



UNIVERSIDADE FEDERAL DE GOIÁS – UFG – REGIONAL CATALÃO
UNIDADE ACADÊMICA ESPECIAL EM MATEMÁTICA E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM E OTIMIZAÇÃO



GUSTAVO HENRIQUE JESUS MESQUITA

OTIMIZAÇÃO TOPOLÓGICA DE ABSORVEDORES DINÂMICOS DE
VIBRAÇÕES SUJEITO A VIBRAÇÕES LIVRE E FORÇADA

DISSERTAÇÃO DE MESTRADO

CATALÃO-GO, 2016

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR AS TESES E DISSERTAÇÕES ELETRÔNICAS NA BIBLIOTECA DIGITAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio da Biblioteca Digital de Teses e Dissertações (BDTD/UFG), regulamentada pela Resolução CEPEC nº 832/2007, sem ressarcimento dos direitos autorais, de acordo com a Lei nº 9610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou *download*, a título de divulgação da produção científica brasileira, a partir desta data.

1. Identificação do material bibliográfico: Dissertação Tese

2. Identificação da Tese ou Dissertação

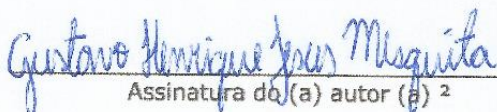
Nome completo do autor: Gustavo Henrique Jesus Mesquita

Título do trabalho: Otimização Topológica de Absorvedores Dinâmicos de Vibrações sujeito a vibração livre e forçada.

3. Informações de acesso ao documento:

Concorda com a liberação total do documento SIM NÃO¹

Havendo concordância com a disponibilização eletrônica, torna-se imprescindível o envio do(s) arquivo(s) em formato digital PDF da tese ou dissertação.


Assinatura do(a) autor (ã) ²

Data: 23 / 09 / 2016

¹ Neste caso o documento será embargado por até um ano a partir da data de defesa. A extensão deste prazo suscita justificativa junto à coordenação do curso. Os dados do documento não serão disponibilizados durante o período de embargo.

² A assinatura deve ser escaneada.

GUSTAVO HENRIQUE JESUS MESQUITA

**OTIMIZAÇÃO TOPOLÓGICA DE ABSORVEDORES DINÂMICOS DE
VIBRAÇÕES SUJEITO A VIBRAÇÕES LIVRE E FORÇADA**

Dissertação apresentada como requisito parcial para obtenção do título de Mestre em Modelagem e Otimização pela Universidade Federal de Goiás – Regional Catalão.

Orientador: Donald Mark Santee

CATALÃO-GO

2016

Ficha de identificação da obra elaborada pelo autor, através do Programa de Geração Automática do Sistema de Bibliotecas da UFG.

Mesquita, Gustavo Henrique Jesus

Otimização Topológica de Absorvedores Dinâmicos de Vibrações sujeito a vibrações livre e forçada. [manuscrito] / Gustavo Henrique Jesus Mesquita. - 2016.

73 f.: il.

Orientador: Prof. Dr. Donald Mark Santee.

Dissertação (Mestrado) - Universidade Federal de Goiás, Unidade Acadêmica Especial de Matemática e Tecnologia, Catalão, Programa de Pós-Graduação em Modelagem e Otimização, Catalão, 2016.

Bibliografia. Apêndice.

Inclui siglas, abreviaturas, símbolos, gráfico, tabelas, algoritmos, lista de figuras.

1. método dos elementos finitos. 2. otimização topológica. 3. atenuação de vibrações. I. Santee, Donald Mark, orient. II. Título.

CDU 621

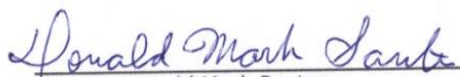


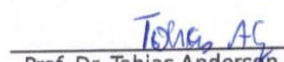
Defesa Nº _____

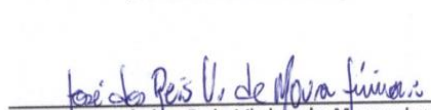
Ata de Defesa Pública - Dissertação de Mestrado

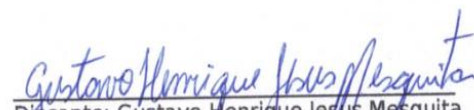
Aos 29 dias do mês de agosto do ano de 2016, às 14h:00min, reuniram-se os componentes da banca examinadora, professores Dr. Donald Mark Santee (presidente e orientador), Dr. Tobias Anderson Guimarães e Dr. José dos Reis Vieira de Moura Junior para, em sessão pública realizada no auditório Congadas, da Regional Catalão (RC), da Universidade Federal de Goiás (UFG), procederem com a avaliação do trabalho intitulado: "Otimização Topológica de Absorvedores Dinâmicos de Vibrações Sujeito a Vibração Livre e Forçada", em nível de Mestrado, área de concentração *Modelagem e Otimização*, de autoria de Gustavo Henrique Jesus Mesquita, discente do Programa de Pós-Graduação em Modelagem e Otimização (PPGMO) da UFG/RC. A sessão foi aberta pelo presidente da banca, que fez a apresentação formal dos membros da banca. A seguir, a palavra foi concedida ao discente que, dentro do tempo regulamentar, procedeu a apresentação de seu trabalho. Terminada a apresentação, cada membro da banca arguiu o candidato, tendo-se adotado o sistema de diálogo sequencial. Terminada a fase de arguição, procedeu-se a avaliação do trabalho. Os membros da banca consideraram o trabalho final: **Aprovado** (unanimidade). Cumpridas as formalidades de pauta, às 15h:30min a presidência da mesa encerrou a sessão e para constar, eu Donald Mark Santee, lavrei a presente Ata que, depois de lida e aprovada, segue assinada pelos membros da banca examinadora e pelo discente e, posteriormente, será homologada pelo Colegiado do PPGMO.

Catalão-GO, 29 de agosto de 2016.


Prof. Dr. Donald Mark Santee
Programa de Pós-Graduação em Modelagem
e Otimização, UFG/RC.
(Presidente da Banca)


Prof. Dr. Tobias Anderson Guimarães
Universidade Federal do Triângulo Mineiro


Prof. Dr. José dos Reis Vieira de Moura Junior
Programa de Pós-Graduação em Modelagem
e Otimização, UFG/RC.


Discente: Gustavo Henrique Jesus Mesquita
Programa de Pós-Graduação em Modelagem
e Otimização, UFG/RC.

GUSTAVO HENRIQUE JESUS MESQUITA

**OTIMIZAÇÃO TOPOLÓGICA DE ABSORVEDORES DINÂMICOS DE
VIBRAÇÕES SUJEITOS A VIBRAÇÕES LIVRE E FORÇADA**

Dissertação apresentada como requisito parcial para obtenção do título de Mestre em Modelagem e Otimização pela Universidade Federal de Goiás – Regional Catalão.

Trabalho aprovado em _____ de _____ de 20____.

Donald Mark Santee

Orientador

Tobias Anderson Guimarães

UFTM / Uberaba

José dos Reis Vieira de Moura Junior

PPGMO / RC

Catalão – GO

2016

Dedico este trabalho primeiramente a Deus, por ser essencial em minha vida, meu guia, socorro presente na hora de angústia, ao meu pai Gervásio, minha mãe Lucimar, minha esposa Jussara e aos meus irmãos.

AGRADECIMENTOS

A Deus por ter me dado saúde e inteligência para superar todas as dificuldades e conseguir chegar onde hoje estou.

A minha esposa Jussara, pelo apoio incondicional, por sua presença nos momentos difíceis e principalmente por seu amor que foi o combustível nesta jornada.

Aos meus pais Gervásio e Lucimar, meus irmãos Heitor e Beatriz, e a toda minha família que, com muito carinho e apoio, não mediram esforços para que eu chegasse até esta etapa de minha vida.

Ao meu orientador Prof. Dr. Donald Mark Santee pela paciência e incentivo que tornaram possível a conclusão desta dissertação.

Ao Prof. Dr. Tobias Anderson Guimarães, pelo convívio na graduação, apoio, compreensão e por sua amizade.

A Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (Capes) pelo apoio financeiro concedido.

E a todos que contribuíram de alguma forma para minha formação como pessoa até agora.

“A menos que modifiquemos à nossa maneira de pensar, não seremos capazes de resolver os problemas causados pela forma como nos acostumamos a ver o mundo”.

(Albert Einstein)

RESUMO

MESQUITA, G.H.J. *Otimização Topológica de Absorvedores Dinâmicos de Vibrações sujeitos a vibrações livre e forçada*. 2016. 70 f. Dissertação (Mestrado em Modelagem e Otimização) – Unidade Acadêmica Especial de Matemática e Tecnologia, Universidade Federal de Goiás – Regional Catalão, Catalão – GO

Propõe-se uma metodologia de aplicação do MEF (Método dos Elementos Finitos), juntamente com uma técnica de otimização topológica como ferramenta de análise e projeto ótimo de sistemas mecânicos sujeitos a vibração forçada. Em particular, chega-se à geometria ótima de um absorvedor dinâmico de vibrações contínuo. Apresenta-se a modelagem matemática, além de se desenvolver um código em Matlab® que implementa a técnica de otimização topológica por distribuição de material aplicado a sistemas mecânicos contínuos com excitação externa e harmônica de tal forma que a menor frequência natural seja predefinida.

Palavras-chaves: método dos elementos finitos, otimização topológica, atenuação de vibrações.

ABSTRACT

MESQUITA, G.H.J. *Otimização Topológica de Absorvedores Dinâmicos de Vibrações sujeito a vibração livre e forçada*. 2016. 70 f. Dissertação (Mestrado em Modelagem e Otimização) – Unidade Acadêmica Especial de Matemática e Tecnologia, Universidade Federal de Goiás – Regional Catalão, Catalão – GO

The aim of this work is to propose a methodology of application of the FEM and topology optimization technique as tools of analysis and optimal design of mechanical systems subject to natural and forced vibrations in order to reach the optimal geometry of a dynamic absorbing of vibrations. It presents the mathematical reasoning and develop a program in Matlab® that implements the topology optimization technique will be employed to generate the optimal material distribution (layout) continuous mechanical systems without external excitation and subject to harmonic forces with preset frequency.

Keywords : finite element method, topological optimization, vibration attenuation.

LISTA DE FIGURAS

Figura 1: ADV contínuo (viga bi apoiada) acoplada a estrutura primária (Adaptado de Cunha Jr., 1999).....	27
Figura 2: Topologia ótima de uma viga engastada sujeita a uma carga estática concentrada na extremidade livre.	28
Figura 3: Três categorias de otimização estrutural. a) otimização de dimensionamento de uma estrutura de treliça, b) forma otimização e c) otimização topológica. Os problemas iniciais são mostrados no lado esquerdo e as soluções ideais são mostrados à direita	31
Figura 4: Diagrama de fluxo do cálculo para otimização topológica usando o método de distribuição de material.....	41
Figura 5: A influência da fração de volume. Uma viga engastada discretizada por 6400 elementos quadrados e otimizada para frações de volume de b) 80% c) 60% d) 40% e e) de 20%. Para volumes pequenos a estrutura ótima resultante parece uma treliça... 41	
Figura 6: Otimização Topológica da viga. Topo: domínio de projeto completo, Meio: domínio da metade do design com condições de contorno de simetria e Inferior: resultado da Otimização Topológica da viga (duas metades).	45
Figura 7: A ordem do índice dos elementos finitos.	47
Figura 8: Relação entre as coordenadas do sistema local e global.	47
Figura 9: Metade da viga bi apoiada com carregamento central. Foi utilizado o código topsemfiltro(60,20,0.5)	50
Figura 10: Metade da viga bi apoiada com carregamento central. Foi utilizado o código topsemfiltro(60,20,0.5,3)	52
Figura 11: Resultado da Otimização Topológica da viga bi apoiada utilizando uma malha de 200x40 elementos, e um filtro de 1.5.....	54
Figura 12: Resultado da Otimização Topológica da viga bi apoiada utilizando uma malha de 100x20 elementos, e um filtro de 1.5.....	54
Figura 13: Sistema Massa-ADV dois graus de liberdade	55
Figura 14: Elemento quadrilátero de 4 nós.....	59
Figura 15: Exemplo 1, tempo de processamento: aprox. 2 minutos.....	64
Figura 16: Exemplo 2: 100 iterações – tempo de processamento: aprox. 4 minutos	64

Figura 17: Exemplo 3: Malha: 20x10, fração de volume: 0,6 - 50 iterações - tempo de processamento: aprox.. 2 horas e 30 minutos	65
Figura 18: Exemplo 4: Malha: 40x20, Fração de volume: 0.8 - tempo de processamento aproximadamente 70 horas	65

LISTA DE SÍMBOLOS

- ADV – Absorvedor Dinâmico de Vibrações
- MEF – Método dos Elementos Finitos
- Matlab® – Matrix Laboratory
- SIMP - *Solid Isotropic Material with Penalization*,
- \mathbb{R}^2 – Conjunto dos pares de números reais
- \mathbb{R}^3 – Conjunto dos ternos dos números reais
- $E_{ij}(x)$ – Matriz de elasticidade
- Ω - Domínio de referência do material
- E_{ad} - Espaço das soluções admissíveis da variável de projeto
- E_{ij}^0 - Matriz de elasticidade de um material isotrópico
- ρ - Densidade
- \bar{u} - Multiplicador de Lagrange para a restrição de equilíbrio
- U – Campo dos deslocamentos cineticamente admissíveis
- $\Lambda, \lambda^-(x), \lambda^+(x)$ - Multiplicadores de Lagrange das restrições
- 2D – Duas dimensões
- r_{min} – Raio mínimo
- E – Módulo de Young
- m_a - Massa do sistema principal
- m_a – Massa do sistema do absorvedor
- k_p - Rigidez do sistema principal
- k_a - Rigidez do sistema do absorvedor
- x_p - Deslocamentos do sistema principal
- x_a – Deslocamento do absorvedor
- λ - Menor autovalor da matriz global
- $[M]$ – Matriz de massa
- $[K]$ – Matriz de rigidez

SUMÁRIO

1. INTRODUÇÃO	26
1.1 Apresentação	26
1.2 Organização do trabalho	29
2. OTIMIZAÇÃO TOPOLÓGICA POR DISTRIBUIÇÃO DE MATERIAL ISOTRÓPICO.....	30
2.1 Formulação do problema e parametrização de projeto	30
2.2 A formulação baseada na flexibilidade mínima	31
A parametrização	33
2.3 O Método de Solução	36
2.3.1 A Função de Atualização das Densidades.....	36
2.3.2 A implementação do método de otimização para o MEF.	39
2.3.3 Sobre o cálculo do gradiente e métodos de programação matemática	41
3. O PROGRAMA	44
3.1 A implementação do programa em Matlab®	44
3.2 Programa Principal: linhas 1 a 36	45
3.3 Atualização das densidades: linhas 37 a 48	48
3.4 O Código do Método dos Elementos Finitos	49
3.5 Um exemplo	50
3.6 Incluindo as penalidades	51
3.7 Tornando a otimização independente da malha	52
4. OTIMIZAÇÃO TOPOLÓGICA PARA O CASO DINÂMICO.....	55
4.1 Modelagem para o caso Dinâmico	55
4.2 Montagem da matriz de massa elementar	57
4.3 A Otimização Topológica para o caso Dinâmico e alterações no código Matlab®	60
4.4 Exemplo de caso: Sistema Dinâmico	63
5. CONCLUSÕES	67
REFERÊNCIAS BIBLIOGRÁFICAS	69
APÊNDICE A.....	71

1. INTRODUÇÃO

1.1 Apresentação

A presença de vibrações mecânicas em estruturas provoca uma série de inconvenientes que podem comprometer a sua integridade física (Inman, 2005). Além de comprometer a integridade de uma estrutura, devido à ocorrência de fratura por fadiga, a vibração mecânica normalmente é a principal fonte de ruído e desconforto acústico.

