

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
DEPARTAMENTO DE ENGENHARIA MECÂNICA

OTIMIZAÇÃO TOPOLÓGICA DE ESTRUTURAS SUJEITAS À RESTRIÇÃO DE FREQUÊNCIA
NATURAL UTILIZANDO PROGRAMAS COMERCIAIS DE CAE

por

Mateus Canuto

Monografia apresentada ao Departamento de Engenharia Mecânica da Escola de Engenharia da Universidade Federal do Rio Grande do Sul, como parte dos requisitos para obtenção do diploma de Engenheiro Mecânico.

Porto Alegre, junho de 2012

OTIMIZAÇÃO TOPOLÓGICA DE ESTRUTURAS SUJEITAS À RESTRIÇÃO DE FREQUÊNCIA
NATURAL UTILIZANDO PROGRAMAS COMERCIAIS DE CAE

por

Mateus Canuto

ESTA MONOGRAFIA FOI JULGADA ADEQUADA COMO PARTE DOS
REQUISITOS PARA A OBTENÇÃO DO TÍTULO DE
ENGENHEIRO MECÂNICO
APROVADA EM SUA FORMA FINAL PELA BANCA EXAMINADORA DO
DEPARTAMENTO DE ENGENHARIA MECÂNICA

Prof. Dr. Arnaldo Ruben Gonzalez
Coordenador do Curso de Engenharia Mecânica

Área de Concentração: **Mecânica dos Sólidos**

Orientador: Prof. Dr. Jun Sérgio Ono Fonseca

Comissão de Avaliação:

Prof^a. Dr^a. Letícia Fleck Fadel Miguel

Prof. Dr. Juan Pablo Raggio Quintas

Prof. Dr. Edson Hikaro Aseka

Porto Alegre, 25 de junho de 2012.

AGRADECIMENTOS

Agradeço primeiramente aos meus pais Flávio e Suzete e meu irmão Flávio Jr., que sempre me apoiaram e incentivaram incondicionalmente durante este curso de graduação.

Ao meu orientador, professor Jun Sérgio Ono Fonseca, pela paciência e disposição em ensinar, estando sempre disponível e respondendo às dúvidas na realização desse trabalho.

Aos meus colegas e amigos pelas horas de descontração e alegria vividas durante este período.

CANUTO, M. **Otimização Topológica de Estruturas Sujeitas à Restrição de Frequência Natural Utilizando Programas Comerciais de CAE**. 2012. 25p. Monografia (Trabalho de Conclusão do Curso em Engenharia Mecânica) – Departamento de Engenharia Mecânica, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2012.

RESUMO

O presente trabalho trata da otimização topológica de uma estrutura contínua no estado plano de tensões sujeita a uma restrição de frequência natural. A frequência natural é obtida por análise via elementos finitos utilizando um software CAE comercial. O processo de otimização utiliza o método de programação linear sequencial e é realizado em um software comercial de programação matemática, que também é responsável pela interação com o software CAE. A topologia final é obtida e avaliada observando problemas como a dependência de malha e instabilidade de tabuleiro. Por fim, é feita a interpretação da topologia obtida, a remodelagem da geometria e nova análise por elementos finitos, comprovando a eficácia da programação de otimização implementada.

PALAVRAS-CHAVE: Otimização topológica, frequência natural, programação matemática.

CANUTO, M. **Topology Optimization of Structures Subject to Natural Frequency Constraint Using Commercial CAE Softwares**. 2012. 25p. Monografia (Trabalho de Conclusão do Curso em Engenharia Mecânica) – Departamento de Engenharia Mecânica, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2012.

ABSTRACT

The present work deals with the topology optimization of a continuous structure in the plane stress subjected to a natural frequency constraint. The natural frequency is obtained by finite elements analysis using commercial CAE software. The optimization process uses the sequential linear programming method and it is performed in commercial mathematical programming software, which is also responsible for the interacting with the CAE software. The final topology is obtained and then checked, looking for issues such as mesh dependency and checkerboard. Finally, it is carried out the obtained topology interpretation, the geometrical remodeling and new finite element analysis, demonstrating the effectiveness of the implemented optimization programming.

KEYWORDS: Topology optimization, natural frequency, mathematical programming.

ÍNDICE

	Pág.
1. INTRODUÇÃO	1
2. OBJETIVOS	1
3. REVISÃO BIBLIOGRÁFICA	1
4. ANÁLISE MODAL	2
5. OTIMIZAÇÃO TOPOLÓGICA.....	2
5.1. Variáveis de Projeto	3
5.2. Função Objetivo	3
5.3. Restrições	3
5.4. Programação Linear Sequencial (SLP)	4
5.5. Limites Móveis	4
5.6. Análise de Sensibilidade	4
5.7. Filtragem	6
6. ELEMENTOS FINITOS	7
7. SOFTWARES UTILIZADOS.....	8
8. ALGORITMO IMPLEMENTADO	9
9. PROBLEMA ANALISADO E RESULTADOS OBTIDOS	10
9.1. Descrição do Problema	10
9.2. Resultados da Otimização.....	10
9.3. Geometria Interpretada	13
10. CONCLUSÕES	14
REFERÊNCIAS.....	15
APÊNDICE A - Segundo problema analisado	16
APÊNDICE B - Programação realizada no software Matlab	17
APÊNDICE C - Macros para extração de dados do software Ansys.....	19

