimensão em y do elemento [m]

% Matriz com os coordenadas dos nós na malha, onde i = nó, j1 = coord_x_nó % e j2 = coord_y_nó:

```
coord_nos(nos_total,3) = zeros; %% Inicializando a matriz com zeros
```

```
delta_x = 0; %% Contador inicial da coordenada em x
delta_y = 0; %% Contador inicial da coordenada em y
```

```
for i = 1:(nos_linha * nos_coluna)
```

```
coord_nos(i,1) = i;
coord_nos(i,2) = delta_x;
coord_nos(i,3) = delta_y;
delta_x = delta_x + a;
if (mod(i,nos_linha)==0)
    delta_x = 0;
    delta_y = delta_y + b;
end
```

end

% Matriz com a relação entre elementos e nós que incidem neles, onde i = % elemento, j1 = nó_1, j2 = nó_2, j3 = nó_3 e j4 = nó_4:

nos_elem(Nx*Ny,5) = zeros; %% Inicializando a matriz com zeros

for $i = 1:(Nx^*Ny)$

nos_elem(i,1) = i; nos_elem(i,2) = i + fix((i-1)/Nx); %% Acréscimo na relação elemento-nós de acordo com a linha completa no loop nos_elem(i,3) = nos_elem(i,2) + 1; nos_elem(i,5) = nos_elem(i,2) + nos_linha; nos_elem(i,4) = nos_elem(i,2) + 1 + nos_linha;

end

% Matriz com a relação de nós com graus de liberdade, onde i = nó, j1 = gl_1 % e j2 = gl_2:

```
nos_gls(nos_linha*nos_coluna,2) = zeros; %% Inicializando a matriz com zeros
```

```
for i = 1:(nos_linha*nos_coluna)
```

nos_gls(i,1) = i * 2 - 1; nos_gls(i,2) = i * 2;

end

%% Aplicação das condições de contorno na estrutura% Exemplo do programa (engastado na extremidade esquerda

fixeddofs = [1 2 101 102]; alldofs = 1:2*nos_total; freedofs = setdiff(alldofs,fixeddofs);

%% Load aplications / Aplicação do carregamento na estrutura

 $Fg = sparse(2*nos_total, 1);$ $U = zeros(2*nos_total, 1);$ force_node = 26; $Fg(2*force_node) = -F;$

%% Construção da matriz de rigidez do elemento: %%

```
coord_nos_elemento = [coord_nos(nos_elem(elemento,2),2), coord_nos(nos_elem(elemento,2),3); coord_nos(nos_elem(elemento,3),2),

coord_nos(nos_elem(elemento,3),3); coord_nos(nos_elem(elemento,4),2), coord_nos(nos_elem(elemento,4),3);

coord_nos(nos_elem(elemento,5),2), coord_nos(nos_elem(elemento,5),3)]; %% Vetor que localiza as coordenadas dos nós que

compõem o elemento analisado
```

% Construção das funções de forma

Pontos_Gauss = [0.577350269189626 0.577350269189626; 0.577350269189626 -0.577350269189626; -0.577350269189626; -0.577350269189626 -0.577350269189626]; %% Coordenadas utilizadas para a integração

for i = 1:4

```
r = Pontos_Gauss(i,1);
s = Pontos_Gauss(i,2);
% Derivadas das funções de forma em:
% I) relação a r:
N1r = (1/4) * (s - 1);
N2r = (1/4) * (1 - s);
N3r = (1/4) * (1 + s);
N4r = (1/4) * (1 + s) * (-1);
% I) relação a s:
N1s = (1/4) * (r - 1);
N2s = (1/4) * (1 + r) * (-1);
N3s = (1/4) * (1 + r);
N4s = (1/4) * (1 - r);
% Determinando os termos da matriz jacobiana:
J11 = coord_nos\_elemento(1,1) * N1r + coord_nos\_elemento(2,1) * N2r + coord_nos\_elemento(3,1) * N3r + coord_nos\_elemento(4,1) + (N3r + coord_nos\_element
               * N4r;
J21 = coord_nos_elemento(1,1) * N1s + coord_nos_elemento(2,1) * N2s + coord_nos_elemento(3,1) * N3s +
               coord_nos_elemento(4,1) * N4s;
J12 = coord_nos_elemento(1,2) * N1r + coord_nos_elemento(2,2) * N2r + coord_nos_elemento(3,2) * N3r + coord_nos_elemento(4,2)
               * N4r;
J22 = coord_nos_elemento(1,2) * N1s + coord_nos_elemento(2,2) * N2s + coord_nos_elemento(3,2) * N3s +
               coord_nos_elemento(4,2) * N4s;
J = [J11 J12; J21 J22];
% Matriz auxiliar G:
G = (1 / det(J)) * [J22 0 - J12 0; 0 - J21 0 J11; -J21 J22 J11 - J12];
```

% Matriz auxiliar P:

$$\begin{split} \mathsf{P} &= [\mathsf{N1r} \ \mathsf{0} \ \mathsf{N2r} \ \mathsf{0} \ \mathsf{N3r} \ \mathsf{0} \ \mathsf{N4r} \ \mathsf{0}; \ \mathsf{0} \ \mathsf{N1r} \ \mathsf{0} \ \mathsf{N2r} \ \mathsf{0} \ \mathsf{N3r} \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0}; \ \mathsf{0} \ \mathsf{N1s} \ \mathsf{0} \ \mathsf{N2s} \ \mathsf{0} \ \mathsf{N3s} \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0}; \ \mathsf{0} \ \mathsf{N1s} \ \mathsf{0} \ \mathsf{N2s} \ \mathsf{0} \ \mathsf{N3s} \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0}; \ \mathsf{0} \ \mathsf{N1s} \ \mathsf{0} \ \mathsf{N2s} \ \mathsf{0} \ \mathsf{N3s} \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0}; \ \mathsf{0} \ \mathsf{N1s} \ \mathsf{0} \ \mathsf{N2s} \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0}; \ \mathsf{0} \ \mathsf{N1s} \ \mathsf{0} \ \mathsf{N2s} \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0}; \ \mathsf{0} \ \mathsf{N1s} \ \mathsf{0} \ \mathsf{N2s} \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0}; \ \mathsf{0} \ \mathsf{N1s} \ \mathsf{0} \ \mathsf{N2s} \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0}; \ \mathsf{0} \ \mathsf{N1s} \ \mathsf{0} \ \mathsf{N2s} \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0}; \ \mathsf{0} \ \mathsf{N1s} \ \mathsf{0} \ \mathsf{N2s} \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0}; \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0}; \ \mathsf{0} \ \mathsf{N1s} \ \mathsf{0} \ \mathsf{N2s} \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0}; \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0}; \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0}; \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0} \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0}; \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0} \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0} \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0} \ \mathsf{0} \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0} \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0} \ \mathsf{0} \ \mathsf{0} \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0} \ \mathsf{0}$$

end

%% Montagem da matriz global:

ID = nos_gls'; loc = zeros (8,(Nx*Ny));

```
for i=1:(Nx*Ny)
```

```
loc (1,i) = nos_gls(nos_elem(i,2),1);
loc (2,i) = nos_gls(nos_elem(i,2),2);
loc (3,i) = nos_gls(nos_elem(i,3),1);
loc (4,i) = nos_gls(nos_elem(i,3),2);
loc (7,i) = nos_gls(nos_elem(i,5),1);
loc (8,i) = nos_gls(nos_elem(i,5),2);
loc (5,i) = nos_gls(nos_elem(i,4),1);
loc (6,i) = nos_gls(nos_elem(i,4),2);
```

```
end
```

```
      I = reshape(repmat(loc,8,1),(Nx*Ny)*64,1);
      %% Matriz que repete os graus de liberdade associados aos elementos (8x)

      Jota = kron(loc(:),ones(8,1));
      %% Organização de I, deixando DOF's iguais em sequência

      Kg = kron(mat,ones(64,1)).*repmat(Ke(:),(Nx*Ny),1);
      %% Inserindo as características de densidade e de K elementar

      K = sparse(I,Jota,Kg);
      %% Montagem de K

      %% Ajustes:
      %
```

```
nnode = nos_total;
Conectivity = nos_elem;
nel = Nx*Ny;
Nodes = coord_nos;
Kg = K;
id = ID;
nelx = Nx;
nely = Ny;
```

%% Programa BESO PAVA

% assemblage of Filter matrix

% Montagem da Matriz de Filtro (alpha_elem = H*alpha_nodal)

[noconnect] = nodalconnect(nnode,Conectivity); % Matriz Conectividade Nodal

[center,elarea] = centroids(nel,Nodes(:,2:3),Conectivity);

[H] = filterH(Conectivity,elem_list,center,Nodes(:,2:3),r_min);

% assemblage of Filter matrix

% Montagem da Matriz de Filtro (alpha_elem = H*alpha_nodal)

[noconnect] = nodalconnect(nnode,Conectivity); % Matriz Conectividade Nodal

[center,elarea] = centroids(nel,Nodes(:,2:3),Conectivity);

```
[H] = filterH(Conectivity,elem_list,center,Nodes(:,2:3),r_min);
```

```
%%%%%%%
% Starting BESO iterations
% Inicio do processo de otimização - BESO
%%%%%%%
tic
while (cond_volume == 0) || (errorS >= tau)
 ite = ite+1; mat_old = mat;
 % solve Equation System / Solução da equação do sistema
 U(freedofs) = Kg(freedofs,freedofs)\Fg(freedofs);
 U(fixeddofs)= 0;
 % Sensibility analysis / Cálculo da sensibilidade (Compliance)
 for iel=elem_list
   nd = Conectivity(iel,2:5);
   index = [nd(1)*2-1 nd(1)*2 nd(2)*2-1 nd(2)*2 nd(3)*2-1 nd(3)*2 nd(4)*2-1 nd(4)*2];
   if mat(iel)==1; alpha_elem(iel) = 0.5*U(index)'*Ke*U(index); % Elemento cheio
   else alpha_elem(iel) = 0;
                                         % Elemento vazio
    end
  end
  % Filter / Filtro Numérico
 [alphaS] = filterBEFSOH(nel,nnode,noconnect,elem_list,node_list,1,alpha_elem,H);
  % Estabilization of Evolutionary process / Estabilizando o Processo Evolucionário
 if ite==1; alphaS_old = alphaS;
  else
    alphaS = (1/2)^{*}(alphaS_old + alphaS);
   alphaS_old = alphaS;
  end
 % Ordering mean compliance / Ordenando a Mean Compliance
 [alphaS_ord,ord] = sort(alphaS,1,'descend');
 % Neste volume calculation / Cálculo do volume desejado
 if (V(ite) - Vfinal >= ER*V(ite)); V(ite+1)=V(ite)*(1-ER);
  elseif (V(ite)-Vfinal <= -ER*V(ite)); V(ite+1)=V(ite)*(1+ER);
  elseif (abs(V(ite)-Vfinal) < ER*V(ite));
    V(ite+1)=Vfinal; cond_volume = 1;
  end
```