Nestes casos, a primeira etapa a ser adotada para a resolução do problema é a construção de modelos convenientes correlacionando o nível de vibração (resposta) com a força de excitação (entrada) atuante na estrutura. Após a modelagem, o último passo consiste ou em modificar as características de projeto da estrutura a fim de reduzir a vibração a níveis aceitáveis, ou em projetar dispositivos adequados de controle das vibrações.

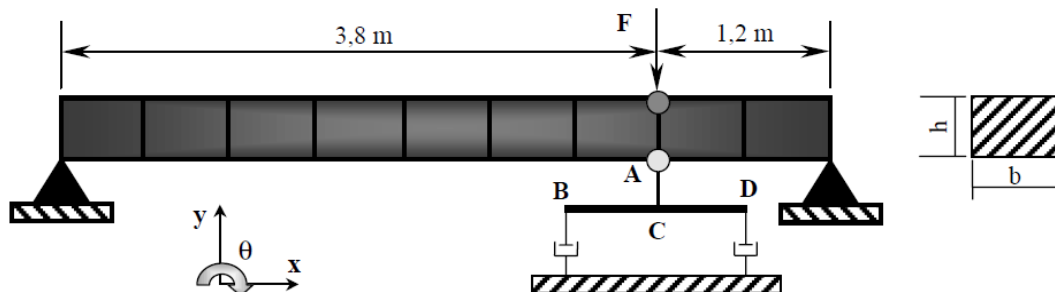
Uma das formas de se atenuar a vibração mecânica consiste na introdução de um dispositivo, conhecido com Absorvedor Dinâmico de Vibrações (ADV), em pontos estratégicos do sistema. Segundo Cunha Jr (1999), um ADV pode ser definido como um sistema vibratório que quando acoplado a estrutura principal, também dita estrutura primária, é capaz de absorver a energia vibratória do caminho entre a saída (resposta) e a entrada aplicada ao sistema (força de excitação).

Neste contexto, existe uma série de classificações e diferentes tipos de ADV's que podem ser projetados e aplicados a cada tipo de situação, como por exemplo, os ADV's de parâmetros concentrados, os ADV's contínuos, os ADV's adaptativos, os ADV's não lineares, dentre outros (Meirovitch, 1986; Cunha Jr., 1999, Inman, 2005; Vigié e Kerchen, 2009).

Nesta dissertação serão estudados apenas os ADV's contínuos, ou seja, os dispositivos de atenuação de vibrações com parâmetros distribuídos de massa e rigidez em forma de viga bi apoiada.

Um exemplo bastante comum de ADV contínuo é o da viga acoplada a estrutura primária que se deseja atenuar as vibrações. A título de ilustração, a figura 1 mostra um esboço de uma viga bi apoiada acoplada a um sistema vibratório amortecido de 1 grau de liberdade (Cunha Jr., 1999).

Figura 1: ADV contínuo (viga bi apoiada) acoplada a estrutura primária (Adaptado de Cunha Jr., 1999)



Fonte: autor

Normalmente os ADV's contínuos são projetados para atenuar as vibrações da estrutura primária sujeita a uma excitação harmônica em uma determinada banda de frequências. O ajuste dos parâmetros de projeto do ADV (massa, rigidez e amortecimento) é feito com base em tentativa e erro ou usando técnicas de otimização. Podem ser encontrados na literatura, diferentes trabalhos que abordam o ajuste dos parâmetros de um ADV contínuo usando algoritmos de otimização clássicos (métodos quasi-Newton, gradiente conjugado, etc.) ou inteligentes (Algoritmo genético, busca tabu, colônia de formigas, recozimento simulado, etc.) (Cunha Jr., 1999; Rade e Steffen Jr., 1999).

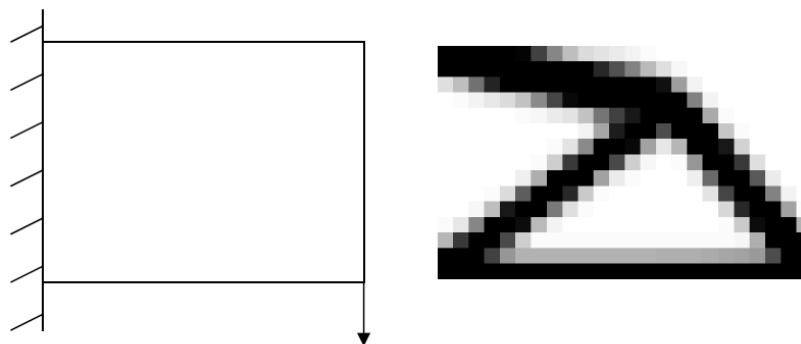
Nestes casos, os parâmetros físicos de massa, rigidez e amortecimento do ADV são otimizados de forma a minimizar o ganho da função de transferência do sistema na banda de frequência da força de excitação que se deseja reduzir a vibração.

Além da otimização paramétrica, podem ser aplicados também outras metodologias de otimização já existentes na literatura no projeto de um ADV contínuo como otimização de topologia, forma ou topográfica. Na otimização de forma, somente o contorno do espaço de projeto da estrutura é modificada para se atingir o ótimo da função objetivo e restrições.

Com a otimização de topologia, tanto o contorno como os furos existentes no espaço de projeto, ou seja, a distribuição de massa poderá ser modificada de acordo com a solução do problema de otimização. Para demonstrar o objetivo desta técnica de projeto, a figura 2 ilustra a topologia ótima de uma viga engastada sujeita a uma carga concentrada

na extremidade livre. Já na otimização paramétrica, somente uma dimensão (espessura, altura ou área) é otimizada satisfazendo o critério adotado para a função objetivo e restrições (Bendsoe e Sigmund, 2007; Christensen e Klarben, 2009).

Figura 2: Topologia ótima de uma viga engastada sujeita a uma carga estática concentrada na extremidade livre.



Fonte: autor

Neste trabalho, aplicou-se a técnica de otimização de topologia na geração do melhor layout, ou distribuição de massa de um ADV contínuo a ser usado na atenuação de vibrações de sistemas vibratórios discretos de 1 grau de liberdade. Ao se definir a banda de frequências de operação da estrutura primária (sistema vibratório discreto) e o ponto de conexão desta com o ADV contínuo, sua distribuição de massa será otimizada usando as técnicas de otimização de topologia.

Podem ser encontradas na literatura alguns trabalhos que utilizam otimização de topologia no projeto de sistemas vibratórios contínuos sujeitos a vibrações naturais e forçadas. Por exemplo, Pedersen (2000) utilizou esta técnica em conjunto com o Método da Homogenização como modelo de material para maximizar a diferença entre os autovalores (frequências naturais) de sistema contínuos sujeitos a vibrações naturais. Por outro lado, Jog (2002) utilizou otimização de topologia para minimizar a energia de deformação de uma estrutura sujeita a excitação harmônica com uma frequência pré-determinada.

O principal desafio deste trabalho foi a elaboração de um código eficiente que conseguisse gerar a topologia ótima de sistemas vibratórios contínuos. Alguns dos problemas que foram encontrados na implementação computacional do problema de otimização de topologia foram: não convexidade da função objetivo e restrições, o que obrigou a se limitar o número de iterações no algoritmo de tal forma que o mesmo pode

encerrar a otimização prematuramente e escolha adequada dos pesos na função objetivo necessários para a convergência do método.

No código para a geração de topologias ótimas considerou-se que estes sistemas devem funcionar como absorvedores de vibração de estruturas primárias que entram na modelagem como uma força harmônica aplicada ao absorvedor.

O código de otimização de topologia foi implementado em linguagem Matlab®. Todas as funções necessárias para a elaboração do código, como por exemplo, extração de autovalores e auto vetores, cálculo da energia de deformação, etc. foram utilizadas do próprio software Matlab®.

A respeito das contribuições deste trabalho, a principal refere-se ao delineamento dos benefícios obtidos com um sistema mecânico otimizado utilizando para isso formas otimizadas capaz de identificar possíveis ameaças a esse sistema contribuindo assim para uma maior qualidade de projeto.

Além disso, a disponibilidade de informações para a comunidade científica em específico, a engenharia mecânica, também foi uma contribuição objetivada pelo trabalho, de forma que este estudo venha a ser o início de futuros desenvolvimentos de sistemas mecânicos que recorreram à otimização topológica.

1.2 Organização do trabalho

Em relação à organização do trabalho, o mesmo está dividido em 4 capítulos, sendo estes ordenados da seguinte maneira:

Capítulo 1 – consiste na introdução do trabalho, onde os objetivos, justificativas e metodologia que permeiam o desenvolvimento do mesmo são abordados;

Capítulo 2 – discorre a Otimização Topológica, seus fundamentos, definições e as mais variadas formas de obtenção de resultados

Capítulo 3 – no decorrer deste é descrito o programa em Matlab® para Otimização Topológica, e traz exemplos de casos que já foram feitos, apresentando minuciosamente sua sistemática de execução para o caso estático;

Capítulo 4 – apresenta a metodologia adotada para se resolver o problema da vibração forçada, e os seus resultados.

2. OTIMIZAÇÃO TOPOLÓGICA POR DISTRIBUIÇÃO DE MATERIAL ISOTRÓPICO

Este capítulo apresenta uma visão geral dos elementos básicos do *método de distribuição de material* para encontrar o melhor formato de uma estrutura linear elástica. O formato de uma estrutura inclui informações sobre a topologia, forma e dimensão da mesma. O *método de distribuição de material* trata todas essas três questões simultaneamente.

Problemas de dimensionamento, otimização de forma, e de topologia abordam diferentes aspectos do problema de projeto de estruturas. Em um problema típico de dimensionamento o objetivo pode ser o de encontrar a distribuição da espessura ótima de uma placa linear elástica ou as áreas ótimas das barras de uma estrutura treliçada. A distribuição ótima de espessura minimiza (ou maximiza) uma quantidade física, tal como a flexibilidade (trabalho das forças externas), pico de tensão, deformação, etc., enquanto o equilíbrio e outras restrições sobre as variáveis de estado e de projeto são satisfeitas. A variável de projeto, é a espessura da placa e a variável de estado pode ser a sua deflexão.

A figura 3 ilustra cada um dos tipos de otimização estrutural. A principal característica do problema de **dimensionamento** (Figura 3a) é que o domínio do modelo e das variáveis de estado são conhecidos *a priori* e são fixos ao longo do processo de otimização. Por outro lado, em um problema de **otimização de forma** (Figura 3b) o objetivo é encontrar a forma ótima deste domínio, isto é, o problema de forma é definido em um domínio que é agora a variável de projeto. A **otimização topológica** (Figura 3c) de estruturas sólidas envolve a determinação de características, tais como o número, localização e a forma dos orifícios, e a conectividade do domínio.

2.1 Formulação do problema e parametrização de projeto

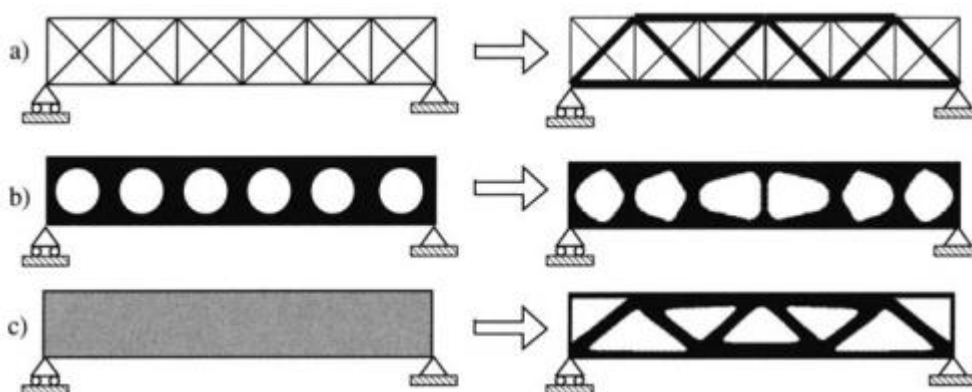
A formulação que será descrita a seguir combina várias características dos problemas tradicionais de otimização estrutural. O objetivo da otimização topológica é encontrar a distribuição ótima de material de uma estrutura dentro de uma região determinada, onde as únicas quantidades conhecidas no problema são:

- As cargas aplicadas;
- Os apoios;

- O volume da estrutura a ser construída; e
- Algumas restrições adicionais tais como a localização e o tamanho dos vazios prescritos ou áreas sólidas.