ÍNDICE DE FIGURAS

Figura 5.1 – Instabilidade de tabuleiro.....	6
Figura 8.1 – Fluxograma do algoritmo implementado.....	9
Figura 9.1 – Viga com massa concentrada.....	10
Figura 9.2 – Topologia obtida para uma malha de 260 elementos.....	10
Figura 9.3 – Convergência da função objetivo para malha de 260 elementos.....	11
Figura 9.4 – Variação da restrição para malha de 260 elementos.....	11
Figura 9.5 – Topologia obtida para malha de 4000 elementos.....	12
Figura 9.6 – Convergência da função objetivo para malha de 4000 elementos.....	12
Figura 9.7 – Variação da restrição para malha de 4000 elementos.....	13
Figura 9.8 – Geometria interpretada.....	14
Figura A.1 – Segunda estrutura analisada.....	16
Figura A.2 – Topologia obtida no segundo problema.....	16
Figura A.3 – Convergência da função objetivo para o segundo problema.....	16
Figura A.4 – Variação da restrição para o segundo problema.....	17
Figura A.5 – Geometria interpretada para o segundo problema.....	17

1. INTRODUÇÃO

O mercado busca constantemente por produtos com um custo cada vez menor, visando um repasse com menor valor em vista da concorrência ou mesmo obtenção de lucros maiores. Na maioria dos casos, este custo está diretamente relacionado com o material utilizado e a quantidade necessária do mesmo. Assim, procura-se a melhor configuração para o produto, de modo que se necessite a menor quantidade de material possível.

Em contrapartida, o produto deve atender ainda à sua funcionalidade, mantendo os requisitos necessários. A partir desse pressuposto que surgiu o conceito de otimização estrutural, com os primeiros trabalhos datados de 1904, por Michell, que se baseou em estudos de Maxwell, 1869 [apud Mosmann, 2003]. Os problemas de otimização estrutural podem ser classificados em três tipos: de forma, paramétricos e topológicos. As otimizações de forma e a paramétrica consistem em encontrar a melhor estrutura possível que satisfaça certa condição com a mudança de forma e parâmetros (comprimento e área, por exemplo) respectivamente. Já a topológica consiste na adição de furos e/ou reforços onde se faz necessário.

Hoje em dia, com o grande desenvolvimento da tecnologia computacional, métodos de otimização baseados na discretização estrutural por elementos finitos têm se tornado bastante frequentes. Além disso, com a evolução dos softwares comerciais na solução de problemas discretizados vem o interesse em utilizá-los nesses métodos.

Nesse trabalho é abordada a otimização topológica aplicada a uma estrutura contínua bidimensional, onde a estrutura resultante não deve ter um valor inferior de frequência natural a um valor mínimo estipulado. A frequência natural foi adotada como uma restrição de projeto pelo fato de ser um fator muito importante na aplicação de certos componentes mecânicos, onde a frequência de excitação pode causar vibrações ressonantes e comprometer o funcionamento do componente, tornando-se assim um aspecto de alta relevância em vários projetos mecânicos.

2. OBJETIVOS

Neste trabalho se propõe avaliar técnicas desenvolvidas para a otimização topológica de estruturas contínuas. Mais precisamente, busca-se desenvolver uma programação computacional, baseada em métodos já conhecidos de otimização topológica, capaz de minimizar a massa de uma estrutura respeitando um valor mínimo para a frequência natural da mesma, e observar a capacidade de implementação do método em um software comercial de elementos finitos, visando um processo viável no meio industrial.

3. REVISÃO BIBLIOGRÁFICA

Assim como a natureza busca constantemente por soluções que utilizem a menor energia possível, em problemas de otimização ocorre a busca constante pelo ótimo. Matematicamente falando, a otimização consiste na extremização de um funcional, onde o ótimo é um ponto de máximo ou mínimo.

Em otimização estrutural, a otimização topológica é a que mais se aproxima da solução ótima, pois dado um domínio, esta busca a melhor distribuição de material possível. Porém, para que seja possível o uso deste tipo de otimização é necessário que o domínio seja discretizado em elementos finitos, e com isso a necessidade de uma parametrização para o meio contínuo. O método da homogeneização utilizado nesse trabalho realiza uma parametrização com base no tensor constitutivo do material, este método é discutido por Bendsøe, 1995.

Problemas como a instabilidade de tabuleiro, a dependência de malha e a não unicidade de soluções são encontrados em muitos casos de otimização topológica, estes problemas são claramente abordados por Sigmund e Petersson, 1998, que ainda sugerem soluções para contorná-los.

A programação linear é um dos métodos mais utilizados na solução de problemas de otimização, por se tratar de um método simples e capaz de encontrar a solução ótima mesmo para os casos mais complexos. A sua desvantagem está relacionada com a necessidade de um número maior de iterações que outros métodos. Técnicas de programação linear são amplamente discutidas no texto de Arora, 1989.

Casos envolvendo restrição de frequência natural em otimização são muito comuns na indústria, pois podem estar presentes em qualquer mecanismo que realize um movimento ou que tente impedir este. Na literatura são encontrados textos abordando este assunto, como Haftka e Gurdal, 1992, que tratam da sensibilidade da frequência em processos de otimização e trabalhos desenvolvidos com esta restrição em Mosmann, 2003.

4. ANÁLISE MODAL

Em vibrações livres, sem amortecimento, a equação do movimento é dada por:

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{K}\mathbf{u} = 0 \quad (4.1)$$

onde \mathbf{M} e \mathbf{K} são as matrizes de massa e rigidez respectivamente, e $\ddot{\mathbf{u}}$ e \mathbf{u} correspondem aos vetores de aceleração e deslocamento.