% Choice of elements to be remouved

% Escolha dos elementos que serão retirados

```
%%%%%%%
  % Linha limite de remoção/adição de material no vetor de material
  nath = round(V(ite+1)*length(elem_list));
  nath = round(nath/4)*4; % Número par de elementos removidos/adicionados
  % Número de elementos adicionados
  nadd = nath-length(nonzeros(mat(ord(1:nath)) == 1));
  % Razão de adicão de materal (Addition Ratio)
  AR = nadd/nel:
  % Atualização do vetor de material
  [mat] = updatematBESO(AR,AR_max,ord,mat,Conectivity,elem_list,nath,ite,V);
  % Atualizando a matriz global
  [Kg] = KupdateBESO(Kg,Ke,id,Conectivity,nel,mat_old,mat);
  Compliance(ite) = sum(alpha_elem);
  % Cálculo do critério de convergencia da energia de deformaçao média
  if ite>=2*N
    errorS=abs(sum(Compliance(ite-N+1:ite))-sum(Compliance(ite-N*2+1:ite-N)))...
      /sum(Compliance(ite-N+1:ite));
  end
  % Show results by iteration / Mostrando as informação da iteração
  disp([' It.: ' sprintf('%4i',ite) ' Obj.: ' sprintf('%6.4e',sum(alphaS)) ...
    ' Vol.: ' sprintf('%6.3f',V(ite+1)) ' error.: ' sprintf('%3.2f %%',errorS*100)])
  if plot_ite==1; % Plot da estrutura em cada iteração
    z=reshape(mat,nelx,nely);
    colormap(gray); imagesc(-z'); axis equal; axis tight; axis off;pause(1e-6);
  end
end
toc
%% Pós-Processamento dos resultados
% Results post process
% Ploting optimized mesh
figure; malha(Nodes,Conectivity(elem_list(mat==1),:),'b',1,0,0)
axis off
%% Cálculo da tensão de von mises elementar
% Elementar von Mises Stress computation
fmac = figure; % colormap('copper')
for iel = elem_list(mat==1)
  nd = Conectivity(iel,2:end);
```

index = [nd(1)*2-1 nd(1)*2 nd(2)*2-1 nd(2)*2 nd(3)*2-1 nd(3)*2 nd(4)*2-1 nd(4)*2]; Pos_xy = Nodes([nd nd(1)],2:3); patch(Pos_xy(:,1),Pos_xy(:,2),repmat(alphaS(iel),5,1),'EdgeColor','none') end axis equal, axis off

% PLOTING COMPLIANCE AND VOLUME FRACTION figure('Name','Compliance and Volume fraction ','NumberTitle','off'); [AX,H1,H2]=plotyy(0:ite-1,Compliance,0:ite-1,V(1:end-1)*100); % [AX,H1,H2]=plotyy(1:ite,Compliance,1:ite,V(2:end)*100);

set(AX,{'ycolor'},{'k';'k'}); legend('Mean Compliance','Volume fraction')

set(get(AX(1),'Ylabel'),'String','Mean Compliance \alpha (Nm)'); set(get(AX(2),'Ylabel'),'String','Volume fraction (V/V_i)');

```
set(AX(1),'YLim',[0.95*min(Compliance) 1.25*max(Compliance)])
set(AX(1),'YTick',[linspace(0.95*min(Compliance),1.25*max(Compliance),11)])
set(AX(2),'YLim',[0 100]); set(AX(2),'YTick',[0:10:100]);
```

xlabel('Iteration'); title('\fontsize{13}Compliance and Volume fraction') set(H1,'LineStyle','-','Marker','x');set(H2,'LineStyle','-','Marker','o'); grid on

Rotina desenvolvida em Matlab® para a solução do problema da Estrutura em Forma de Mão-Francesa, utilizando o método dos elementos finitos e aplicando o algoritmo de otimização BESO (adaptação do código desenvolvido por Tainan Calixto e Renato Pavanello).

clear all; close all; clc;

% Vinicius Lourenço da Silva Daldon

% RA: 178275

- % EM 919 Trabalho de graduação II
- % Título: Comparação e implementação dos métodos de otimização topológica estrutural evolucionária BESO e baseada em programação matemática SIMP
- % O presente programa consiste em uma adaptação do código desenvolvido por

% Tainan Calixto e Renato Pavanello (DMC FEM - UNICAMP)

%% Dimensionando a malha:

Lx = 100e-3; %% Dimensão em x da malha [m] Ly = 100e-3; %% Dimensão em y da malha [m] Lz = 1e-3; %% Espessura [m]

Nx = 100; %% Número de elementos em x

Ny = 100; %% Número de elementos em y

%% Propriedades do material: E = 100e9; %% Módulo de elasticidade [Pa] v = 0.3; %% Coeficiente de Poisson %% Módulo da força aplicada: F = 100; %% Módulo da força aplicada [N] %% Parametros do programa original: %%%%%%% % Optimization parameter %%%%%%% Vfinal = 0.35; % Final Volume Fraction / Porcentagem de volume final ER = 0.010;% Evolutionary Remotion parameter / Porcentagem de remoção de material a cada iteração AR_max = 0.05; % Evolutionary Addition parameter/ Porcentagem de adição de material máximo (Addition ratio) % Filter radios / Raio mínimo de ponderação do filtro r_min = 6e-3; tau = 0.01/100; % Convergence parameter / Precisão do critério de convergência (Mean Compliance) N = 5; % Number of iteration for convergence / Numero de iterações para o cálculo do erro % plot index / Plot da estrutura em cada iteração [0-n,1-s] plot_ite=1; emat = [E v Lz]; nel = Nx^*Ny ; nnode = $(Nx+1)^*(Ny+1);$ elem_list = 1:nel; node_list = 1:nnode; ndof = 2;sdof = nnode*ndof; % Initial values variables / Inicializando as variáveis errorS = 0.9999; V = 1.0d0;mat = ones((Nx*Ny),1); alpha_elem = zeros((Nx*Ny),1); ite = 0; cond_volume = 0; %% Construção da malha: nos_linha = Nx + 1; %% Número de nós por linha

nos_coluna = Ny + 1;%% Número de nós por colunanos_X_Y = [nos_linha nos_coluna];%% Vetor de relação nó-linha e nó colunanos_total = nos_linha * nos_coluna;%% Número de nós na malhaa = Lx / Nx;%% Dimensão em x do elemento [m]b = Ly / Ny;%% Dimensão em y do elemento [m]