Nesta formulação o tamanho físico, a forma e a conectividade da estrutura são inicialmente desconhecidos.

Figura 3: Três categorias de otimização estrutural. a) otimização de dimensionamento de uma estrutura de treliça, b) forma otimização e c) otimização topológica. Os problemas iniciais são mostrados no lado esquerdo e as soluções ideais são mostrados à direita



Fonte: Bendsoe e Sigmund, 2007

A topologia, forma e tamanho da estrutura não são representados por funções paramétricas padrão, mas por um conjunto de funções definidas, distribuídas em um domínio de projeto fixo. Estas funções, por sua vez, representam uma parametrização do tensor de rigidez do contínuo e é a escolha apropriada dessa parametrização que conduz à formulação da otimização topológica.

2.2 A formulação baseada na flexibilidade mínima

Na formulação para a otimização de forma, escrito como um problema de distribuição de material, os conceitos são análogos às formulações para o dimensionamento de estruturas discretas e contínuas. O tipo de problema considerado leva, do ponto de vista computacional, a um problema de grande escala quanto ao número de variáveis de projeto.

Segundo Bendsoe e Sigmund (2007), devido a essa grande quantidade de variáveis, desde os primeiros problemas tratados nesta área empregou-se o tipo mais

simples de formulação tanto em termos da função objetivo quanto das restrições, ou seja, buscou-se minimizar flexibilidade (maximizar a rigidez global) usando as restrições mais simples possíveis.

Na construção da formulação considera-se um corpo ocupando um domínio Ω^{material} que é parte de um domínio de referência maior Ω , subconjunto do \mathbb{R}^2 ou \mathbb{R}^3 . Esse domínio de referência Ω é escolhido de modo a permitir uma definição das cargas aplicadas e condições de contorno.

Partindo-se do domínio de referência Ω define-se o problema de otimização topológica como determinar a matriz de densidade relativa contendo zero (ausência de material) e um (presença de material) dentro do domínio de projeto.

Definindo-se a forma quadrática de energia, que é o trabalho virtual das forças internas de um corpo elástico no ponto de equilíbrio u , devido a um deslocamento virtual arbitrário v , tem-se:

$$a(u, v) = \int_{\Omega} E_{ij}(x) \epsilon_i(u) \epsilon_j(v) d\Omega \quad (2.1)$$

onde $\epsilon_i(u)$ é o vetor das deformações no equilíbrio.

O trabalho das forças externas é:

$$l(u) = \int_{\Omega} f u \, d\Omega + \int_{T_T} t u \, ds \quad (2.2)$$

Assim o problema de minimizar a flexibilidade (ou maximizar a rigidez) pode ser escrito como:

$$\begin{aligned} \min \quad & l(u) \\ & u \in U, E \\ a_E(u, v) = l(v) \quad & \text{para qualquer } v \in U, E \in E_{ad} \end{aligned} \quad (2.3)$$

A equação de equilíbrio está escrita em sua forma variacional fraca, com U denotando o espaço do campo de deslocamentos cinematicamente admissíveis, f são as forças do corpo e t as trações de fronteira por parte de tração $T_T \subset T \equiv \partial\Omega$ da fronteira. O índice E indica que a forma quadrática e a_E depende das variáveis de projeto.

Na equação (2.3) E_{ad} denota o espaço das soluções admissíveis da variável de

projeto, E_{ad} . Existe uma restrição para E_{ad} que representa a limitação da quantidade de material que pode ser usada. Essa restrição pode ser expressa como:

$$\int_{\Omega^{mat}} 1 \, d\Omega \leq V \quad (2.4)$$

Uma implementação dessa restrição sobre E_{ad} será apresentada mais adiante.

Nessa dissertação a formulação (2.3) será discretizada para resolução pelo Método dos Elementos Finitos. Observa-se, também, que há dois campos vetoriais de interesse em (2.3), que são: o campo de deslocamentos \mathbf{u} e a matriz de elasticidade \mathbf{E} . Ao usar a mesma malha de elementos finitos para ambos os campos, e considerar \mathbf{E} como constante dentro de cada elemento, pode se escrever a forma discretizada de (2.3) como:

$$\begin{aligned} & \min \mathbf{f}^T \mathbf{u} \\ \text{s. a. : } & \mathbf{K}(E_e) \mathbf{u} = \mathbf{f}, \\ & E_e \in E_{ad}. \end{aligned} \quad (2.5)$$

onde \mathbf{u} e \mathbf{f} são os vetores de deslocamento e de carregamento, respectivamente. A matriz de rigidez \mathbf{K} depende do módulo de elasticidade, E_e , do elemento e , numerada como $e = 1, \dots, N$, onde \mathbf{K} pode ser escrita na forma:

$$\mathbf{K} = \sum_{e=1}^N \mathbf{K}_e(E_e) \quad (2.6)$$

Onde \mathbf{K}_e é a matriz de rigidez do elemento escrito em termos das coordenadas globais.

A parametrização

Na otimização topológica busca-se determinar a alocação ideal de um determinado material isotrópico no espaço, ou seja, determinar em quais pontos do espaço deve haver material e quais pontos devem permanecer vazios. Ou seja, pensa-se na representação geométrica da estrutura como uma representação em preto-branco de uma imagem pixelizada, onde cada pixel é um elemento finito. O pixel preto é onde há material e o pixel branco é uma região vazia. Dessa forma as variáveis de projeto são as densidades

relativas dos elementos e serão binárias ou contínuas, dependendo do modelo de material e método de otimização adotado.

O problema de otimização topológica é, portanto, um problema de programação inteira.

Adicionado a isso existe a restrição de volume, onde, para o domínio de referência Ω , busca-se determinar o subconjunto ótimo Ω^{mat} de pontos materiais tais que o conjunto E_{ad} de matrizes de elasticidade admissíveis consistem das matrizes que:

$$E_{ij} = 1 \Omega^{mat} E_{ij}^0, \quad 1 \Omega^{mat} = \begin{cases} 1 & \text{se } x \in \Omega^{mat} \\ 0 & \text{se } x \in \Omega \setminus \Omega^{mat} \end{cases} \quad (2.7)$$

$$\int_{\Omega} 1 \Omega^{mat} d\Omega = \text{Vol}(\Omega^{mat}) \leq V.$$

A última desigualdade expressa o limite, V , sobre a quantidade de material à disposição, de modo que otimização é limitada a um volume fixo. A matriz E_{ij}^0 é a matriz de elasticidade de um material *isotrópico* dado. Nesta formulação em E_{ad} é usada uma distribuição discreta para representar os módulos de elasticidade dos elementos.

Segundo Bendsoe e Sigmund (2007) a abordagem mais comumente usada para resolver este tipo de problema é de substituir as variáveis binárias por variáveis contínuas e, introduzir alguma forma de penalização na função objetivo que força a solução para valores binários. A otimização é então tratada como um problema de dimensionamento. Isso pode ser feito modificando-se a matriz de rigidez, de modo que ela dependa continuamente de uma certa função que é interpretada como uma densidade de material. Esta função torna-se, então, a variável de projeto.

O que se quer é que o resultado da otimização consista quase inteiramente de regiões com o máximo de material ou totalmente sem. Isto significa que os valores intermediários desta função de densidade artificial devem ser penalizados.

Uma possibilidade de penalidade é o chamado modelo SIMP, do inglês *Solid Isotropic Material with Penalization*, que aplica uma potência $p > 1$ sobre a função de densidade (que varia de zero a um) assim pode-se escrever:

$$E_{ij}(x) = \rho(x)^p E_{ij}^0, \quad p > 1, \quad (2.8)$$

$$\int_{\Omega} \rho(x) d\Omega \leq V; \quad 0 \leq \rho(x) \leq 1, \quad x \in \Omega$$

Aqui, a densidade $\rho(x)$ é a função artificial e E_{ij}^0 representa as propriedades de elasticidade do material de um dado material isotrópico. Refere-se a ρ como uma densidade de material, o volume é avaliado como $\int_{\Omega} \rho(x) d\Omega$. A densidade interpola o módulo de elasticidade do material entre 0 e E_{ij}^0 :

$$E_{ij}(\rho = 0) = 0, \quad E_{ij}(\rho = 1) = E_{ij}^0 \quad (2.9)$$

O que significa que se o resultado da otimização der densidade zero ou um, em todos os pontos, essa é uma solução binária desejada pois permite que a solução seja comparada com um modelo físico. Em SIMP opta-se por utilizar a penalidade $p > 1$, de modo que as densidades intermediárias são desfavoráveis no sentido de que a rigidez obtida é pequena em relação ao custo (volume) do material. Em outras palavras, através da especificação de um valor de p maior do que um torna não viável ter densidades intermediárias na concepção ótima.

Assim, a penalização é conseguida sem a utilização de qualquer esquema de penalização explícita. Para problemas onde a restrição de volume está ativa, a experiência mostra que a otimização realmente resulta em soluções binárias quando se escolhe p suficientemente grande. Segundo Bendsoe e Sigmund (2007) para a obtenção de soluções binárias geralmente é necessário que $p \geq 3$.

Além disso, provou-se existente uma solução binária ótima desde que a restrição de volume seja compatível, e que a penalidade p seja suficientemente grande que o problema de comprimento mínimo na forma *discreta*, existe uma solução de 0-1 de forma globalmente ótima, desde que o volume de restrição é compatível (Rietz 2001).

O uso do esquema de interpolação SIMP contorna o problema de programação inteira original (2.7). Ele converte o problema de programação inteira em um problema de dimensionamento, mas que chega a uma solução com as variáveis inteiras. Entretanto há um problema que o SIMP não resolve, é a falta de garantia de existência de soluções para o problema com carregamento distribuído.

Isto não é um problema apenas teórico. Ele também tem o efeito de tornar os resultados computacionais sensíveis à discretização da malha de elementos finitos. O esquema de interpolação não resolve esse problema diretamente, assim é necessário o uso de outras técnicas para garantir que a solução seja independente da escolha da malha.

2.3 O Método de Solução

A utilização de um método de interpolação como o SIMP permite converter o problema de topologia ótima em um problema de dimensionamento em um domínio fixo. Em comparação com muitos problemas de dimensionamento tradicionais, como por exemplo, placas, pórticos, etc., o presente problema difere no fato de que o número de variáveis de projeto é normalmente muito grande. Assim a eficiência do método de otimização é importante.

Na formulação usada nesta dissertação, a otimização topológica será tratada como uma otimização de densidade onde há apenas uma restrição (do volume total), além das restrições laterais.

2.3.1 A Função de Atualização das Densidades

A seguir são apresentadas as condições de otimalidade da densidade, ρ , para a formulação SIMP. Seguindo o critério de otimalidade padrão utilizado na otimização estrutural, a modelagem contínua simples com carregamento único dada pela equação (2.3) pode ser utilizada para gerar um esquema de atualização.

A chave é conceber uma função de iteração que, para uma solução previamente calculada, atualiza cada uma das variáveis de projeto, isto é, a densidade de cada elemento da discretização por elementos finitos, independente das atualizações em outros pontos, com base nas condições necessárias de otimalidade. A partir do problema de otimização (2.3) modificada para o caso da interpolação SIMP, Chega-se a:

$$\begin{aligned} & \min l(u) \\ & u \in U, \rho \\ \text{s. a. : } & a_E(u, v) = l(v) \text{ para qualquer } v \in U, \end{aligned} \quad (2.10)$$

$$E_{ij}(x) = \rho(x)^p E_{ij}^0,$$

$$\int_{\Omega} \rho(x) d\Omega \leq V; \quad 0 < \rho_{\min} \leq \rho \leq 1.$$

Foi introduzida uma densidade limite inferior ρ_{\min} , a fim de evitar singularidade do problema de equilíbrio na solução pelo Método dos Elementos Finitos. Será usado o

valor recomendado por Bendsoe e Sigmund (2007) de $\rho_{\min} = 10^{-3}$.

Sendo $\Lambda, \lambda^-(x), \lambda^+(x)$ os Multiplicadores de Lagrange das restrições de (2.10) tem-se:

$$\begin{aligned} \mathcal{L} = & \mathbf{l}(\mathbf{u}) - \{\mathbf{a}_E(\mathbf{u}, \bar{\mathbf{u}}) - \mathbf{l}(\bar{\mathbf{u}})\} + \Lambda \left(\int_{\Omega} \rho(x) d\Omega \right) - V \\ & + \int_{\Omega} \lambda^+(x)(\rho(x) - 1) d\Omega + \int_{\Omega} \lambda^-(x)(\rho_{\min} - \rho(x)) d\Omega, \end{aligned} \quad (2.11)$$

Sendo \mathcal{L} o Lagrangeano do problema de otimização, onde $\bar{\mathbf{u}}$ é o multiplicador de Lagrange para a restrição de equilíbrio. Note que $\bar{\mathbf{u}}$ pertence ao conjunto U de campos de deslocamento cinematicamente admissíveis. Sob a suposição de que $\rho \geq \rho_{\min} > 0$ (de modo que os campos de deslocamento são únicos), as condições de otimalidade no que diz respeito às variações do campo de deslocamento u dão que $\bar{u} = u$ enquanto a condição para ρ torna-se:

$$\frac{\partial E_{ij}}{\partial \rho} \varepsilon_i(u) \varepsilon_j(u) = \Lambda + \lambda^+ - \lambda^- \quad (2.12)$$

com as condições

$$\lambda^- \geq 0, \lambda^+ \geq 0, \lambda^-(\rho_{\min} - \rho(x)) = 0, \lambda^+(\rho(x) - 1) = 0 \quad (2.12 \text{ b})$$

Para densidades intermediarias $\rho_{\min} < \rho < 1$ as condições (2.4), podem ser escritas como

$$p\rho(x)^{p-1} E_{ij}^0 \varepsilon_i(u) \varepsilon_j(u) = \Lambda \quad (2.13)$$

que expressa que o termo de energia de deformação do lado esquerdo é constante e igual à Λ para todas as densidades intermediárias. Assim chega-se à seguinte função de atualização das densidades dos elementos:

$$\rho_{k+1} = \begin{cases} \max\{(1 - \zeta)\rho_k, \rho_{\min}\} & \text{se } \rho_k B_k^n \leq \max\{(1 - \zeta)\rho_k, \rho_{\min}\} \\ \min\{(1 + \zeta)\rho_k, 1\} & \text{se } \min\{(1 + \zeta)\rho_k, 1\} \leq \rho_k B_k^n \\ \rho_k B_k^n & \text{caso contrário} \end{cases} \quad (2.14)$$

A equação (2.14) é o esquema de atualização da densidade relativa que é baseado em uma heurística que tem por objetivo controlar a convergência do processo de otimização onde, ρ_k indica o valor da variável de densidade na iteração no passo k , e B_k é dado pela expressão:

$$B_k = \Lambda_k^{-1} p \rho(x)^{p-1} E_{ij}^0 \varepsilon_i(u_k) \varepsilon_j(u_k) \quad (2.15)$$

onde u_k é o campo de deslocamento na interação k , determinado a partir da equação de equilíbrio e dependente de ρ_k . Note-se que um ótimo (local) é alcançado se $B_k = 1$ para densidades ($\rho_{\min} < \rho < 1$).