A partir da equação (4.1) se aplica a suposição de resposta com comportamento harmônico, de onde se obtém o deslocamento e a aceleração, representados pelas seguintes equações:

$$\mathbf{u} = \boldsymbol{\phi} \text{sen}(\omega t) \quad (4.2)$$

$$\ddot{\mathbf{u}} = -\omega^2 \boldsymbol{\phi} \text{sen}(\omega t) \quad (4.3)$$

onde $\boldsymbol{\phi}$ é a amplitude modal, ω a frequência e t o tempo. Substituindo as equações (4.2) e (4.3) na equação (4.1) e fazendo algumas operações algébricas, chega-se a:

$$(\mathbf{K} - \omega^2 \mathbf{M})\{\boldsymbol{\phi}\} = 0 \quad (4.4)$$

Assim, é estabelecido o problema de autovalores, onde duas soluções são possíveis: a trivial, onde $\{\boldsymbol{\phi}\} = 0$ e a solução desejada, onde o primeiro termo deve ser igual a zero:

$$\det[(\mathbf{K} - \omega^2 \mathbf{M})] = 0 \quad (4.5)$$

Fazendo-se $\lambda = \omega^2$, obtêm-se a forma mais usual do problema de autovalores:

$$\det[(\mathbf{K} - \lambda \mathbf{M})] = 0 \quad (4.6)$$

Para a obtenção dos autovalores do problema anterior, os softwares comerciais de elementos finitos oferecem vários métodos de solução. Nesse trabalho foi escolhido o método de Lanczos, que se baseia na normalização dos autovetores pela matriz de massa, pois apresenta uma solução mais rápida para a extração de poucos modos e frequências em relação a outros métodos [Ansys, 2007].

5. OTIMIZAÇÃO TOPOLÓGICA

Na otimização topológica busca-se minimizar ou maximizar um funcional que respeite as restrições estabelecidas, distribuindo o material da melhor maneira possível dentro de um domínio admissível de projeto. Sua definição, assim como os outros tipos de otimização, de uma forma geral é dada por:

$$\begin{aligned}
& \text{Minimizar} && f(x) \\
& \text{Sujeito a} && g_j(x) \leq 0, j = 1, \dots, n_g \\
& && e \quad h_k(x) = 0, k = 1, \dots, n_h
\end{aligned} \tag{5.1}$$

onde x é a variável de projeto, $f(x)$ é o funcional objetivo, $g_j(x)$ e $h_k(x)$ são as restrições de desigualdade e igualdade.

5.1. Variáveis de Projeto

As variáveis de projeto são os parâmetros que podem ser alterados na busca pelo ótimo. É comum que se utilize em otimização topológica a parametrização do tensor constitutivo do material, onde a variável de projeto é uma pseudo-densidade ρ_i que pode assumir valores entre 0 e 1, determinando a existência ou não de material em um certo domínio. Em outras palavras, para um problema discretizado em elementos finitos e considerando-se o material linear isotrópico, têm-se a seguinte forma:

$$\begin{aligned}
E_i &= \rho_i E^0 \\
0 &\leq \rho_i \leq 1
\end{aligned} \tag{5.2}$$

onde ρ_i é a densidade do elemento i , E_i é o módulo de elasticidade do elemento i e E^0 é o módulo de elasticidade do material base.

Contudo, por se tratar de um problema contínuo, a solução passa a depender da discretização do domínio (dependência de malha) [Mosmann, 2003], o que significa dizer que para diferentes malhas de elementos finitos, diferentes soluções são possíveis.

5.2. Função Objetivo

No processo de otimização, a função objetivo é a função que se deseja extremizar, é também conhecida como função custo. Nesse trabalho tem-se como objetivo a minimização da massa de uma estrutura e, como se trata de um material incompressível, o volume está diretamente relacionado com a massa, sendo assim a função objetivo adotada assume a seguinte forma:

$$V = \sum_{i=1}^{Nelem} \rho_i^n v_i \tag{5.3}$$

onde V representa o volume total, v_i é o volume do elemento i e n é um expoente de penalização sobre as densidades intermediárias. A penalização tem o intuito de aproximar a solução para uma configuração 0-1 no valor das densidades, porém a função sem a penalização que é convexa passa ser não-convexa, induzindo o problema a apresentar mínimos locais.

5.3. Restrições

Como descrito anteriormente, a restrição adotada no problema analisado é a frequência natural, sendo a função de desigualdade descrita por:

$$\omega_n \geq \omega_{nlim} \tag{5.4}$$

onde ω_n é a frequência natural do primeiro modo da estrutura analisada e ω_{nlim} é o valor mínimo que ela pode atingir.

Problemas com restrição de frequência natural são muito comuns na indústria, onde em várias situações algum componente mecânico é submetido a vibrações. Estas vibrações excitam o componente e, se possuírem uma frequência próxima da natural, pode ocorrer o efeito de ressonância, onde a amplitude da vibração aumenta gradativamente até o colapso do mecanismo. Assim, a possibilidade de se otimizar uma estrutura mantendo a frequência de ressonância distante da frequência de excitação se torna bastante vantajosa.