% Matriz com os coordenadas dos nós na malha, onde i = nó, j1 = coord_x_nó % e j2 = coord_y_nó:
coord_nos(nos_total,3) = zeros; %% Inicializando a matriz com zeros

```
delta_x = 0; %% Contador inicial da coordenada em x
delta_y = 0; %% Contador inicial da coordenada em y
```

```
for i = 1:(nos_linha * nos_coluna)
```

```
coord_nos(i,1) = i;
coord_nos(i,2) = delta_x;
coord_nos(i,3) = delta_y;
delta_x = delta_x + a;
if (mod(i,nos_linha)==0)
```

```
delta_x = 0;
delta_y = delta_y + b;
end
```

end

% Matriz com a relação entre elementos e nós que incidem neles, onde i = % elemento, j1 = nó_1, j2 = nó_2, j3 = nó_3 e j4 = nó_4:

nos_elem(Nx*Ny,5) = zeros; %% Inicializando a matriz com zeros

for $i = 1:(Nx^*Ny)$

nos_elem(i,1) = i; nos_elem(i,2) = i + fix((i-1)/Nx); %% Acréscimo na relação elemento-nós de acordo com a linha completa no loop nos_elem(i,3) = nos_elem(i,2) + 1; nos_elem(i,5) = nos_elem(i,2) + nos_linha; nos_elem(i,4) = nos_elem(i,2) + 1 + nos_linha;

end

% Matriz com a relação de nós com graus de liberdade, onde i = nó, j1 = gl_1 % e j2 = gl_2:

nos_gls(nos_linha*nos_coluna,2) = zeros; %% Inicializando a matriz com zeros

```
for i = 1:(nos_linha*nos_coluna)
```

```
nos_gls(i,1) = i * 2 - 1;
nos_gls(i,2) = i * 2;
```

end

%% Aplicação das condições de contorno na estrutura % Exemplo do programa (engastado na extremidade esquerda fixeddofs = [1 203 405 607 809 1011 1213 18989 19191 19393 19595 19797 19999 20201 2 204 406 608 810 1012 1214 18990 19192 19394 19596 19798 20000 20202];

alldofs = 1:2*nos_total; freedofs = setdiff(alldofs,fixeddofs);

%% Load aplications / Aplicação do carregamento na estrutura

$$\label{eq:generalized_force_node} \begin{split} &\mathsf{Fg} = \mathsf{sparse}(2^*\mathsf{nos_total},1);\\ &\mathsf{U} = \mathsf{zeros}(2^*\mathsf{nos_total},1);\\ &\mathsf{force_node} = 10201;\\ &\mathsf{Fg}(2^*\mathsf{force_node}) = \mathsf{-F}; \end{split}$$

%% Construção da matriz de rigidez do elemento: %%

% Construção das funções de forma

Pontos_Gauss = [0.577350269189626 0.577350269189626; 0.577350269189626 -0.577350269189626; -0.577350269189626; -0.577350269189626 -0.577350269189626]; %% Coordenadas utilizadas para a integração

for i = 1:4

r = Pontos_Gauss(i,1); s = Pontos_Gauss(i,2);

% Derivadas das funções de forma em: % I) relação a r: N1r = (1/4) * (s - 1); N2r = (1/4) * (1 - s); N3r = (1/4) * (1 + s); N4r = (1/4) * (1 + s) * (-1);

% l) relação a s: N1s = (1/4) * (r - 1);N2s = (1/4) * (1 + r) * (-1);N3s = (1/4) * (1 + r);N4s = (1/4) * (1 - r);

% Determinando os termos da matriz jacobiana:

J11 = coord_nos_elemento(1,1) * N1r + coord_nos_elemento(2,1) * N2r + coord_nos_elemento(3,1) * N3r + coord_nos_elemento(4,1) * N4r;

- J21 = coord_nos_elemento(1,1) * N1s + coord_nos_elemento(2,1) * N2s + coord_nos_elemento(3,1) * N3s + coord_nos_elemento(4,1) * N4s;
- J12 = coord_nos_elemento(1,2) * N1r + coord_nos_elemento(2,2) * N2r + coord_nos_elemento(3,2) * N3r + coord_nos_elemento(4,2) * N4r:
- J22 = coord_nos_elemento(1,2) * N1s + coord_nos_elemento(2,2) * N2s + coord_nos_elemento(3,2) * N3s + coord_nos_elemento(4,2) * N4s;

J = [J11 J12; J21 J22];

```
% Matriz auxiliar G:
```

G = (1 / det(J)) * [J22 0 - J12 0; 0 - J21 0 J11; -J21 J22 J11 - J12];

% Matriz auxiliar P:

```
\begin{split} \mathsf{P} &= [\mathsf{N1r} \ \mathsf{0} \ \mathsf{N2r} \ \mathsf{0} \ \mathsf{N3r} \ \mathsf{0} \ \mathsf{N4r} \ \mathsf{0}; \ \mathsf{0} \ \mathsf{N1r} \ \mathsf{0} \ \mathsf{N2r} \ \mathsf{0} \ \mathsf{N3r} \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0}; \ \mathsf{0} \ \mathsf{N1s} \ \mathsf{0} \ \mathsf{N2s} \ \mathsf{0} \ \mathsf{N3s} \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0}; \ \mathsf{0} \ \mathsf{N1s} \ \mathsf{0} \ \mathsf{N2s} \ \mathsf{0} \ \mathsf{N3s} \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0}; \ \mathsf{0} \ \mathsf{N1s} \ \mathsf{0} \ \mathsf{N2s} \ \mathsf{0} \ \mathsf{N3s} \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0}; \ \mathsf{0} \ \mathsf{N1s} \ \mathsf{0} \ \mathsf{N2s} \ \mathsf{0} \ \mathsf{N3s} \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0}; \ \mathsf{0} \ \mathsf{N1s} \ \mathsf{0} \ \mathsf{N2s} \ \mathsf{0} \ \mathsf{N3s} \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0}; \ \mathsf{0} \ \mathsf{N1s} \ \mathsf{0} \ \mathsf{N2s} \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0}; \ \mathsf{0} \ \mathsf{N1s} \ \mathsf{0} \ \mathsf{N2s} \ \mathsf{0} \ \mathsf{N3s} \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0}; \ \mathsf{0} \ \mathsf{N1s} \ \mathsf{0} \ \mathsf{N2s} \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0}; \ \mathsf{0} \ \mathsf{N1s} \ \mathsf{0} \ \mathsf{N2s} \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0}; \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0} \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0}; \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0}; \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0} \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0}; \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0} \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0} \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0} \ \mathsf{0} \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0} \ \mathsf{0} \ \mathsf{N4s} \ \mathsf{0} \
```

```
end
```