A função de atualização (2.14) adiciona material em áreas com uma energia de deformação específica maior do que Λ (isto é, quando $B_k > 1$) e remove material quando a energia está abaixo deste valor ; isso só ocorre se a atualização não violar os limites de ρ .

A partir da integração (2.13) pode-se ver que Λ é proporcional (por um fator ρ) à densidade média de energia de deformação da parte da estrutura que é dada por valores intermediários da densidade.

A variável η em (2.14) é um parâmetro de ajuste e ζ um limite para a variação na densidade. Ambos η e ζ limitam as mudanças que podem acontecer em cada iteração e podem ser ajustados para melhorar a eficiência do método.

A atualização ρ_{k+1} depende implicitamente do Multiplicador de Lagrange Λ , assim Λ deve ser calculado por um método numérico, a fim de satisfazer a restrição do volume. O volume calculado a partir dos valores atualizados das densidades é uma função contínua e decrescente do Multiplicador de Lagrange Λ .

Além disso, o volume é estritamente decrescente nos intervalos que interessam, que são nos pontos (isto é, nos elementos finitos) onde as restrições nas densidades não estão ativas. Isto significa que se pode determinar de forma única o valor de Λ , usando um Método da Bissecção ou um Método de Newton.

Os valores de η e ζ são escolhidos por experimentação. Bendsoe e Sigmund (2007) recomendam como valores típicos para η e ζ os valores 0,5 e 0,2, respectivamente.

Observa-se que as condições (2.12 e 2.12b) implicam que a energia de deformação específica é constante em áreas de densidade intermediária, ao mesmo tempo que é pequena em regiões com uma densidade $\rho \geq \rho_{\min}$ e mais elevada em regiões com uma densidade igual a 1.

2.3.2 A implementação do método de otimização para o MEF.

Será feita, a seguir, uma descrição geral de como a formulação apresentada pode ser implementada no Método dos Elementos Finitos e ao final a figura (4) retratará esse processo.

Pré-processamento da geometria e do carregamento

-Escolhe-se um domínio de referência adequado que permite a definição de frações de superfície, limites fixos, etc.

-Escolhem-se as partes do domínio de referência que deve ser deixada como domínios sólidos ou espaço vazios.

-Os carregamentos e as condições de contorno (força e restrições de movimento no nós) são definidas.

-Constrói-se uma malha de elementos finitos sobre o domínio de referência. Esta malha deve ser fina o suficiente para descrever a estrutura final com uma resolução razoável, além de ser fina o suficiente para representar as regiões vazias e as que obrigatoriamente devem ser sólidas. A malha permanece inalterada durante todo o processo de otimização.

-A densidade uniforme, de cada elemento finito é uma variável de projeto.

Otimização

Para calcular a distribuição ideal da densidade sobre o domínio de referência usa a função de atualização para calcular as novas densidades a cada iteração. A estrutura do algoritmo é:

-Faz-se uma distribuição inicial das densidades, por exemplo, usa-se uma

distribuição homogênea de material.

-As sensibilidades (derivadas) do Lagrangeano são calculadas e utilizadas para atualizar a distribuição de massa da estrutura.

As iterações são:

-Para a distribuição de densidade atual, calcula-se pelo Método dos Elementos Finitos os deslocamentos e deformações resultantes.

-Computa-se o trabalho das forças internas.

Se há apenas uma melhora desprezível em relação à última distribuição de densidades, para-se as iterações.

Senão, continua-se

-Atualizam-se as densidades, com base na função dada em (2.14). Este passo possui um laço para o cálculo do valor do Multiplicador de Lagrange Λ para a restrição de volume.

-Volta-se ao início do laço.

Para o caso em que há partes da estrutura que são fixas (como sólido e/ou vazio) a função de atualização das densidades só deve ser invocada para as áreas que podem ser redesenhadas (reforçados ou esvaziadas).

Pós-processamento dos resultados

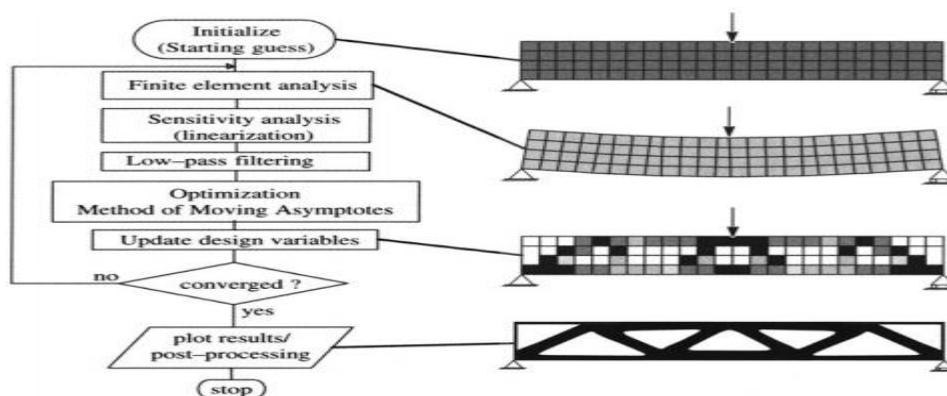
-Interpreta-se a distribuição ótima de material como se essa definisse a forma final do objeto.

Nessa dissertação usou-se como interpolação a penalidade do SIMP. A otimização topológica utilizando o SIMP com uma penalidade p alta, dá origem a um objeto muito bem definido, que consiste quase inteiramente de áreas totalmente com material, ou totalmente sem material, ou seja, muito pouco cinza.

É importante salientar que o algoritmo descrito pode ser implementado em qualquer tipo de malha de elementos finitos e qualquer tipo de domínio de referência Ω . Isto dá uma flexibilidade significativa para o método em termos de definir condições de contorno e partes fixas da estrutura.

No entanto trabalhar-se-á com domínios retangulares (em 2D), e com uma malha que consiste em quadrados o que simplifica a implementação e pode ser utilizado para acelerar a parte do processo de análise.

Figura 4: Diagrama de fluxo do cálculo para otimização topológica usando o método de distribuição de material.

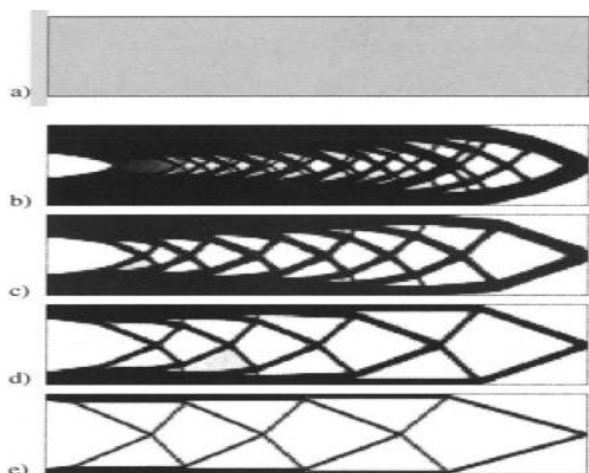


Fonte: Bendsoe e Sigmund, 2007

2.3.3 Sobre o cálculo do gradiente e métodos de programação matemática

Considerar-se-á o problema de otimização como um problema onde as variáveis de projeto são as densidades nos elementos. Assim o campo de deslocamentos é uma função destas variáveis de projeto. O campo de deslocamentos é calculado por meio da equação de equilíbrio e encontrar as derivadas dos deslocamentos em relação a densidade é denominado análise de sensibilidade.

Figura 5: A influência da fração de volume. Uma viga engastada discretizada por 6400 elementos quadrados e otimizada para frações de volume de b) 80% c) 60% d) 40% e e) de 20%. Para volumes pequenos a estrutura ótima resultante parece uma treliça.



Fonte: Bendsoe e Sigmund, 2007

O método de programação matemática a ser usado será um método análogo ou Método do Máximo Declive (*Steepest Descent*), nesse algoritmo o cálculo do gradiente é um elemento importante.

O Gradiente. O problema de otimização escrito para ser resolvido pelo MEF é:

$$\begin{aligned} & \min \mathbf{f}^T \mathbf{u} \\ & \mathbf{u}, \rho_e \\ \text{s. a.:} & \left(\sum_{e=1}^N \rho_e^p \mathbf{K}_e \right) \mathbf{u} = \mathbf{f}, \\ & \sum_{e=1}^N v_e \rho_e \leq V, \quad 0 < \rho_{\min} \leq \rho_e \leq 1, \quad e = 1, \dots, N, \end{aligned} \quad (2.16)$$

Para resolver este problema, através do algoritmo de programação matemática, primeiro ele é reescrito como um problema nas variáveis de projeto ρ :

$$\begin{aligned} & \min c(\rho_e) \\ & \rho_e \\ \text{s. a.:} & \sum_{e=1}^N v_e \rho_e \leq V, \quad 0 < \rho_{\min} \leq \rho_e \leq 1, \quad e = 1, \dots, N, \end{aligned} \quad (2.17)$$

Nesse caso a equação de equilíbrio é considerada como parte da função objetivo:

$$c(\rho_e) = \mathbf{f}^T \mathbf{u}, \quad \text{onde } \mathbf{u} \text{ é a solução de: } \left(\sum_{e=1}^N \rho_e^p \mathbf{K}_e \right) \mathbf{u} = \mathbf{f}$$

Quando os gradientes são exigidos pelo algoritmo de otimização utilizado para resolver (2.8), estes são calculados apenas em relação a ρ . Para funções que dependem também dos deslocamentos, derivadas podem ser obtidas pela regra da cadeia. Estas expressões irão então conter derivadas do deslocamento, o qual por sua vez pode ser obtida tomando a derivada da equação de equilíbrio $\mathbf{K}\mathbf{u} = \mathbf{f}$.

O método mais eficaz para o cálculo das derivadas é usar o método adjunto descrito a seguir, em que as derivadas do deslocamento não são calculadas

explicitamente. Nesse caso, o problema de otimização (2.17) pode-se reescrever a função $c(\rho)$ adicionando zero da função:

$$c(\rho) = \mathbf{f}^T \mathbf{u} - \tilde{\mathbf{u}}^T (\mathbf{K}\mathbf{u} - \mathbf{f}) \quad (2.18)$$

onde $\tilde{\mathbf{u}}$ é um vetor arbitrário, mas fixo. A partir deste, após o rearranjo de termos, obtém-se que

$$\frac{\partial c}{\partial \rho_e} = (\mathbf{f}^T - \tilde{\mathbf{u}}^T \mathbf{K}) \frac{\partial \mathbf{u}}{\partial \rho_e} - \tilde{\mathbf{u}}^T \frac{\partial \mathbf{K}}{\partial \rho_e} \mathbf{u} \quad (2.19)$$

Este pode por sua vez ser escrito como

$$\frac{\partial c}{\partial \rho_e} = -\tilde{\mathbf{u}}^T \frac{\partial \mathbf{K}}{\partial \rho_e} \mathbf{u} \quad (2.20)$$

assim, quando $\tilde{\mathbf{u}}$ satisfaz a equação transposta:

$$\mathbf{f}^T - \tilde{\mathbf{u}}^T \mathbf{K} = \mathbf{0} \quad (2.21)$$

Esta última equação é a equação de equilíbrio e por comparação vê-se diretamente que $\tilde{\mathbf{u}} = \mathbf{u}$. Além disso, a forma da matriz de rigidez significa que as derivadas da função objetivo $c(\rho)$ para problema (2.17) é de forma particularmente simples:

$$\frac{\partial c}{\partial \rho_e} = -p\rho_e^{p-1} \mathbf{u}^T \mathbf{K}_e \mathbf{u} . \quad (2.22)$$

Assim as derivadas para o cálculo do gradiente podem ser obtidas a partir da solução do MEF. Além disso, percebe-se que as derivadas são locais no sentido de que a derivada envolve apenas as informações ao nível do elemento: no entanto, há um efeito de outras variáveis implícitas no deslocamento \mathbf{u} . Finalmente, vê-se que a derivada é negativa para todos os elementos, de modo que a intuição física, é confirmada pelo fato de que o material adicional em qualquer elemento reduz a flexibilidade o que torna a estrutura mais rígida

3. O PROGRAMA

Será apresentada uma implementação de um código feito em Matlab® visando a otimização topológica de estruturas estaticamente carregadas. A programação parte de um código desenvolvido por Sigmund (2001) e disponível na internet. O código contém 99 linhas de entrada, onde são subdivididas por linhas de comentários para que se possa identificar o que ocorre dentro do código.

Nessa subdivisão, pode-se observar que ficaram definidas 36 linhas para o programa principal, onde ocorre todo o processo de otimização, 12 linhas para o método de otimização baseado na função de atualização, mais outras 16 linhas de código baseado num filtro para tornar a solução independente da malha e finalizando, 35 linhas de um código básico para cálculos de elementos finitos.