5.4. Programação Linear Sequencial (SLP)

Vários métodos podem ser utilizados na solução do problema de otimização, porém quando se tem muitas variáveis de projeto os métodos numéricos e a programação matemática se tornam mais vantajosos [Arora, 2004]. A programação linear consiste na linearização das funções objetivo e restrições, caso elas não forem lineares, por expansão em séries de Taylor com truncamento no primeiro termo, necessitando apenas das derivadas de primeira ordem. Assim, as funções dadas pelas equações (5.3) e (5.4), que não são lineares, tornam-se:

$$V(\rho^{k+1}) = V(\rho^k) + \sum_{i=1}^{Nelem} (\rho_i^{k+1} - \rho_i^k) \left(\frac{dV}{d\rho_i} \right) \Big|_{\rho_i^k} \quad (5.5)$$

$$\omega_n(\rho^{k+1}) = \omega_n(\rho^k) + \sum_{i=1}^{Nelem} (\rho_i^{k+1} - \rho_i^k) \left(\frac{d\omega_n}{d\rho_i} \right) \Big|_{\rho_i^k} \quad (5.6)$$

onde o índice k representa a atual iteração de otimização. Como a linearização por série de Taylor é válida somente em uma região muito pequena em torno de um ponto, no caso ρ_i^k , o problema passa a ser resolvido sequencialmente em intervalos definidos pelos limites móveis.

5.5. Limites Móveis

Os limites móveis delimitam o intervalo onde a linearização é válida. Se o valor do intervalo for muito grande, o erro será grande e pode não ocorrer a convergência, e se for muito pequeno, acarreta em um número demasiado de iterações e uma demora maior na solução do problema. Em problemas de otimização é comum que os limites móveis sejam um valor fixo em função de uma porcentagem do valor atual das variáveis, e assim se determinam os limites superior e inferior da variável:

$$\rho_i^U = \rho_i + L_i \quad (5.7)$$

$$\rho_i^L = \rho_i - L_i \quad (5.8)$$

onde L_i é o limite móvel (geralmente entre 5% e 10% do valor da variável), ρ_i^U e ρ_i^L são o limite superior e inferior respectivamente.

Porém, para garantir a convergência da análise é implementado neste trabalho um ajuste heurístico dos limites móveis, baseado na variação da variável nas três últimas iterações. Este ajuste é parecido com o utilizado por Mosmann, 2003, onde o limite móvel é acrescido de 5% caso o sinal da variação do valor da variável não mude dentro de três iterações, e caso contrário é decrescido do mesmo percentual.

5.6. Análise de Sensibilidade

Como visto anteriormente, a linearização das funções objetivo e restrição depende da derivada das mesmas em relação às variáveis de projeto. Estas derivadas correspondem à sensibilidade das funções a uma variação das variáveis.

Como o volume depende diretamente da variável de projeto, a sua derivada é simples e corresponde à seguinte equação:

$$\frac{dV}{d\rho_i} = \frac{1}{n} \rho_i^{\left(\frac{1}{n}-1\right)} v_i \quad (5.9)$$

Já a derivada da frequência natural não é tão simples, pois como visto na equação (4.4) se traduz na derivada dos autovalores, que não são diretamente dependentes das variáveis de projeto. Sua derivada é apresentada de forma mais detalhada e exata por Haftka e Gürdal, 1992, aqui é sugerida da forma como segue. Assim, multiplicando-se a equação (4.4) por ϕ^T se obtém:

$$\phi^T (K - \lambda M) \phi = 0 \quad (5.10)$$

Derivando-se a equação anterior em relação à variável de projeto e sabendo que a derivada de ϕ e ϕ^T é muito pequena, chega-se à seguinte expressão:

$$\phi^T \left(\frac{dK}{d\rho_i} - \frac{d\lambda}{d\rho_i} M - \lambda \frac{dM}{d\rho_i} \right) \phi = 0 \quad (5.11)$$

Isolando $\frac{d\lambda}{d\rho_i}$ é possível chegar a:

$$\frac{d\lambda}{d\rho_i} = \frac{\phi^T \left(\frac{dK}{d\rho_i} - \lambda \frac{dM}{d\rho_i} \right) \phi}{\phi^T M \phi} \quad (5.12)$$

Agora, para que se possa simplificar a equação (5.12) é necessário que o seu denominador seja normalizado em relação à massa:

$$\phi^T M \phi = 1 \quad (5.13)$$

$$\frac{d\lambda}{d\rho_i} = \phi^T \left(\frac{dK}{d\rho_i} - \lambda \frac{dM}{d\rho_i} \right) \phi \quad (5.14)$$

Tal normalização exige que a solução de autovalores respeite a mesma operação adotada, este é outro motivo para que a escolha da ferramenta de solução utilize o método de Lanczos.

Ainda é utilizada a hipótese de que a derivada da matriz de massa possui valores muito pequenos se comparados com a derivada da matriz de rigidez, o que simplifica ainda mais a equação (5.14):

$$\frac{d\lambda}{d\rho_i} = \phi^T \frac{dK}{d\rho_i} \phi \quad (5.15)$$

Assim, a derivada dos autovalores depende da derivada da matriz de rigidez em relação a variável de projeto. No caso de otimização analisado, existe uma variável associada a cada elemento finito e multiplicando a sua matriz de rigidez. Esta dedução pode ser feita a partir equação (5.2), observando-se que a matriz constitutiva do material torna-se diretamente proporcional à variável e com isso a derivada do autovalor se resume à:

$$\frac{d\lambda}{d\rho_i} = \phi_i^T K_{ei} \phi_i \quad (5.16)$$

onde K_{ei} é a matriz de rigidez base do elemento i e ϕ_i é o autovetor correspondente.