%% Montagem da matriz global:

```
ID = nos_gls';
loc = zeros (8,(Nx*Ny));
```

for i=1:(Nx*Ny)

```
loc (1,i) = nos_gls(nos_elem(i,2),1);
loc (2,i) = nos_gls(nos_elem(i,2),2);
loc (3,i) = nos_gls(nos_elem(i,3),1);
loc (4,i) = nos_gls(nos_elem(i,3),2);
loc (7,i) = nos_gls(nos_elem(i,5),1);
loc (8,i) = nos_gls(nos_elem(i,5),2);
loc (5,i) = nos_gls(nos_elem(i,4),1);
loc (6,i) = nos_gls(nos_elem(i,4),2);
```

```
end
```

```
I = reshape(repmat(loc,8,1),(Nx*Ny)*64,1); %% Matriz que repete os graus de liberdade associados aos elementos (8x)
Jota = kron(loc(:),ones(8,1)); %% Organização de I, deixando DOF's iguais em sequência
Kg = kron(mat,ones(64,1)).*repmat(Ke(:),(Nx*Ny),1); %% Inserindo as características de densidade e de K elementar
K = sparse(I,Jota,Kg); %% Montagem de K
%% Ajustes:
nnode = nos_total;
```

```
Conectivity = nos_tota,
nel = Nx*Ny;
Nodes = coord_nos;
Kg = K;
id = ID;
```

nelx = Nx;

```
nely = Ny;
```

%% Programa BESO PAVA

% assemblage of Filter matrix

% Montagem da Matriz de Filtro (alpha_elem = H*alpha_nodal)

[noconnect] = nodalconnect(nnode,Conectivity); % Matriz Conectividade Nodal

[center,elarea] = centroids(nel,Nodes(:,2:3),Conectivity);

[H] = filterH(Conectivity,elem_list,center,Nodes(:,2:3),r_min);

% assemblage of Filter matrix

% Montagem da Matriz de Filtro (alpha_elem = H*alpha_nodal)

[noconnect] = nodalconnect(nnode,Conectivity); % Matriz Conectividade Nodal

[center,elarea] = centroids(nel,Nodes(:,2:3),Conectivity);

[H] = filterH(Conectivity,elem_list,center,Nodes(:,2:3),r_min);

% Starting BESO iterations

% Inicio do processo de otimização - BESO

tic

```
while (cond_volume == 0) || (errorS >= tau)
```

ite = ite+1; mat_old = mat;

```
% solve Equation System / Solução da equação do sistema
U(freedofs) = Kg(freedofs,freedofs)\Fg(freedofs);
U(fixeddofs)= 0;
```

```
% Sensibility analysis / Cálculo da sensibilidade (Compliance)
```

```
for iel=elem_list
```

```
nd = Conectivity(iel,2:5);
```

```
index = [nd(1)*2-1 nd(1)*2 nd(2)*2-1 nd(2)*2 nd(3)*2-1 nd(3)*2 nd(4)*2-1 nd(4)*2];
```

```
if mat(iel)==1; alpha_elem(iel) = 0.5*U(index)'*Ke*U(index); % Elemento cheio
```

else alpha_elem(iel) = 0; % Elemento vazio

```
end
```

end

```
% Filter / Filtro Numérico
```

[alphaS] = filterBEFSOH(nel,nnode,noconnect,elem_list,node_list,1,alpha_elem,H);

```
% Estabilization of Evolutionary process / Estabilizando o Processo Evolucionário
if ite==1; alphaS_old = alphaS;
else
    alphaS = (1/2)*(alphaS_old + alphaS);
    alphaS_old = alphaS;
end
```

```
% Ordering mean compliance / Ordenando a Mean Compliance
[alphaS_ord,ord] = sort(alphaS,1,'descend');
```

```
% Neste volume calculation / Cálculo do volume desejado
if (V(ite) - Vfinal >= ER*V(ite)); V(ite+1)=V(ite)*(1-ER);
elseif (V(ite)-Vfinal <= -ER*V(ite)); V(ite+1)=V(ite)*(1+ER);
elseif (abs(V(ite)-Vfinal) < ER*V(ite));
```

```
V(ite+1)=Vfinal; cond_volume = 1;
```

end

```
% Choice of elements to be remouved
```

% Escolha dos elementos que serão retirados

% Linha limite de remoção/adição de material no vetor de material

```
nath = round(V(ite+1)*length(elem_list));
```

nath = round(nath/4)*4; % Número par de elementos removidos/adicionados

```
% Número de elementos adicionados
nadd = nath-length(nonzeros(mat(ord(1:nath)) == 1));
```

```
% Razão de adicão de materal (Addition Ratio)
AR = nadd/nel:
```

% Atualização do vetor de material [mat] = updatematBESO(AR,AR_max,ord,mat,Conectivity,elem_list,nath,ite,V);

```
% Atualizando a matriz global
[Kg] = KupdateBESO(Kg,Ke,id,Conectivity,nel,mat_old,mat);
```

Compliance(ite) = sum(alpha_elem);

% Cálculo do critério de convergencia da energia de deformaçao média

if ite>=2*N

errorS=abs(sum(Compliance(ite-N+1:ite))-sum(Compliance(ite-N*2+1:ite-N)))...