Se retiradas as linhas de comentários e as linhas que visam a produção e análise de elementos finitos, tem-se apenas 49 linhas de comando necessárias para resolver os problemas de otimização topológica.

3.1 A implementação do programa em Matlab®

O programa principal é chamado a partir do *prompt* de comando do Matlab® pela linha

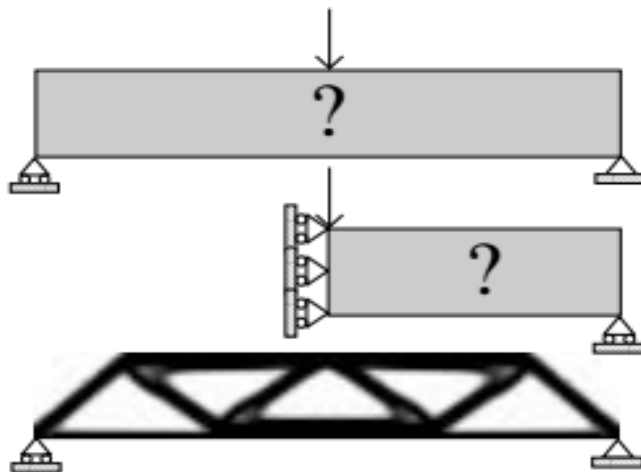
topsemfiltro(nelx,nely,volfrac)

Onde **nelx** e **nely** são o número de elementos nas direções horizontal e vertical, respectivamente, **volfrac** é a fração de volume. Esses são os parâmetros principais da otimização, mas dentro do programa define-se também outros elementos, em particular o carregamento e as condições de apoio.

Durante o processo de otimização topológica, a cada iteração o código gera uma imagem da distribuição de densidade atual.

As condições de contorno usadas nos exemplos subsequentes correspondem à metade de uma viga simplesmente apoiada (Figura 6). A carga é aplicada verticalmente no canto superior esquerdo e há condições de contorno de simetria ao longo da borda esquerda e a estrutura é suportada verticalmente e livre horizontalmente no canto inferior direito.

Figura 6: Otimização Topológica da viga. **Topo:** domínio de projeto completo, **Meio:** domínio da metade do design com condições de contorno de simetria e **Inferior:** resultado da Otimização Topológica da viga (duas metades).



Fonte: Bendsoe e Sigmund, 2007

3.2 Programa Principal: linhas 1 a 36

O código abaixo mostra o programa principal que se encontra nas linhas 1 a 36, iniciando na (linha 1) onde tem-se informações sobre a origem do código, na (linha 2) declara-se a função principal do código na qual será realizada todo processo de otimização, seguindo na (linha 4) inicializa as densidades (chamadas nesse código de x) onde o material é distribuído uniformemente no domínio. Depois de informar outros comandos de inicialização até a (linha 10) a iteração principal começa com uma chamada para a sub-rotina de Elementos Finitos (linha 12) que retornará valores dos deslocamentos U dos nós como será mostrado nas linhas de comando abaixo.

```

1- Otimização Topológica. Baseado em Sigmund (2000)
2- function topsemfiltro(nelx,nely,volfrac)
3- % Inicializações
4- x(1:nely,1:nelx) = volfrac;
5- loop = 0;
6- change = 1.;
7- % Início da iterações
8- while change > 0.01
9- loop = loop + 1;
10- xold = x;
11- % Solução pelo MEF
12- [U]=FE(nelx,nely,x);

```

Uma vez que a matriz de rigidez é a mesma para todos os elementos, da matriz de rigidez do elemento é chamada pela sub-rotina apenas uma vez na (linha 14). Na (linha 15) é inicializada uma variável como sendo ($c = 0$), que permitirá a atualização da flexibilidade (compliance) da estrutura.

```
13- % Função objetivo e gradiente
14- [KE] = lk;
15- c = 0.;
```

A matriz de rigidez do elemento, nas coordenadas locais, está codificada na parte final do programa (linhas 86 a 99) e consiste basicamente na transcrição dos coeficientes do elemento quadrilátero. A matriz de rigidez 8×8 do elemento quadrilátero de 4 nós foi obtida analiticamente. O módulo Young E e coeficiente de Poisson pode ser alterado nas linhas 88 e 89 respectivamente.

```
86- %%%%%%%%%%% Matriz de Rigidez do Elemento Finito %%%%%%%%%%%
87- function [KE]=lk
88- E = 1.;
89- nu = 0.3;
90- k=[ 1/2-nu/6    1/8+nu/8  -1/4-nu/12  -1/8+3*nu/8  ...
91-    -1/4+nu/12  -1/8-nu/8    nu/6        1/8-3*nu/8];
92- KE = E/(1-nu^2)*[ k(1) k(2) k(3) k(4) k(5) k(6) k(7) k(8)
93-                  k(2) k(1) k(8) k(7) k(6) k(5) k(4) k(3)
94-                  k(3) k(8) k(1) k(6) k(7) k(4) k(5) k(2)
95-                  k(4) k(7) k(6) k(1) k(8) k(3) k(2) k(5)
96-                  k(5) k(6) k(7) k(8) k(1) k(2) k(3) k(4)
97-                  k(6) k(5) k(4) k(3) k(2) k(1) k(8) k(7)
98-                  k(7) k(4) k(5) k(2) k(3) k(8) k(1) k(6)
99-                  k(8) k(3) k(2) k(5) k(4) k(7) k(6) k(1)];
```

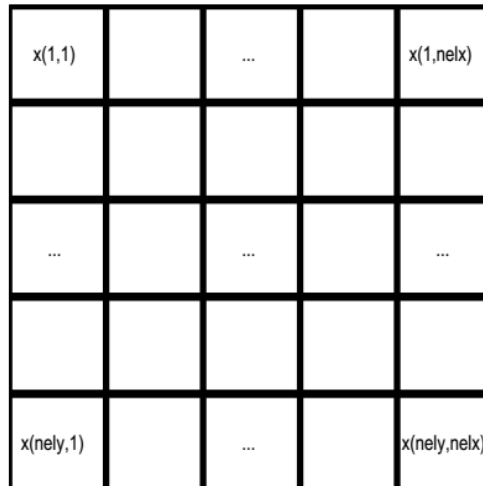
Após a função objetivo, ainda no programa principal, segue-se um laço (duplo), da linha 16 a 24, que varre cada um dos elementos e calcula a função objetivo e o gradiente. Nas linhas 18 a 19 são calculadas duas variáveis n_1 e n_2 , que são os números dos nós superior à esquerda e direita do elemento nas coordenadas globais. Essas variáveis são utilizadas para mapear o deslocamento de \mathbf{U}_e do sistema de coordenadas globais para o sistema local.

```
16- for ely = 1:nely
17-     for elx = 1:nelx
18-         n1 = (nely+1)*(elx-1)+ely;
19-         n2 = (nely+1)* elx +ely;
20-         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);
21-         c = c + x(ely,elx)*Ue'*KE*Ue;
22-         dc(ely,elx) = x(ely,elx)*Ue'*KE*Ue;
23-     end
```

24- end

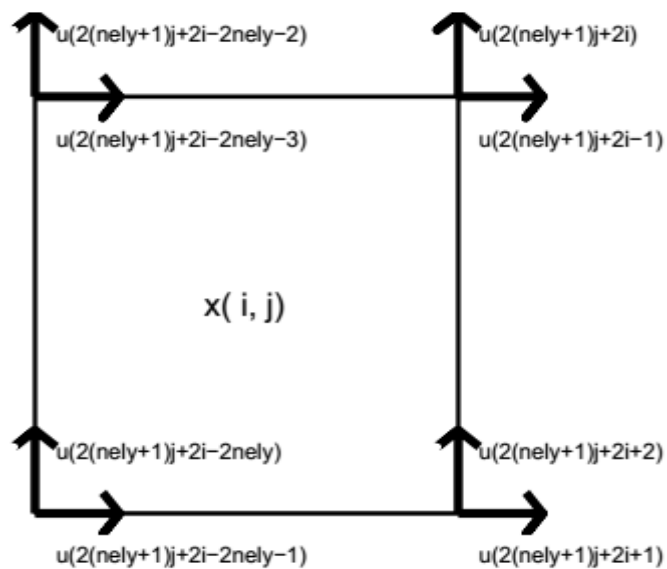
As seguintes figuras retratam tanto como ficará a ordem dos índices da matriz de rigidez, quanto a relação entre as coordenadas do sistema local para o sistema global.

Figura 7: A ordem do índice dos elementos finitos.



Fonte: Rietz, 2000

Figura 8: Relação entre as coordenadas do sistema local e global.



Fonte: Rietz, 2000

A linha 28 chama a função de atualização das densidades OC. A função objetivo, bem como outros parâmetros calculados até aqui são impressos pelas linhas 30 a 33 e o resultado da distribuição da densidade é plotada pelos comandos internos do Matlab® mostrados na linha 35. A verificação necessária que informa se o ciclo principal será finalizado é se a maior mudança nas variáveis de projeto é inferior a 1%, mudança qual é determinada na linha 30, caso contrário, todos os passos anteriores deverão ser repetidos. E isso fecha o programa principal.

```

27- % Atualização das Densidades
28- [x] = OC(nelx,nely,x,volfrac,dc);
29- % Mostra os resultados
30- change = max(max(abs(x-xold)));
31- disp([' It.: ' sprintf('%4i',loop) ' Obj.: ' sprintf('%10.4f',c) ...
32-      ' Vol.: ' sprintf('%6.3f',sum(sum(x))/(nelx*nely)) ...
33-      ' ch.: ' sprintf('%6.3f',change)])
34- % Plota a figura em escala de cinza
35- colormap(gray);imagesc(-x);axis equal;axis tight; axis off;pause(1e-6);
36- end

```

3.3 Atualização das densidades: linhas 37 a 48

A finalidade das linhas 37 a 48 são de atualizar as variáveis de projeto, isto é, as densidades nos elementos respeitando-se as restrições laterais e a restrição de volume. Sabendo-se que o volume do material ($\text{sum}(\text{sum}(x_{\text{new}}))$) é uma função monotonamente decrescente dos multiplicadores de Lagrange (**lag**). O Método da Bissecção é usado nas linhas 40 a 48, e tem como objetivo encontrar um multiplicador de Lagrange que possa satisfazer a restrição de volume.

```

37- %%%%%%%%%%% Atualização das Densidades %%%%%%%%%%%
38- function [xnew]=OC(nelx,nely,x,volfrac,dc)
39- l1 = 0; l2 = 100000; move = 0.2;
40- while (l2-l1 > 1e-4)
41-     lmid = 0.5*(l2+l1);
42-     xnew = max(0.001,max(x-move,min(1.,min(x+move,x.*sqrt(-dc./lmid)))));
43-     if sum(sum(xnew)) - volfrac*nelx*nely > 0;
44-         l1 = lmid;
45-     else
46-         l2 = lmid;
47-     end
48- end

```

O Método da Bissecção é inicializado para valores do multiplicador de Lagrange de l_1 bem pequeno e um valor de l_2 muito grande (linha 39). No método o intervalo que

cerca o multiplicador de Lagrange será repetidamente reduzido à metade até que seu tamanho obedeça ao critério de convergência dado na linha 40.

3.4 O Código do Método dos Elementos Finitos

O código de elementos finitos é escrito nas linhas 65 a 85, e consiste basicamente em montar a Matriz de Rigidez Global, montar o vetor de carregamento externo, remover as coordenadas relativas aos apoios e resolver o sistema linear resultante. O programa faz uso da função *sparse* encontrada no Matlab® para inicializar a matriz de rigidez global K e o vetor das forças externas F .

```
65- %%%%%%%%%%% A Resolução do MEF %%%%%%%%%%%
66- function [U]=FE(nelx,nely,x)
67- [KE] = lk;
68- K = sparse(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1));
69- F = sparse(2*(nely+1)*(nelx+1),1); U = zeros(2*(nely+1)*(nelx+1),1);
```

A montagem da matriz de rigidez global é dada por um laço duplo que varre todos os elementos (linhas 70 a 77). Como já descrito no programa principal, as variáveis n_1 e n_2 tem o mesmo significado e são usadas para inserir os coeficientes da matriz de rigidez elementar nos lugares certos da matriz de rigidez global.

```
70- for elx = 1:nelx
71-   for ely = 1:nely
72-     n1 = (nely+1)*(elx-1)+ely;
73-     n2 = (nely+1)*elx +ely;
74-     edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1; 2*n2+2; 2*n1+1; 2*n1+2];
75-     K(edof,edof) = K(edof,edof) + x(ely,elx)*KE;
76-   end
77- end
```

As linhas 78 a 82 definem o carregamento, guardado no vetor F , nesse caso apenas uma força (para baixo, por isso o sinal negativo) no nó de coordenadas globais (2,1), e as restrições ao deslocamento, que representam os apoios da viga.

```
78- % Definição do carregamento e dos apoios
79- F(2,1) = -1;
80- fixeddofs = union(1:2:2*(nely+1), 2*(nelx+1)*(nely+1));
81- alldofs = 1:2*(nely+1)*(nelx+1);
82- freeddofs = setdiff(alldofs, fixeddofs);
```

Tanto os nós quanto os elementos são numerados da coluna da esquerda para a direita. Além disso, cada nó tem dois graus de liberdade (horizontal e vertical), o comando de $F(2,1) = -1$ na linha 79 aplica uma força unitária vertical no canto superior esquerdo. Os apoios são implementados através da eliminação das coordenadas fixas ao se resolver o sistema linear. Os comandos em Matlab® para a solução do sistema linear estão descritos nas linhas 83 a 85 abaixo:

```
83- % Solução do Sistema linear
84- U(freedofs,:) = K(freedofs,freedofs) \ F(freedofs,:);
85- U(fixeddofs,:)= 0;
```

Onde *freedofs* indicam as coordenadas livres. É mais direto definir quais são as coordenadas fixas (*fixeddofs*), e posteriormente, as coordenadas livres. *Freedofs* são calculadas pelo comando *setdiff* que encontra os graus de liberdade livres como a diferença entre todos os graus de liberdade e os graus de liberdade fixos, comando encontrado na (linha 82).