Em otimizações envolvendo autovalores, um problema que surge está relacionado com a possibilidade de se encontrar autovalores repetidos durante o processo de otimização, ou seja, autovetores que estejam associados a um mesmo autovalor. Este problema causa instabilidades no método de otimização e dificulta a convergência. Existem maneiras para se contornar esse tipo de problema quando se deseja controlar a frequência de um modo específico, contudo no caso analisado a frequência mínima é a de interesse e não a relacionada a um modo específico.

5.7. Filtragem

A filtragem é uma técnica utilizada para minimizar os efeitos causados por problemas conhecidos em otimização topológica como a instabilidade de tabuleiro, a dependência de malha e a existência de mínimos locais.

A dependência de malha tende a resultar em soluções mais complexas para um refino maior de malha, tornando difícil a manufatura de um componente com um número elevado de detalhes e encarecendo o produto final. Já a não unicidade de soluções fornece resultados diferentes para uma mesma discretização do problema com uma simples mudança de parâmetros utilizados na otimização.

A instabilidade de tabuleiro consiste na formação de uma solução com densidades alternadas em uma região do domínio, configurando um tabuleiro de xadrez, como pode ser visto na Figura 5.1.

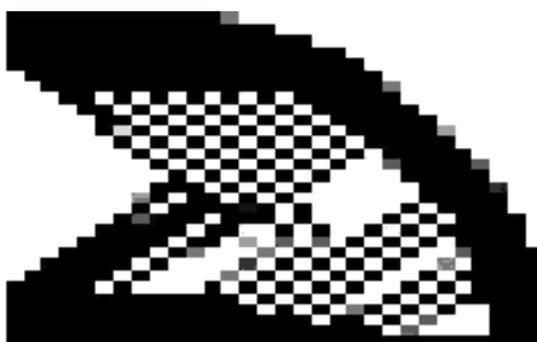


Figura 5.1 – Instabilidade de tabuleiro.

Este problema é o mais comum e ocorre principalmente devido a utilização de penalização das densidades intermediárias.

Neste trabalho é utilizada uma técnica de filtragem proposta por Cardoso, 2000, que se aplica na alteração dos limites móveis utilizando uma média ponderada entre os elementos situados em uma região determinada por um raio definido pelo usuário. Assim, o filtro tende a formar regiões mais uniformes e diminuindo a incidência dos problemas citados anteriormente. As expressões que representam o filtro são:

$$W_i = \frac{R_{max} - R_{ij}}{R_{max}} \quad (5.17)$$

$$\bar{W}_i = \frac{\sum_{j=1}^{nv} W_j}{nv} \quad (5.18)$$

$$\hat{a}_i = \frac{a_i v_i + \bar{W}_i \sum_{j=1}^{nv} a_j v_j}{v_i + \bar{W}_i \sum_{j=1}^{nv} v_j} \quad (5.19)$$

onde W_i é o peso de um elemento, R_{max} é o raio máximo de atuação do filtro, R_{ij} é a distância entre os centróides dos elementos i e j , nv é o número de vizinhos, \overline{W}_i é o peso médio dos elementos vizinhos, v é o volume do elemento e a é a densidade do elemento.

Contudo, a ação do filtro utilizado tende a ser atenuada para valores elevados dos limites móveis [Mosmann, 2003].

6. ELEMENTOS FINITOS

Como descrito anteriormente, o método de otimização topológica adotado nesse trabalho necessita a discretização do domínio em elementos finitos. Aqui, é apresentada a formulação utilizada para a obtenção da matriz de rigidez do elemento finito, necessária para o cálculo da derivada dos autovalores.

O elemento finito utilizado possui quatro nós e formulação bilinear isoparamétrica. As suas funções de interpolação são definidas por:

$$\begin{aligned} N_1 &= \frac{1}{4}(1-s)(1-t) \\ N_2 &= \frac{1}{4}(1+s)(1-t) \\ N_3 &= \frac{1}{4}(1+s)(1+t) \\ N_4 &= \frac{1}{4}(1-s)(1+t) \end{aligned} \quad (6.1)$$

Com as funções de interpolação são definidos os deslocamentos nos nós pela seguinte equação:

$$\mathbf{U}_e = \mathbf{X}_e^T \mathbf{N} \quad (6.2)$$

onde \mathbf{U}_e é o vetor de deslocamentos dos nós do elemento, \mathbf{X}_e é o vetor das posições nodais e \mathbf{N} é o vetor contendo as funções de interpolação.

A matriz jacobiana \mathbf{J} de transformação para o caso bidimensional é definida por:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial u_1}{\partial s} & \frac{\partial u_2}{\partial s} \\ \frac{\partial u_1}{\partial t} & \frac{\partial u_2}{\partial t} \end{bmatrix} \quad (6.3)$$

A matriz de transformação deformação-deslocamento \mathbf{B} é dada por:

$$\mathbf{B} = \begin{bmatrix} \frac{\partial N_1}{\partial u_1} & 0 & \frac{\partial N_2}{\partial u_1} & 0 & \frac{\partial N_3}{\partial u_1} & 0 & \frac{\partial N_4}{\partial u_1} & 0 \\ 0 & \frac{\partial N_1}{\partial u_2} & 0 & \frac{\partial N_2}{\partial u_2} & 0 & \frac{\partial N_3}{\partial u_2} & 0 & \frac{\partial N_4}{\partial u_2} \\ \frac{\partial N_1}{\partial u_2} & \frac{\partial N_1}{\partial u_1} & \frac{\partial N_2}{\partial u_2} & \frac{\partial N_2}{\partial u_1} & \frac{\partial N_3}{\partial u_2} & \frac{\partial N_3}{\partial u_1} & \frac{\partial N_4}{\partial u_2} & \frac{\partial N_4}{\partial u_1} \end{bmatrix} \quad (6.4)$$