/sum(Compliance(ite-N+1:ite));

```
end
```

```
% Show results by iteration / Mostrando as informação da iteração
  disp([' It.: ' sprintf('%4i',ite) ' Obj.: ' sprintf('%6.4e',sum(alphaS)) ...
    ' Vol.: ' sprintf('%6.3f',V(ite+1)) ' error.: ' sprintf('%3.2f %%',errorS*100)])
  if plot_ite==1; % Plot da estrutura em cada iteração
    z=reshape(mat,nelx,nely);
    colormap(gray); imagesc(-z'); axis equal; axis tight; axis off;pause(1e-6);
  end
end
toc
%% Pós-Processamento dos resultados
% Results post process
% Ploting optimized mesh
figure; malha(Nodes,Conectivity(elem_list(mat==1),:),'b',1,0,0)
axis off
%% Cálculo da tensão de von mises elementar
% Elementar von Mises Stress computation
fmac = figure; % colormap('copper')
for iel = elem_list(mat==1)
  nd = Conectivity(iel,2:end);
  index = [nd(1)*2-1 nd(1)*2 nd(2)*2-1 nd(2)*2 nd(3)*2-1 nd(3)*2 nd(4)*2-1 nd(4)*2];
  Pos_xy = Nodes([nd nd(1)], 2:3);
  patch(Pos_xy(:,1),Pos_xy(:,2),repmat(alphaS(iel),5,1),'EdgeColor','none')
end
axis equal, axis off
% PLOTING COMPLIANCE AND VOLUME FRACTION
figure('Name','Compliance and Volume fraction ','NumberTitle','off');
[AX,H1,H2]=plotyy(0:ite-1,Compliance,0:ite-1,V(1:end-1)*100);
% [AX,H1,H2]=plotyy(1:ite,Compliance,1:ite,V(2:end)*100);
set(AX,{'ycolor'},{'k';'k'}); legend('Mean Compliance','Volume fraction')
set(get(AX(1),'Ylabel'),'String','Mean Compliance \alpha (Nm)');
set(get(AX(2),'Ylabel'),'String','Volume fraction (V/V_i)');
set(AX(1),'YLim',[0.95*min(Compliance) 1.25*max(Compliance)])
set(AX(1),'YTick',[linspace(0.95*min(Compliance),1.25*max(Compliance),11)])
set(AX(2),'YLim',[0 100]); set(AX(2),'YTick',[0:10:100]);
xlabel('Iteration'); title('\fontsize{13}Compliance and Volume fraction')
set(H1,'LineStyle','-','Marker','x');set(H2,'LineStyle','-','Marker','o');
grid on
```

Rotina desenvolvida em Matlab® para a solução do problema da Estrutura de Michell, utilizando o método dos elementos finitos e aplicando o algoritmo de otimização SIMP (adaptação do código desenvolvido por Sigmund, 2001). Utilizar o comando "clear all;close all;clc;top(50,25,0.5,3.0,1.5)" como input do Código.

%%%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLE SIGMUND, JANUARY 2000 %%% %%%% CODE MODIFIED FOR INCREASED SPEED, September 2002, BY OLE SIGMUND %%%

```
%%clear all;close all;clc;top(50,25,0.5,3.0,1.5)
%INPUT do código
function top(nelx,nely,volfrac,penal,rmin);
% INITIALIZE
x(1:nely,1:nelx) = volfrac;
loop = 0;
change = 1.;
% START ITERATION
while change > 0.01
 loop = loop + 1;
 xold = x;
% FE-ANALYSIS
 [U]=FE(nelx,nely,x,penal);
% OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
 [KE] = lk;
 c = 0.;
 for ely = 1:nely
  for elx = 1:nelx
   n1 = (nely+1)^*(elx-1)+ely;
   n2 = (nely+1)^* elx + ely;
   Ue = U([2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1;2*n2+2; 2*n1+1;2*n1+2], 1);
   c = c + x(ely,elx)^penal*Ue'*KE*Ue;
   dc(ely,elx) = -penal*x(ely,elx)^(penal-1)*Ue'*KE*Ue;
  end
 end
% FILTERING OF SENSITIVITIES
 [dc] = check(nelx,nely,rmin,x,dc);
% DESIGN UPDATE BY THE OPTIMALITY CRITERIA METHOD
 [x] = OC(nelx,nely,x,volfrac,dc);
% PRINT RESULTS
 change = max(max(abs(x-xold)));
 disp([' It.: ' sprintf('%4i',loop) ' Obj.: ' sprintf('%10.4f',c) ...
    ' Vol.: ' sprintf('%6.3f',sum(sum(x))/(nelx*nely)) ...
    ' ch.: ' sprintf('%6.3f',change )])
% Armazenando os resultados para a plotagem posterior:
 MCompl(loop) = c;
 Vfraction(loop) = sum(sum(x))/(nelx*nely);
 Change(loop) = change;
 Contloop(loop) = loop;
```

% PLOT DENSITIES

colormap(gray); imagesc(-x); axis equal; axis tight; axis off;pause(1e-6); end

figure('Name','Compliance and Volume fraction ','NumberTitle','off');

[AX,H1,H2] = plotyy(Contloop,MCompl,Contloop,Vfraction*100); set(AX,{'ycolor'},{'k';'k'}); legend('Mean Compliance','Volume fraction') set(get(AX(1),'Ylabel'),'String','Mean Compliance \alpha (Nm)'); set(get(AX(2),'Ylabel'),'String','Volume fraction (V/V_i)');

set(AX(1),'YLim',[0.95*min(MCompl) 1.25*max(MCompl)]) set(AX(1),'YTick',[linspace(0.95*min(MCompl),1.25*max(MCompl),11)]) set(AX(2),'YLim',[0 100]); set(AX(2),'YTick',[0:10:100]);

```
xlabel('Iteration'); title('\fontsize{13}Compliance and Volume fraction')
set(H1,'LineStyle','-','Marker','x');set(H2,'LineStyle','-','Marker','o');
grid on
```