3.5 Um exemplo

A figura 9 mostra o resultado da otimização da viga bi apoiada (apenas metade da viga). A grande quantidade de áreas cinzas indicam que nessas áreas haveria menos material. Apesar de ser uma solução ótima (no sentido de maximizar a rigidez) não é uma estrutura prática de se realizar fisicamente.

Figura 9: Metade da viga bi apoiada com carregamento central. Foi utilizado o código **topsemfiltro(60,20,0.5)**



3.6 Incluindo as penalidades

Para que o processo de otimização evite deixar regiões parcialmente cheias de material, o método SIMP propõe a inclusão da penalidade para regiões parcialmente cheias. A seguir descrevem-se os locais onde o programa deverá ser alterado.

Primeiramente acrescenta-se na linha 2 mais um parâmetro na função para ser chamada pelo *prompt* do Matlab®:

```
2- function topsemfiltro(nelx,nely,volfrac,penal)
```

Assim a chamada ao programa fica:

```
topsemfiltro(nelx,nely,volfrac,penal)
```

onde *penal* é a variável que controla a penalidade.

Outra função que é afetada pela penalidade é a função FE que constrói a matriz de rigidez global para a solução do sistema linear. Acrescentando-se a penalidade na linha 75 tem-se:

```
75- K(edof,edof) = K(edof,edof) + x(ely,elx)^penal*KE;
```

Consequentemente na definição da função (linha 66) e na chamada à função no programa principal (linha 12), deve-se colocar a variável **penal**. Assim:

```
12- [U]=FE(nelx,nely,x,penal);
66- function [U]=FE(nelx,nely,x,penal)
```

Finalmente, a penalidade aparece no cálculo da função objetivo (linha 21) e o gradiente (linha 22), ambos no programa principal. As linhas modificadas ficam:

```
21- c = c + x(ely,elx)^penal*Ue'*KE*Ue;
22- dc(ely,elx) = penal*x(ely,elx)^(penal-1)*Ue'*KE*Ue;
```

Exemplo:

A figura 10 mostra o resultado da viga bi-apoiada com condições de simetria quando se coloca uma penalidade cúbica ($p=3$). Nota-se praticamente a inexistência de regiões cinza. Essa é uma viga fisicamente realizável.

Figura 10: Metade da viga bi apoiada com carregamento central. Foi utilizado o código `topsemfiltro(60,20,0.5,3)`



Fonte: autor

Esta topologia não é fisicamente realizável, pois como não foi utilizado o filtro, aparece na topologia a presença de “tabuleiros de xadrez” (checkerboard) que dificulta a interpretação do resultado

Dependendo da malha escolhida pode-se ter resultados diferentes. Para contornar esse problema introduz-se uma mudança na atualização das densidades, que elimina esse problema.

3.7 Tornando a otimização independente da malha

Para que a otimização fique independente da malha, a proposta de Sigmund (2007), é de trocar o gradiente que guia a otimização por uma média ponderada dos gradientes dos elementos finitos vizinhos dentro de um raio chamado de r_{min} , onde o peso depende da distância entre o elemento vizinho e o elemento atual, quanto maior a distância, menor o peso. Bendsoe (2007) chama esse procedimento de “filtro”.

O filtro é implementado como segue:

$$\frac{\widehat{\partial c}}{\partial x_e} = \frac{1}{x_e \sum_{f=1}^N \widehat{H}_f} \sum_{f=1}^N \widehat{H}_f x_f \frac{\partial c}{\partial x_f} \quad (3.1)$$

onde o peso \widehat{H}_f é calculado como:

$$\widehat{H}_f = r_{min} - \text{dist}(e, f), \quad (3.2)$$

$$\{f \in N \mid \text{dist}(e, f) \leq r_{min}\}, \quad e = 1, \dots, N,$$

o operador $\text{dist}(e, f)$ é definido como a distância entre o centro do elemento e e o centro do elemento f . O peso \widehat{H}_f é zero fora da área do filtro. O peso decai linearmente com a distância do elemento f . Em vez do gradiente original, as derivadas modificadas são utilizadas na função de atualização.

Para implementar o filtro foram feitas as seguintes modificações:

- a) Foi acrescentado um parâmetro a mais nos argumentos da função, que é o programa principal, `topsemfiltro`, que foi renomeado, ainda para `top` ficando a linha 2

```
2- function top(nelx,nely,volfrac,rmin)
```

- b) Foram acrescentadas as linhas 25 e 26. A linha 26 chama a função `check` que recalcula o gradiente da função objetivo em relação às densidades, ou seja, aplica o filtro no gradiente.

```
25- % Suavização, filtragem do gradiente
26- [dc] = check(nelx,nely,rmin,x,dc);
```

- c) A implementação das fórmulas (3.1) e (3.2), na forma da função `check`, são as linhas 49 a 64 conforme abaixo.

```
49- %%%%%%%%%%% Filtro para independência de malha %%%%%%%%%%%
50- function [dcn]=check(nelx,nely,rmin,x,dc)
51- dcn=zeros(nely,nelx);
52- for i = 1:nelx
53-     for j = 1:nely
54-         sum=0.0;
55-         for k = max(i-floor(rmin),1):min(i+floor(rmin),nelx)
56-             for l = max(j-floor(rmin),1):min(j+floor(rmin),nely)
57-                 fac = rmin-sqrt((i-k)^2+(j-l)^2);
58-                 sum = sum+max(0,fac);
59-                 dcn(j,i) = dcn(j,i) + max(0,fac)*x(l,k)*dc(l,k);
60-             end
```

```

61-     end
62-     dcn(j,i) = dcn(j,i) / (x(j,i) * sum);
63-     end
64- end

```

Nem todos os elementos no domínio são pesquisados a fim de encontrar os elementos que se encontram dentro do raio r_{min} , mas apenas aqueles dentro de um quadrado com o lado comprimentos duas vezes o tamanho de r_{min} em torno do elemento considerado.

Selecionar r_{min} menor que 1 na chamada da função `check` fará com que a filtragem não ocorra.

As figuras 11 e 12 mostram o resultado do mesmo problema resolvido usando-se malhas diferentes de 200x40 e 100x40 respectivamente

Figura 11: Resultado da Otimização Topológica da viga bi apoiada utilizando uma malha de 200x40 elementos, e um filtro de 1.5.



Fonte: autor

Figura 12: Resultado da Otimização Topológica da viga bi apoiada utilizando uma malha de 100x20 elementos, e um filtro de 1.5. A imagem mostra o mesmo perfil otimizado, mas com uma estrutura interna que parece menos refinada e com membros mais grossos devido à menor resolução da malha.



Fonte: autor

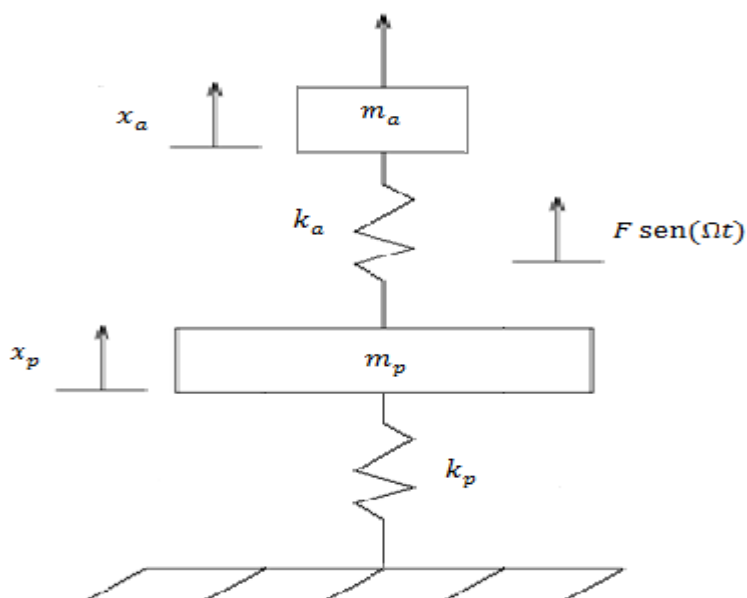
4. OTIMIZAÇÃO TOPOLÓGICA PARA O CASO DINÂMICO

Considerando que a proposta da dissertação é a otimização topológica de um Absorvedor Dinâmico de Vibrações, esse capítulo da dissertação aborda a inclusão de uma força dinâmica na modelagem. Segue então os passos utilizados para chegar na montagem de uma topologia ótima de um ADV.

Inicialmente será apresentada a fundamentação teórica do modelo dinâmico, seguindo da montagem da matriz de massa elementar, depois, as alterações realizadas no código a fim de construir essa estrutura ótima juntamente com os resultados obtidos, e finalizando, mostram-se alguns dos resultados que foram obtidos por esse projeto.

4.1 Modelagem para o caso Dinâmico

A figura (13) ilustra um sistema principal de massa m_p com uma massa acoplada, o absorvedor, de massa m_a , formando um sistema de dois graus de liberdade com uma força periódica aplicada ao sistema primário.



Fonte: Autor

O sistema de equações a seguir descreve matematicamente esse problema:

$$\begin{aligned} m_p \ddot{x}_p + (k_p - k_a)x_p - k_a x_a &= F \text{ sen}(\Omega t) \\ m_a \ddot{x}_a - k_a x_p + k_a x_a &= 0 \end{aligned} \quad (4.1)$$

Onde k_p e k_a são a rigidez do sistema principal e do sistema do absorvedor respectivamente, x_p e x_a são os deslocamentos do sistema principal e do absorvedor em relação ao ponto de equilíbrio, \ddot{x}_p e \ddot{x}_a são as derivadas segunda em relação ao tempo.

Admitindo-se que o sistema vibre com a mesma frequência Ω da força excitadora tem-se que:

$$\begin{aligned} x_p(t) &= X_p \text{ sen}(\Omega t) \\ x_a(t) &= X_a \text{ sen}(\Omega t) \end{aligned} \quad (4.2)$$

Substituindo-se (4.2) em (4.1) chega-se ao sistema algébrico (4.3), a partir do qual pode-se determinar a amplitude máxima de vibração do sistema principal x_p e do amortecido x_a .

$$\begin{bmatrix} k_p + k_a - \Omega^2 m_p & -k_a \\ -k_a & k_a - \Omega^2 m_a \end{bmatrix} \begin{Bmatrix} X_p \\ X_a \end{Bmatrix} = \begin{Bmatrix} F \\ 0 \end{Bmatrix} \quad (4.3)$$

A solução de (4.3) é:

$$\begin{aligned} X_p &= \frac{(k_a - \Omega^2 m_a)F}{(k_p + k_a - \Omega^2 m_p)(k_a - \Omega^2 m_a) - k_a^2} \\ X_a &= \frac{k_a F}{(k_p + k_a - \Omega^2 m_p)(k_a - \Omega^2 m_a) - k_a^2} \end{aligned} \quad (4.4)$$

Chamando-se de ω_p à frequência natural do sistema principal na ausência do absorvedor $\omega_p = \sqrt{\frac{k_p}{m_p}}$ e ω_a a frequência do amortecido na ausência do sistema principal $\omega_a = \sqrt{\frac{k_a}{m_a}}$ as equações (4.4) podem ser reescritas como:

$$X_p = \frac{\left[1 - \left(\frac{\Omega}{\omega_a}\right)^2\right] F/k_p}{\left[1 + \frac{m_a}{m_p \left(\frac{\omega_a}{\omega_p}\right)^2} - \left(\frac{\Omega}{\omega_p}\right)^2\right] \left[1 - \left(\frac{\Omega}{\omega_a}\right)^2\right] - m_a/m_p (\omega_a \omega_p)^2} \quad (4.5)$$

$$X_a = \frac{F/k_p}{\left[1 + \frac{m_a}{m_p \left(\frac{\omega_a}{\omega_p}\right)^2} - \left(\frac{\Omega}{\omega_p}\right)^2\right] \left[1 - \left(\frac{\Omega}{\omega_a}\right)^2\right] - m_a/m_p (\omega_a \omega_p)^2}$$

A partir da equação (4.5) pode-se observar que quando $\omega_a = \Omega$, a amplitude de vibração X_p é nula, isto é, nessa frequência o absorvedor cumpre sua função.

Nesse sentido o objetivo é criar a topologia do absorvedor que será usada para “atenuar” as vibrações do sistema primário.

4.2 Montagem da matriz de massa elementar

Como na modelagem dinâmica é necessário se conhecer as frequências naturais, então é necessário se calcular a matriz de massa que, junto com a matriz de rigidez, dependerá das densidades dos elementos. Apresenta-se a seguir o desenvolvimento da matriz de massa elementar que será acrescentada no código de otimização topológica.

O tipo de elemento finito utilizado no código de otimização topológica é o isoparamétrico de quatro nós, portanto parte-se desse mesmo elemento para a construção da matriz de massa elementar.

As funções de forma do elemento quadrilátero isoparamétrico de quatro nós são (Kim e Sankar, 2007):

$$N_1(s, t) = \frac{1}{4}(1 - s)(1 - t)$$

$$N_2(s, t) = \frac{1}{4}(1 + s)(1 - t)$$

$$N_3(s, t) = \frac{1}{4}(1 + s)(1 + t)$$

$$N_4(s, t) = \frac{1}{4}(1-s)(1+t)$$

Assim a posição de um ponto qualquer, $x(s, t)$, do elemento pode ser calculado por:

$$x(s, t) = [N_1 \ N_2 \ N_3 \ N_4] \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{Bmatrix}$$

$$y(s, t) = [N_1 \ N_2 \ N_3 \ N_4] \begin{Bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{Bmatrix}$$

Onde os vetor $\{x\}$ e $\{y\}$ representa a posição, na direção horizontal e vertical respectivamente, dos nós do elemento finito.