Para o estado plano de tensões, considerando material linear isotrópico, a matriz constitutiva \mathbf{C} é definida por:

$$C = \begin{bmatrix} \frac{E}{1-\nu^2} & \frac{E\nu}{1-\nu^2} & 0 \\ \frac{E\nu}{1-\nu^2} & \frac{E}{1-\nu^2} & 0 \\ 0 & 0 & \frac{E}{2(1+\nu)} \end{bmatrix} \quad (6.5)$$

onde E é o módulo de elasticidade do material e ν é o coeficiente de Poisson.

Assim, é possível se obter a matriz de rigidez do elemento finito resolvendo-se a seguinte equação:

$$K_e = \int_{-1}^1 \int_{-1}^1 \mathbf{B}^T \mathbf{C} \mathbf{B} J ds dt \quad (6.6)$$

onde J é o determinante da matriz jacobiana.

7. SOFTWARES UTILIZADOS

O presente trabalho empregou dois softwares principais, sendo eles:

- ANSYS 12.1 - Software comercial de análises numéricas via elementos finitos;
- MATLAB R2010 - Software comercial de processamento matemático.

Alguns softwares de análises por elementos finitos disponíveis hoje em dia no mercado, assim como o Ansys, já possuem ferramentas de otimização. Porém, estas ferramentas estão restringidas para alguns poucos casos de otimização. Para a realização desse trabalho foi feito um estudo da possibilidade de se implementar toda a rotina de otimização dentro do ambiente de programação do software Ansys, visando aproveitar o seu bom desempenho na solução das análises sem perder tempo com transferência de dados para outros softwares. No entanto, se verificou que a linguagem de programação utilizada pelo Ansys é de difícil utilização se comparada com outras linguagens conhecidas de programação matemática. Além de não possuir uma ferramenta para solução de programação linear, o que torna inviável a sua utilização no cálculo de otimização. Assim, o software Ansys ficou responsável apenas pela análise modal do problema proposto.

O software Matlab foi então utilizado para a programação do método de otimização. Ele também ficou responsável pela comunicação com o Ansys, solicitando a sua inicialização a cada iteração. Nele, foi utilizado a ferramenta "linprog" usando o método Simplex para a solução do problema linear de otimização. O Simplex é um método muito conhecido e eficiente na otimização de problemas lineares, seu pressuposto é de que a solução ótima se encontra em um dos "cantos" da polítope formada pelas restrições do problema e assim percorre estes extremos passando de uma solução básica viável para outra reduzindo o valor da função objetivo. Uma discussão completa deste método pode ser encontrada em várias bibliografias como em Arora, 1989.

A análise modal poderia ainda ser programada e realizada também com o software Matlab, evitando assim o tempo gasto entre a troca de informação entre dois softwares. Porém, a programação necessária para realização da análise modal, utilizando o mesmo método de solução adotado com o Ansys, seria um tanto complexa. Além, do método de solução utilizado pelo Ansys compensar o tempo perdido entre a troca de dados dos dois softwares, por fornecer uma resposta mais rápida se comparado com um método de solução mais viável em programação.

8. ALGORITMO IMPLEMENTADO

Na implementação de análise empregada o processamento do MATLAB solicita a inicialização do software Ansys para primeiramente gerar o modelo de análise, fazer a sua discretização e após exportar dados necessários para o cálculo da matriz de rigidez de cada elemento finito. Após, é iniciado o processo de otimização, onde é solicitada a inicialização do Ansys a cada iteração e são exportados os autovalores e autovetores para o novo cálculo de densidades até a sua convergência. Na Figura 8.1 é apresentado um fluxograma simplificado do algoritmo implementado.

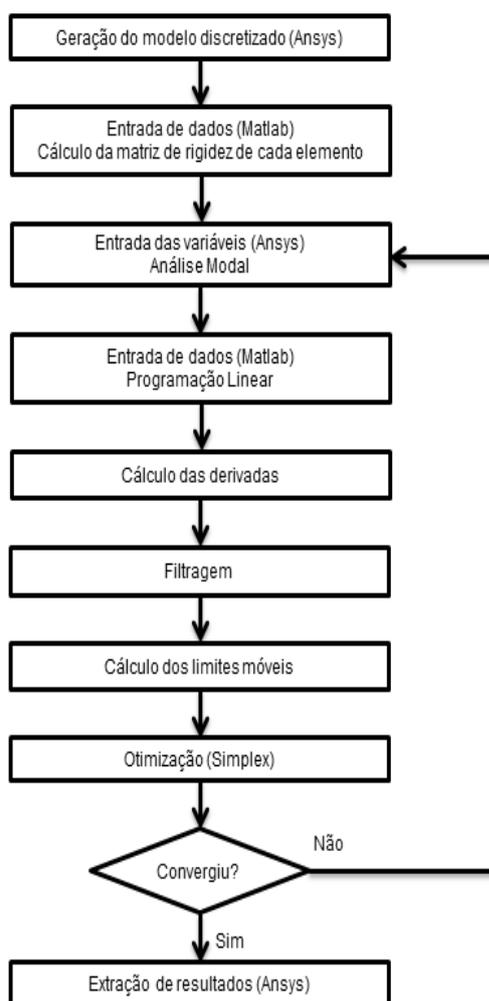


Figura 8.1 – Fluxograma do algoritmo implementado.