V = U;

```
function [xnew]=OC(nelx,nely,x,volfrac,dc)
I1 = 0; I2 = 100000; move = 0.2;
while (12 - 11 > 1e - 4)
Imid = 0.5^{*}(I2+I1);
xnew = max(0.001,max(x-move,min(1.,min(x+move,x.*sqrt(-dc./Imid)))));
if sum(sum(xnew)) - volfrac*nelx*nely > 0;
 I1 = Imid;
else
 I2 = Imid;
end
end
function [dcn]=check(nelx,nely,rmin,x,dc)
dcn=zeros(nely,nelx);
for i = 1:nelx
for j = 1:nely
 sum=0.0;
 for k = max(i-floor(rmin),1):min(i+floor(rmin),nelx)
  for I = max(j-floor(rmin),1):min(j+floor(rmin),nely)
   fac = rmin-sqrt((i-k)^2+(j-l)^2);
   sum = sum+max(0,fac);
   dcn(j,i) = dcn(j,i) + max(0,fac)^*x(l,k)^*dc(l,k);
  end
 end
 dcn(j,i) = dcn(j,i)/(x(j,i)*sum);
end
```

end **FE-ANALYSIS** %%%%%%%%%%% function [U]=FE(nelx,nely,x,penal) [KE] = lk; K = sparse(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1)); $F = sparse(2^{(nely+1)^{(nelx+1),1)}; U = zeros(2^{(nely+1)^{(nelx+1),1)}; U = zeros(2^{(nely+1)^{(nelx+1),1}); U = zeros(2^{(nely+1)^{(nelx+1),1}}); U = zeros(2^{(nely+1),1}); U$ for elx = 1:nelx for ely = 1:nely $n1 = (nely+1)^*(elx-1)+ely;$ $n2 = (nely+1)^* elx + ely;$ edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1; 2*n2+2; 2*n1+1; 2*n1+2]; K(edof,edof) = K(edof,edof) + x(ely,elx)^penal*KE; end end % DEFINE LOADS AND SUPPORTS (HALF MBB-BEAM) F(1352,1) = -10000;fixeddofs = union([$(2^{(nely+1)-1}):1:2^{(nely+1)}$],[$(2^{(nelx+1)^{(nely+1)-1}}):1:2^{(nelx+1)^{(nely+1)}}$]; alldofs = $[1:2^{(nely+1)^{(nelx+1)}}];$ freedofs = setdiff(alldofs,fixeddofs); % SOLVING U(freedofs,:) = K(freedofs,freedofs) \ F(freedofs,:); U(fixeddofs,:) = 0;function [KE]=lk E = 100e9;nu = 0.3; k=(1e-3)*[1/2-nu/6 1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 ... -1/4+nu/12 -1/8-nu/8 nu/6 1/8-3*nu/8]; $KE = E/(1-nu^2)^{*}[k(1) k(2) k(3) k(4) k(5) k(6) k(7) k(8)$ k(2) k(1) k(8) k(7) k(6) k(5) k(4) k(3) k(3) k(8) k(1) k(6) k(7) k(4) k(5) k(2) k(4) k(7) k(6) k(1) k(8) k(3) k(2) k(5) k(5) k(6) k(7) k(8) k(1) k(2) k(3) k(4) k(6) k(5) k(4) k(3) k(2) k(1) k(8) k(7) k(7) k(4) k(5) k(2) k(3) k(8) k(1) k(6) k(8) k(3) k(2) k(5) k(4) k(7) k(6) k(1)]; % %%%%%%%% % This Matlab code was written by Ole Sigmund, Department of Solid % % Mechanics, Technical University of Denmark, DK-2800 Lyngby, Denmark. % % Please sent your comments to the author: sigmund@fam.dtu.dk % % % % The code is intended for educational purposes and theoretical details %

% "A 99 line topology optimization code written in Matlab" %

% by Ole Sigmund (2001), Structural and Multidisciplinary Optimization, %

%

% Vol 21, pp. 120--127. %

```
%
```

% are discussed in the paper

%%%%%%%%

%%clear all;close all;clc;top(100,100,0.35,3.0,1.5)

Rotina desenvolvida em Matlab® para a solução do problema da Estrutura em Forma de Mão-Francesa, utilizando o método dos elementos finitos e aplicando o algoritmo de otimização SIMP (adaptação do código desenvolvido por Sigmund, 2001). Utilizar o comando "clear all;close all;clc;top(100,100,0.35,3.0,1.5)" como input do Código.

%%%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLE SIGMUND, JANUARY 2000 %%% %%%% CODE MODIFIED FOR INCREASED SPEED, September 2002, BY OLE SIGMUND %%%

%INPUT do código function top(nelx,nely,volfrac,penal,rmin); % INITIALIZE x(1:nely,1:nelx) = volfrac; loop = 0;change = 1.; % START ITERATION while change > 0.01 loop = loop + 1;xold = x;% FE-ANALYSIS [U]=FE(nelx,nely,x,penal); % OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS [KE] = lk; c = 0.; for ely = 1:nely for elx = 1:nelx $n1 = (nely+1)^*(elx-1)+ely;$ n2 = (nely+1)* elx +ely; Ue = U([2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1;2*n2+2; 2*n1+1;2*n1+2],1); c = c + x(ely,elx)^penal*Ue'*KE*Ue; dc(ely,elx) = -penal*x(ely,elx)^(penal-1)*Ue'*KE*Ue; end end % FILTERING OF SENSITIVITIES [dc] = check(nelx,nely,rmin,x,dc); % DESIGN UPDATE BY THE OPTIMALITY CRITERIA METHOD