A matriz de massa elementar é dada por (Nascimento, 2005):

$$M^e_{ij} = \rho \int_{-1}^1 \int_{-1}^1 N_i^* N_j^* |J| ds dt \quad (4.6)$$

Onde:

$$|J| = \det \begin{bmatrix} \frac{\partial x}{\partial s} & \frac{\partial x}{\partial t} \\ \frac{\partial y}{\partial s} & \frac{\partial y}{\partial t} \end{bmatrix} \quad (4.7)$$

Onde $|J|$ representa o Jacobiano e N^* pode ser escrito por:

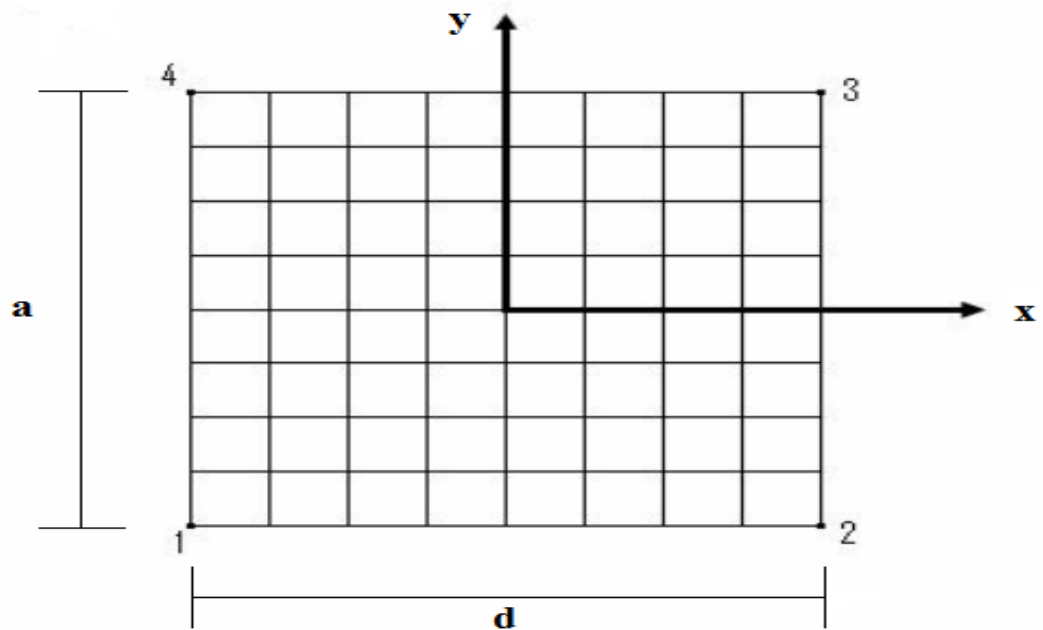
$$N^* = [N_1 \ 0 \ N_2 \ 0 \ N_3 \ 0 \ N_4 \ 0]$$

Considerando o elemento quadrilátero, tem-se:

$$x_1 = x_4, \ x_2 = x_3, \ x_2 = x_1 + d, \ y_1 = y_2, \ y_4 = y_3, \ y_4 = y_1 + a \quad (4.8)$$

Conforme mostra a figura 14:

Figura 14: Elemento quadrilátero de 4 nós



Fonte: Autor

Substituindo (4.8) e (4.7) em (4.6) chega-se a:

$$M_{11}^e = \frac{ad}{9} \rho$$

$$M_{13}^e = M_{31}^e = \frac{ad}{18} \rho$$

$$M_{15}^e = M_{51}^e = \frac{ad}{36} \rho$$

$$M_{17}^e = M_{71}^e = \frac{ad}{18} \rho$$

$$M_{33}^e = \frac{ad}{9} \rho$$

$$M_{35}^e = M_{53}^e = \frac{ad}{18} \rho$$

$$M_{37}^e = M_{73}^e = \frac{ad}{36} \rho$$

$$M_{55}^e = \frac{ad}{9} \rho$$

$$M_{57}^e = M_{75}^e = \frac{ad}{18} \rho$$

$$M^e_{77} = \frac{ad}{9}\rho$$

E os demais elementos formadores da matriz são nulos.

Assim a matriz de massa elementar pode tomar a seguinte forma de representação de seus termos:

$$M^e = ad\rho \begin{bmatrix} 1/9 & 0 & 1/18 & 0 & 1/36 & 0 & 1/18 & 0 \\ 0 & 1/9 & 0 & 1/18 & 0 & 1/36 & 0 & 1/18 \\ 1/18 & 0 & 1/9 & 0 & 1/18 & 0 & 1/36 & 0 \\ 0 & 1/18 & 0 & 1/9 & 0 & 1/18 & 0 & 1/36 \\ 1/36 & 0 & 1/18 & 0 & 1/9 & 0 & 1/18 & 0 \\ 0 & 1/36 & 0 & 1/18 & 0 & 1/9 & 0 & 1/18 \\ 1/18 & 0 & 1/36 & 0 & 1/18 & 0 & 1/9 & 0 \\ 0 & 1/18 & 0 & 1/36 & 0 & 1/18 & 0 & 1/9 \end{bmatrix}$$

Que é a matriz de massa elementar implementada no código de otimização topológica.

4.3 A Otimização Topológica para o caso Dinâmico e alterações no código Matlab®.

Para implementação do código de otimização topológica, o objetivo é basicamente fazer com que $\lambda = \Omega^2$, onde λ é o menor autovalor da matriz global $[M]^{-1} [K]$, que é o quadrado da menor frequência natural, e Ω a frequência de vibração do sistema primário que se deseja absorver. Assim a função objetivo implementada no código será $c = (\lambda - \Omega^2)^2$.

Para penalizar as densidades intermediárias usou-se a função $\rho_{ij} = (1 - \rho_{ij})$ para cada elemento finito, a função objetivo se modificará e tornará:

$$c = p_1(\lambda - \Omega^2)^2 - p_2 \sum_i \sum_j [\rho_{ij}(1 - \rho_{ij})]$$

sendo $p_1 = 10^8$ e $p_2 = 0,02$ pesos obtidos por tentativa e erro, onde foram realizado vários testes com valores diferentes até chegar a estes.

Assim, do ponto de vista da otimização, o problema é enunciado conforme a equação (4.9):

$$\min \left(p_1 (\lambda - \Omega^2)^2 - p_2 \sum_i \sum_j \left[\rho_{ij} (1 - \rho_{ij}) \right] \right) \quad (4.9)$$

\mathbf{u}, ρ_e

sujeita a : $(\mathbf{K} - \Omega^2 \mathbf{M})\mathbf{u} = \mathbf{f}$,

$$\sum_{e=1}^N v_e \rho_e \leq V, \quad 0 < \rho_{\min} \leq \rho_e \leq 1, \quad e = 1, \dots, N,$$

Onde a primeira restrição refere-se à solução do sistema global. As variáveis \mathbf{K} e \mathbf{M} são as matrizes de rigidez e massa globais respectivamente, \mathbf{u} e \mathbf{f} os vetores da amplitude de vibração e a amplitude da força excitadora com frequência Ω .

Para atualização das densidades, o gradiente é obtido numericamente pela expressão:

$$\frac{\partial c}{\partial \rho} = \frac{c(\rho + \Delta\rho) - c(\rho)}{\Delta\rho} . \quad (4.10)$$

É importante observar que a cada iteração do algoritmo será necessário avaliar o gradiente da função objetivo em relação à densidade de cada elemento finito.

Para que isso seja feito, é necessário se montar a matriz global, resolver o sistema, e determinar o menor autovalor para todos os elementos finitos.

Esse passo torna o processo de otimização extremamente lento, devido à quantidade de avaliações dos autovalores.

As alterações apresentadas a seguir, foram realizadas no código descrito no Capítulo 3, o qual foi utilizado para casos estáticos, e agora tem o objetivo de trabalhar com casos dinâmicos.

Logo no início do código, mais especificamente na linha (2), onde a declaração da função principal do código, na qual será realizada todo processo de otimização, há o acréscimo de mais uma variável, variável essa o omega :

```
2- function topDin(nelx,nely,volfrac,penal,rmin,omega)
```

Que será utilizada no cálculo da função objetivo como mostra a linha de comando (24) verificada a seguir:

```
24- c = p1*(omega^2-autovalor)^2+p2*sum(sum(x.*(1-x)));
```

Outra alteração realizada no código, consiste na inicialização de um vetor para guardar o menor autovalor de cada iteração, e um contador na linha (9), `iii=1`, que tem como função auxiliar o cálculo dos autovalores da matriz nas linhas (30-31) como será visto posteriormente.

Seguindo o código tem-se os valores de penalização das densidades intermediárias, ou seja, os pesos utilizados nas linhas (10-12):

```
9- iii=1;
10- deltax = 0.2;
11- p1 = 1e+8;
12- p2 = 0.02;
```

Seguindo com as alterações, na linha (21) posteriormente a criação da variável que representa a matriz de rigidez elementar, foi criada uma variável para a matriz de massas elementar, a $[ME] = mk$, como pode ser visto a seguir.

```
20- [KE] = lk;
21- [ME] = mk;
```

Continuando com as alterações, na linha (101) temos a criação da variável `aaa`, que realiza a função de calcular os autovalores do produto da inversa das matrizes de rigidez e massa respectivamente, juntamente com o comando do Matlab® `sort`, que ordena o vetor de autovalores em ordem crescente.

Já na linha (102) temos a variável que aloca o menor autovalor calculado, como é mostrado:

```
aaa = sort(eigs(M\K));
autovalor = aaa(1);
```

Um critério de parada foi inserido na linha (34), com intuito de limitar o número de iterações para os testes que foram realizados, uma vez que essa formulação mostrou que frequentemente a otimização entra em um laço infinito:

```
33- iii=iii+1;
34- if iii>50
35-     break
36- end
```

A próxima modificação se deu apenas no final do código, nas linhas (114-131) foi criada uma nova função para o cálculo da matriz de massa elementar, onde na linha (115) os valores encontrados $m = [1/9 \ 1/18 \ 1/36]$, na linha (116) é criada a variável para a matriz identidade de dimensão 8x8.

Já na linha (117) inicia-se a montagem de forma específica para a matriz de massa elementar, onde é originado o primeiro elemento como é mostrado; $ME = m(1) * ME$.

Nas linhas (119 a 131) o código alterado inclui alguns laços de repetição para calcular o preenchimento do restante dos elementos que compõe a matriz de massa elementar, assim temos:

```

119- for i=1:6
120-     ME(i, i+2)=m(2);
121-     ME(i+2, i)=m(2);
122- end
123-
124- for i=1:4
125-     ME(i, i+4)=m(3);
126-     ME(i+4, i)=m(3);
127- end
128- for i=1:2
129-     ME(i, i+6)=m(2);
130-     ME(i+6, i)=m(2);
131- end

```

Com esses laços de repetição para a montagem da matriz de massa elementar, finaliza-se as alterações feitas no código para o caso do sistema dinâmico, onde o código completo com todas alterações e descrições que não foram apresentadas encontra-se no apêndice A.

4.4 Exemplo de caso: Sistema Dinâmico

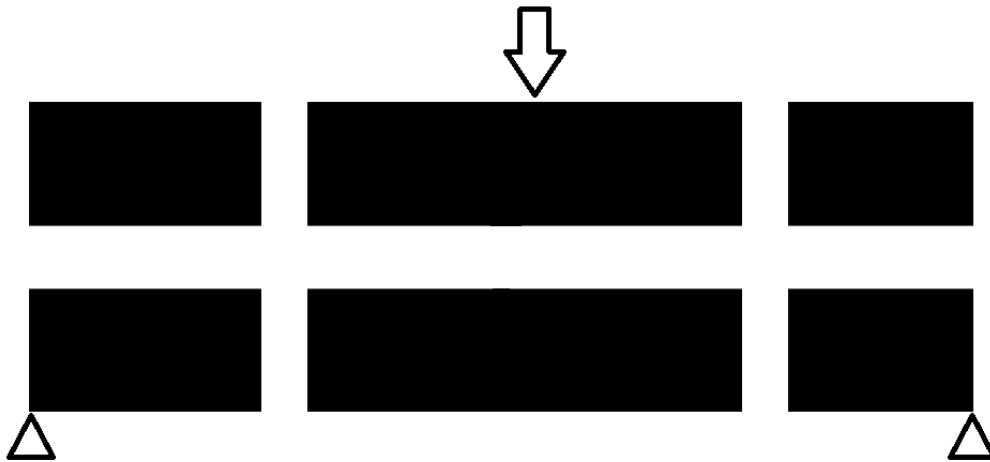
O exemplo retratado a seguir mostra a otimização topológica de uma viga bi-apoiada sob um carregamento periódico de frequência dada.

Para o exemplo 1 (figura 15), foram usados os seguintes parâmetros:

- Malha: 10x5;
- Fração de Volume: 0,8;
- Raio do filtro: 1,5 elementos;
- Frequência a ser absorvida: 4,24 rad/s;

- Número máximo de iterações: 50.

Figura 15: Exemplo 1, tempo de processamento: aprox. 2 minutos

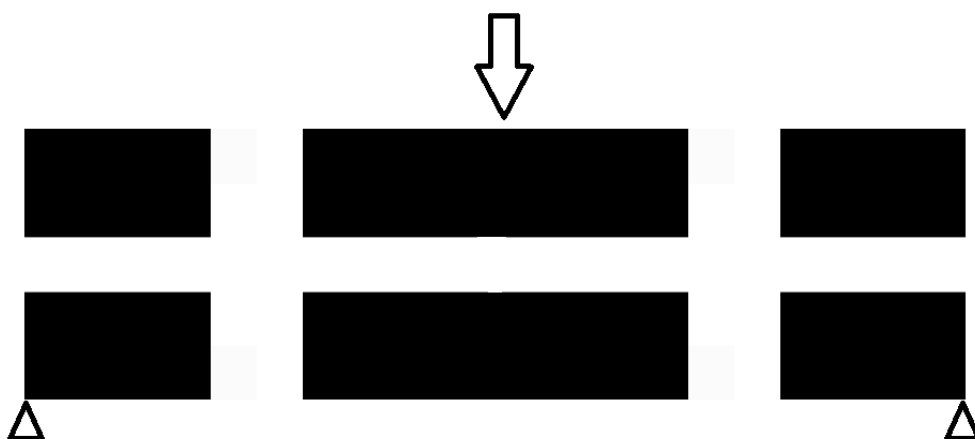


Fonte: Autor

Apesar de fisicamente não realizável, o exemplo 1 indica que as regiões de onde devem ser retirados o material, de tal forma que o menor autovalor seja mais afetado são as regiões longe do apoio e do ponto de aplicação da carga.