Uma das vantagens do algoritmo desenvolvido é que a utilização do software CAE na análise modal permite a aplicação em problemas onde a geometria em análise é submetida a condições de contorno complexas, sem a necessidade de modificações na estrutura básica da programação matemática implementada.

Nos apêndices B e C encontram-se a programação desenvolvida em Matlab e as macros necessárias para a extração de dados do software Ansys.

9. PROBLEMA ANALISADO E RESULTADOS OBTIDOS

9.1. Descrição do Problema

O problema analisado para verificação da programação implementada consiste em uma viga com dimensões de 8x5x1 metros engastada na face esquerda e com massa concentrada de 100.000 kg no nó do canto inferior direito. A Figura 9.1 ilustra o problema analisado.



Figura 9.1 – Viga com massa concentrada.

Para a análise de elementos finitos foi considerado estado plano de tensões utilizando-se o elemento de quatro nós com formulação bilinear isoparamétrica “PLANE182” e o elemento “MASS21” para a aplicação de massa concentrada, ambos disponíveis no software Ansys. Foi utilizado material linear isotrópico com módulo de elasticidade base igual a $2,1 \times 10^{11} \text{ N/m}^2$, coeficiente de Poisson igual a 0,3 e massa específica igual a 7850 kg/m^3 .

No apêndice A é apresentada ainda a solução para outro problema, também no estado plano de tensões.

9.2. Resultados da Otimização

Primeiramente foi realizada uma otimização com uma malha formada por 260 elementos utilizando o expoente de penalização de densidades igual a 1/5, densidades variando entre 10^{-3} e 1 (visto que densidades iguais a zero causam instabilidade numérica), limites móveis iniciais iguais a 0,05 e tolerância para a convergência de 10^{-3} . Para o processo de filtragem foi utilizado um raio de 0,5 m que abrange 4 elementos e um ajuste de peso de 0,3. A estrutura foi submetida à minimização de volume com restrição de frequência natural mínima de 27 Hz e a topologia final obtida pode ser visualizada na Figura 9.2.

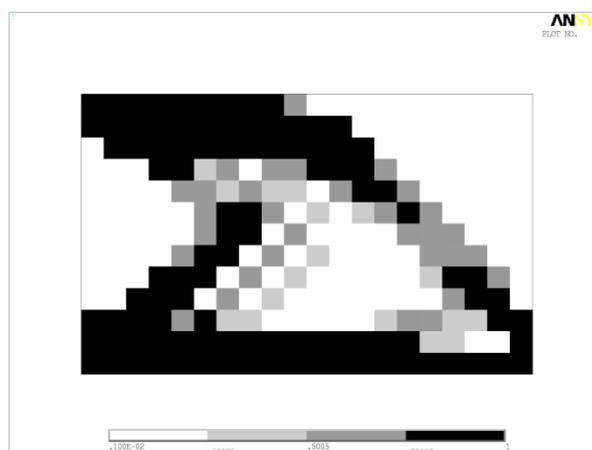


Figura 9.2 – Topologia obtida para uma malha de 260 elementos.

Nota-se na Figura 9.2 que mesmo com a utilização do filtro ocorre a instabilidade de tabuleiro em algumas regiões e a presença de elementos com densidades intermediárias. A convergência da função objetivo e a variação da restrição são apresentadas na Figura 9.3 e na Figura 9.4 respectivamente.

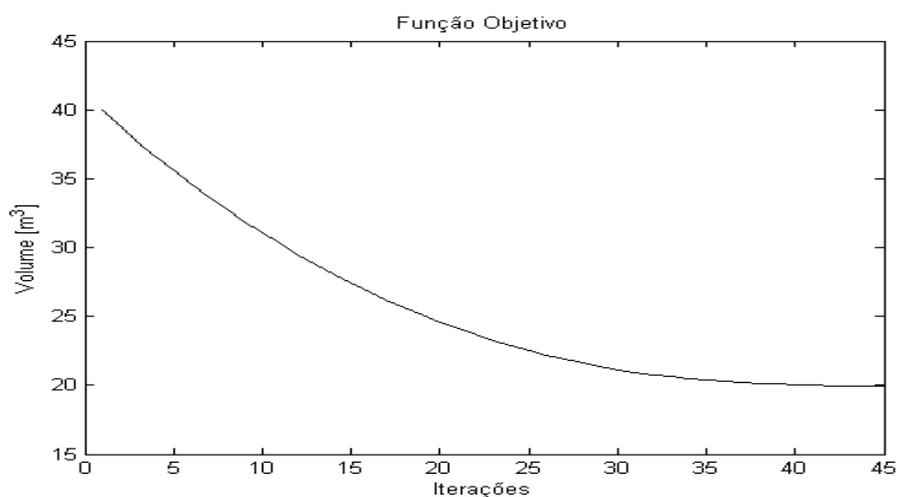


Figura 9.3 – Convergência da função objetivo para malha de 260 elementos.