[x] = OC(nelx,nely,x,volfrac,dc); % PRINT RESULTS change = max(max(abs(x-xold))); disp([' lt.: ' sprintf('%4i',loop) ' Obj.: ' sprintf('%10.4f',c) ... ' Vol.: ' sprintf('%6.3f',sum(sum(x))/(nelx*nely)) ... ' ch.: ' sprintf('%6.3f',change)]) % Armazenando os resultados para a plotagem posterior: MCompl(loop) = c; Vfraction(loop) = sum(sum(x))/(nelx*nely); Change(loop) = change; Contloop(loop) = loop; % PLOT DENSITIES colormap(gray); imagesc(-x); axis equal; axis tight; axis off;pause(1e-6); end

figure('Name','Compliance and Volume fraction ','NumberTitle','off');

$$\begin{split} & [AX,H1,H2] = plotyy(Contloop,MCompl,Contloop,Vfraction*100); \\ & set(AX,{'ycolor'},{'k';'k'}); \ legend('Mean Compliance','Volume fraction') \\ & set(get(AX(1),'Ylabel'),'String','Mean Compliance \alpha (Nm)'); \\ & set(get(AX(2),'Ylabel'),'String','Volume fraction (V/V_i)'); \end{split}$$

set(AX(1),'YLim',[0.95*min(MCompl) 1.25*max(MCompl)]) set(AX(1),'YTick',[linspace(0.95*min(MCompl),1.25*max(MCompl),11)]) set(AX(2),'YLim',[0 100]); set(AX(2),'YTick',[0:10:100]);

xlabel('Iteration'); title('\fontsize{13}Compliance and Volume fraction') set(H1,'LineStyle','-','Marker','x');set(H2,'LineStyle','-','Marker','o'); grid on

V = U;

function [xnew]=OC(nelx,nely,x,volfrac,dc) I1 = 0; I2 = 100000; move = 0.2; while (12-11 > 1e-4) $lmid = 0.5^{*}(l2+l1);$ xnew = max(0.001,max(x-move,min(1.,min(x+move,x.*sqrt(-dc./Imid))))); if sum(sum(xnew)) - volfrac*nelx*nely > 0; I1 = Imid;else I2 = Imid;end end function [dcn]=check(nelx,nely,rmin,x,dc) dcn=zeros(nely,nelx); for i = 1:nelx

```
for j = 1:nely
  sum=0.0;
  for k = max(i-floor(rmin),1):min(i+floor(rmin),nelx)
   for l = max(j-floor(rmin),1):min(j+floor(rmin),nely)
   fac = rmin-sqrt((i-k)^2+(j-l)^2);
    sum = sum+max(0,fac);
    dcn(j,i) = dcn(j,i) + max(0,fac)^{*}x(l,k)^{*}dc(l,k);
   end
  end
  dcn(j,i) = dcn(j,i)/(x(j,i)*sum);
 end
end
%%%%%%%%%%%
                                                                                                             FE-ANALYSIS
function [U]=FE(nelx,nely,x,penal)
[KE] = Ik;
K = sparse(2^{(nelx+1)^{(nely+1)}}, 2^{(nelx+1)^{(nely+1)}});
F = sparse(2^{(nely+1)^{(nelx+1)},1); U = zeros(2^{(nely+1)^{(nelx+1)},1);
for elx = 1:nelx
for ely = 1:nely
  n1 = (nely+1)^*(elx-1)+ely;
  n2 = (nely+1)^* elx + ely;
  edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1; 2*n2+2; 2*n1+1; 2*n1+2];
  K(edof,edof) = K(edof,edof) + x(ely,elx)^penal*KE;
 end
end
% DEFINE LOADS AND SUPPORTS (HALF MBB-BEAM)
F(20202,1) = -100;
fixeddofs = [1 2 3 4 5 6 7 8 9 10 11 12 13 14 189 190 191 192 193 194 195 196 197 198 199 200 201 202];
alldofs = [1:2^{(nely+1)^{(nelx+1)}}];
freedofs = setdiff(alldofs,fixeddofs);
% SOLVING
U(freedofs,:) = K(freedofs,freedofs) \ F(freedofs,:);
U(fixeddofs,:) = 0;
function [KE]=lk
E = 1e9;
nu = 0.3:
k=(1e-3)*[ 1/2-nu/6 1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 ...
 -1/4+nu/12 -1/8-nu/8 nu/6
                            1/8-3*nu/8];
KE = E/(1-nu^2)^*[k(1) k(2) k(3) k(4) k(5) k(6) k(7) k(8)]
         k(2) k(1) k(8) k(7) k(6) k(5) k(4) k(3)
         k(3) k(8) k(1) k(6) k(7) k(4) k(5) k(2)
         k(4) k(7) k(6) k(1) k(8) k(3) k(2) k(5)
         k(5) k(6) k(7) k(8) k(1) k(2) k(3) k(4)
         k(6) k(5) k(4) k(3) k(2) k(1) k(8) k(7)
         k(7) k(4) k(5) k(2) k(3) k(8) k(1) k(6)
         k(8) k(3) k(2) k(5) k(4) k(7) k(6) k(1)];
```

%

%

% This Matlab code was written by Ole Sigmund, Department of Solid % % Mechanics, Technical University of Denmark, DK-2800 Lyngby, Denmark. % Please sent your comments to the author: sigmund@fam.dtu.dk % % % % The code is intended for educational purposes and theoretical details % % are discussed in the paper % % "A 99 line topology optimization code written in Matlab" % % by Ole Sigmund (2001), Structural and Multidisciplinary Optimization, % % Vol 21, pp. 120--127. % % % % The code as well as a postscript version of the paper can be % % downloaded from the web-site: http://www.topopt.dtu.dk % % % % Disclaimer: % % The author reserves all rights but does not guaranty that the code is %

% free from errors. Furthermore, he shall not be liable in any event %

% caused by the use of the program.

%