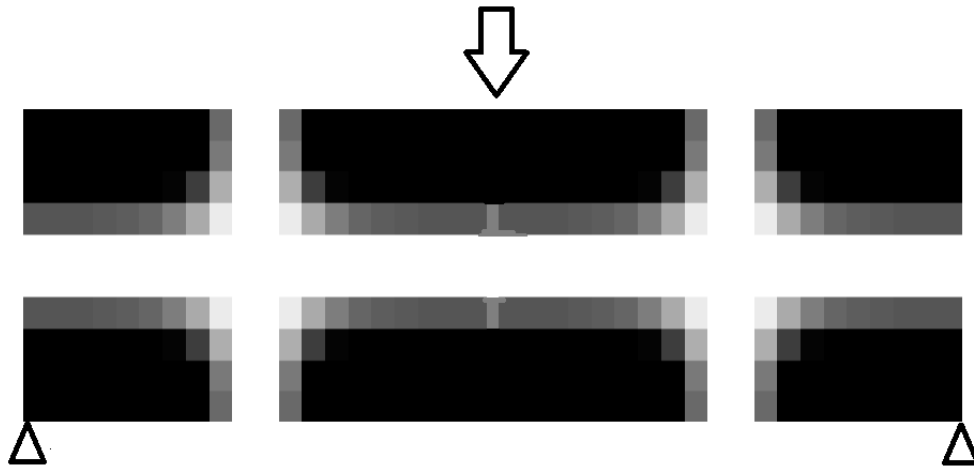
No exemplo 2, aumenta-se o número de iterações, e verifica-se que as conclusões tiradas a partir do exemplo 1 permanecem válidas.

Figura 16: Exemplo 2: 100 iterações – tempo de processamento: aprox. 4 minutos



Fonte: Autor

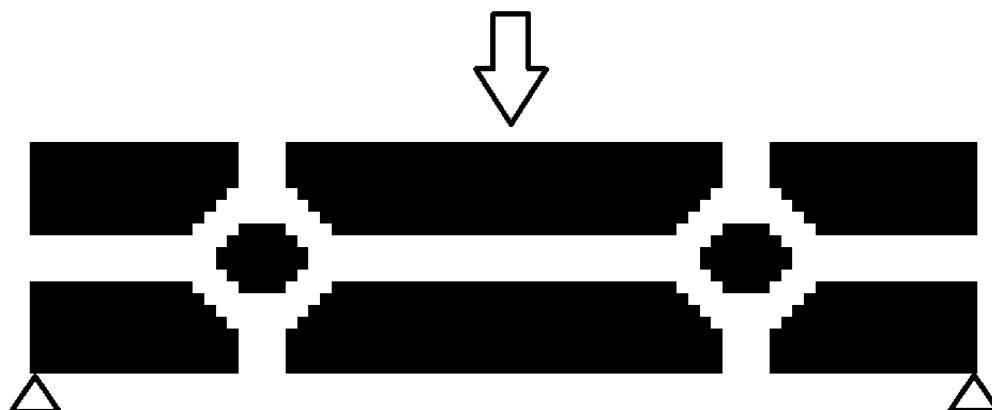
Figura 17: Exemplo 3: Malha: 20x10, fração de volume: 0,6 - 50 iterações - tempo de processamento: aprox.. 2 horas e 30 minutos



Fonte: Autor

A figura 17, mostra o exemplo 3 onde se refinou a malha, além de se retirar mais material com uma fração do volume de 0,6. Observa-se que o processo de otimização indica a retirada de material deverá ser feita no centro da semi-viga. Observa-se, também, que a penalidade não foi suficiente para evitar algumas regiões cinza.

Figura 18: Exemplo 4: Malha: 40x20, Fração de volume: 0.8 - tempo de processamento aproximadamente 70 horas



Fonte: Autor

No exemplo 4, da figura 18, refinou-se ainda mais a malha, usando uma fração de volume de 0,8. Observa-se que a região que mais afeta o menor autovalor não é exatamente o centro da semi-viga, mas uma região ao redor desse ponto.

Entretanto é importante lembrar que a geometria da otimização não é fisicamente realizável. Ao se produzir um ADV, a região central terá que ser vazia.

5. CONCLUSÕES

O principal objetivo almejado pelo presente trabalho foi o de desenvolver uma topologia para um sólido contínuo, em forma de viga que pudesse funcionar como um amortecedor dinâmico de vibrações passivo, pois a presença de vibrações mecânicas em estruturas provoca uma série de inconvenientes que podem comprometer a sua integridade física.

Além de comprometer a integridade de uma estrutura, devido à ocorrência de fratura por fadiga, a vibração mecânica normalmente é a principal fonte de ruído e desconforto acústico.

A primeira etapa almejada para a resolução do problema foi a construção de um código em MATLAB® que correlacionasse a vibração do amortecedor com a força de excitação originada da estrutura cuja vibração se deseja absorver.

Após o desenvolvimento desse código, tornou-se possível modificar as variáveis de topologia do absorvedor a fim de reduzir a vibração de uma determinada frequência a níveis aceitáveis.

Um detalhe importante da otimização de topologia, é que tanto o contorno como os furos existentes no espaço de projeto, ou seja, a distribuição de massa pode ser modificada à medida que se caminha em direção à topologia ótima.

Assim, neste trabalho, aplicou-se a técnica de Otimização Topológica por Distribuição de Massa, isto é, as variáveis de projeto foram as densidades nos diferentes pontos do absorvedor.

O principal desafio deste trabalho foi a elaboração do código, que foi baseado no Método dos Elementos Finitos, onde cada elemento finito possuía uma densidade uniforme. Essas densidades foram as variáveis de projeto.

Alguns dos problemas encontrados na implementação computacional do problema de otimização de topologia foram:

- Não convergência da função de iteração composta por uma combinação da função objetivo com as restrições, o que obrigou a se limitar o número de iterações no algoritmo de tal forma que o mesmo pode encerrar a otimização prematuramente. Isso não afetou os resultados da topologia pois as soluções que formavam o conjunto dos pontos fixos geravam topologias bastante próximas.

- A escolha adequada dos pesos da função objetivo e da penalidade necessária para a convergência do método.

A abordagem utilizada na implementação do código de otimização topológica, foi de fazer com que a menor frequência natural do absorvedor fosse igual à frequência da estrutura principal que se deseja absorver. Acrescentou-se na função objetivo uma restrição que penaliza densidades intermediárias.

Enquanto que a restrição de volume ficou embutida na função de atualização das densidades. O gradiente da função objetivo foi obtida numericamente, o que tornou a execução do código bastante lenta.

Quanto aos resultados da topologia ótima obtida, é importante observar que a mesma tem um significado qualitativo apenas, uma vez que não se levou em consideração as propriedades físicas reais de um material isotrópico, seja ele metálico ou um polímero. Outro detalhe que deve ser observado é o de que a geometria resultante não é fisicamente realizável, uma vez que possuem vazios que tornam o absorvedor descontínuo.

Apesar do problema das regiões descontínuas, os resultados das otimizações mostram que a melhor região para que o material seja removido é o mais distante possível dos apoios, e do ponto de aplicação da força originária da estrutura principal.

Essa regra geral pode servir de base para um modelo simulado em um software de Elementos Finitos, para um ajuste mais fino dos parâmetros de material e geometria.

Assim um próximo passo natural é o de simular um ADV pelo método dos elementos finitos para uma determinação mais apurada da frequência natural do mesmo.

Como proposta para trabalhos futuros tem-se o uso de funções objetivos diferentes como:

- A flexibilidade (*compliance*) para o caso dinâmico, similar ao caso estático.
- O volume, isto é, a quantidade de material. Nesse caso a prescrição do autovalor passa a ser uma restrição.

Assim, pode-se observar que o projeto de um ADV contínuo ainda é um problema em aberto, e que o mesmo pode ter várias soluções, dependendo que se deseja minimizar.

REFERÊNCIAS BIBLIOGRÁFICAS

Bendsoe, M. P., Sigmund, O., 2007, Topology Optimization: Theory, Methods and Applications, Ed. Springer Verlag.

Christensen, P. W., Klarbring, A., 2009, An Introduction to Structural Optimization, Ed. Springer Verlag.

Cunha Jr., S. S., 1999, Estudo Teórico e Numérico de Absorvedores Dinâmicos de Vibrações, Dissertação de Mestrado, Universidade Federal de Uberlândia.

Fish, J., Belytschko, T., 2007, Primeiro Curso em Elementos Finitos, Ed. Livros Técnicos e Científicos.

Inman, D. J., 2005, Engineering Vibration, Ed. Prentice Hall.

Jog, C., 2002, Topology Design Subjected to Periodic Loading, Journal of Sound and Vibration, pp. 687 – 709, Vol. 253(3).

Kim, N., Sankar, B., 2009, Introduction to Finite Element Analysis and design, Ed. Livros Técnicos e Científicos.

Meirovitch, L., 1986, Elements of Vibration Analysis, Ed. McGraw-Hill.

Nascimento, R., F., Análise dinâmica de vigas utilizando o elemento finito de Timoshenko com refinamento p-adaptativo, 2005, 212 p., Dissertação de mestrado – Universidade Federal Paulista.

Pedersen, N.L., 2000, Maximization of Eigenvalues Using Topology Optimization, Structural Optimization, Vol. 20, pp. 2 – 11.

Rade, D. A., Steffen Jr., V., 1999, Optimization of Dynamic Vibration Absorbers Over a Frequency Band, Proceedings of 17th International Modal Analysis Conference, Kissimee, FL, pp. 188 – 193.

Rietz, A., 2001, Sufficiency of a finite exponent in SIMP (power law) methods, Structural and Mutidisciplinary Optimization, pp. 159-163.

Sigmund, O., 2001, A 99 Line Topology Optimization Code Written in Matlab®, Structural Optimization, pp. 120 – 127, Vol. 21.

Viguié, R., Kerschen, J., 2009, Nonlinear Vibration Absorber Coupled to a Nonlinear Primary System: A Tuning Methodology, Journal of Sound and Vibration, Vol. 326, pp. 780 – 793.

APÊNDICE A

```

%%%Otimização Topológica - Código alterado para o caso dinâmico
%%%(Sigmund 200) %%
function topADV(nelx,nely,volfrac,penal,rmin,omega)
% Inicialização
x(1:nely,1:nelx) = volfrac;
loop = 0;
change = 1.;
% Início das iterações
xxx=zeros(200);
iii=1;
deltax = 0.2;
mult = 1e+8;

while change > 0.0001
    loop = loop + 1;
    xold = x;
% Solução pelo MEF
[U, autovalor]=FE(nelx,nely,x,penal);
disp(autovalor);
% Função objetivo e gradiente
[KE] = lk;
[ME] = mk;

c = mult*(omega^2-autovalor)^2+0.02*sum(sum(x.*(1-x)));
for ely = 1:nely
    for elx = 1:nelx
        x1 = x;
        x1(ely,elx) = x1(ely,elx)+deltax;
        [U1, autovalor1]=FE(nelx,nely,x1,penal);
        c1 = mult*(omega^2-autovalor1)^2+0.02*sum(sum(x1.*(1-x1)));
        dc(ely,elx) = -(c1-c)/deltax;
    end
end
iii=iii+1;
if iii>100
    break
end
% Suavização, filtragem do gradiente
[dc] = check(nelx,nely,rmin,x,dc);
% Atualização das densidades
[x] = OC(nelx,nely,x,volfrac,dc);

% Mostra os resultados
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 )])
% Plota a figura em escala cinza
colormap(gray); imagesc(-x); axis equal; axis tight; axis
off;pause(1e-6);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [xnew]=OC(nelx,nely,x,volfrac,dc)
l1 = 0; l2 = 100000; move = 0.2;

```

```

while (l2-l1 > 1e-4)
    lmid = 0.5*(l2+l1);
    xnew = max(0.001,max(x-move,min(1.,min(x+move,x.*atan(dc./lmid)))));
    if sum(sum(xnew)) - volfrac*nelx*nely > 0;
        l1 = lmid;
    else
        l2 = lmid;
    end
end
end

%%%%%%%%%% Filtro para independência de malha %%%%%%%%%%%
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
%dcn = dc;
end
%%%%%%%%%% A Resolução do MEF %%%%%%%%%%%
function [U, autovalor]=FE(nelx,nely,x,penal)
[KE] = lk;
[ME] = mk;
K = sparse(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1));
M = 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;
        M(edof,edof) = M(edof,edof) + x(ely,elx)^penal*ME;
    end
end
end

aaa = sort(eigs(M\K));
autovalor = aaa(1);

% Definição do carregamento e dos apoios
% F(2,1) = -1;
% fixeddofs = union([1:2:2*(nely+1)], [2*(nelx+1)*(nely+1)]);
% alldofs = [1:2*(nely+1)*(nelx+1)];
% freeddofs = setdiff(alldofs, fixeddofs);

% Solução do Sistema linear

U(freeddofs,:) = K(freeddofs,freeddofs) \ F(freeddofs,:);
U(fixeddofs,:)= 0;

```

```

%%%%%%%%%% Matriz de rigidez do elemento %%%%%%%%%%%
function [KE]=lk
E = 1.;
nu = 0.3;
k=[ 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)];

```

```

%%%%%%%%%% Matriz de massa do elemento %%%%%%%%%%%
function [ME]=mk
m = [1/9 1/18 1/36];
ME = eye(8);
ME = m(1)*ME;

for i=1:6
    ME(i,i+2)=m(2);
    ME(i+2,i)=m(2);
end

for i=1:4
    ME(i,i+4)=m(3);
    ME(i+4,i)=m(3);
end

for i=1:2
    ME(i,i+6)=m(2);
    ME(i+6,i)=m(2);
end

```