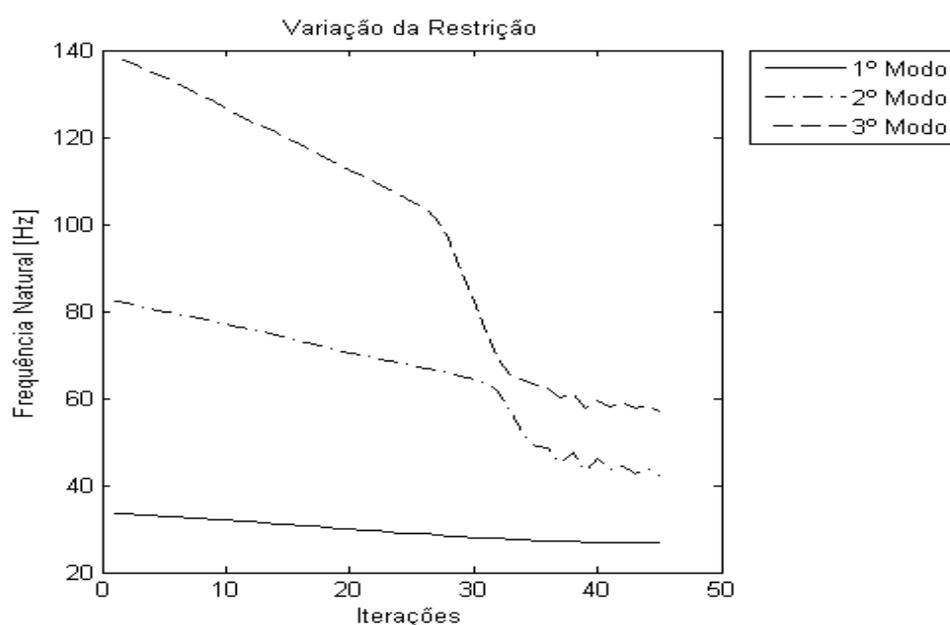


Figura 9.4 – Variação da restrição para malha de 260 elementos.

Na Figura 9.3 é possível observar que o volume da estrutura foi reduzido em 50% com apenas 45 iterações necessárias. Já na Figura 9.4, se percebe que a frequência do 1º modo se estabiliza enquanto as frequências dos modos superiores variam livremente e tendem a se aproximar da frequência mínima.

Uma das possíveis explicações para ter ocorrido a instabilidade de tabuleiro e a presença de densidades intermediárias seria devido à utilização de uma malha grosseira, que prejudica a ação do filtro e dificulta a penalização da função objetivo. Logo, visando resolver estes problemas, foi realizada uma nova otimização com um refinamento maior de malha, utilizando-se 4000 elementos e um raio de filtragem de 0,11 (também abrangendo 4

elementos). A topologia obtida apresentou-se mais definida e pode ser visualizada na Figura 9.5.

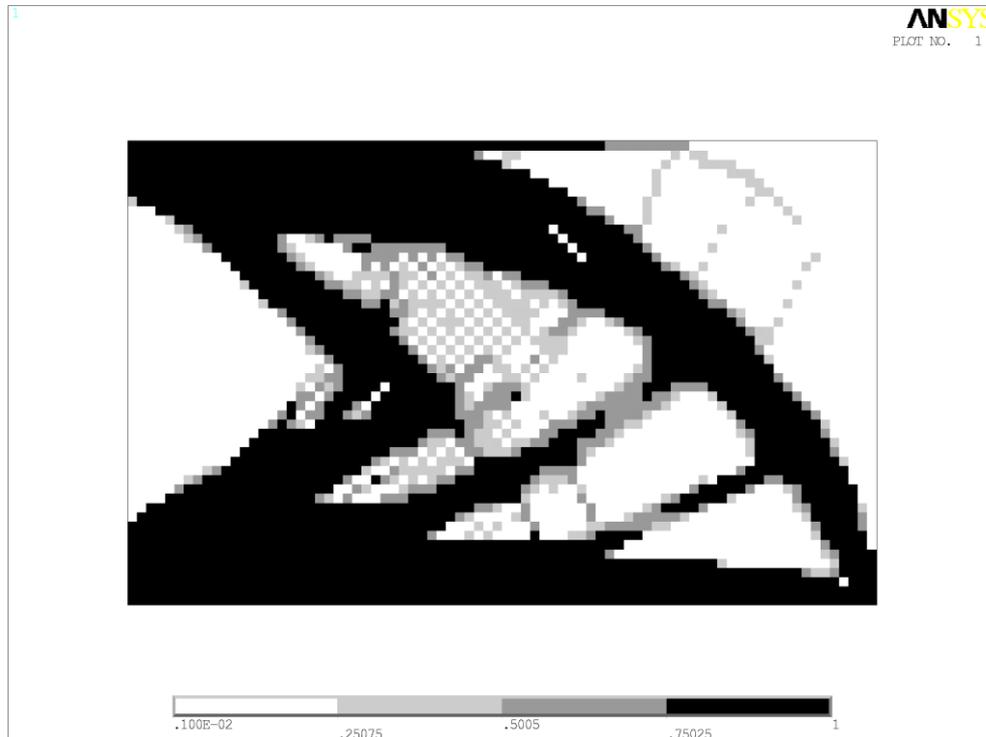


Figura 9.5 – Topologia obtida para malha de 4000 elementos.

Porém, ainda é possível observar a instabilidade de tabuleiro em certas regiões e a presença de alguns elementos com densidades intermediárias, o que demonstra certa ineficiência no método de filtragem utilizado para o caso em análise. A convergência da função objetivo e a variação da restrição para este caso são apresentadas nas Figura 9.6 e Figura 9.7, respectivamente.

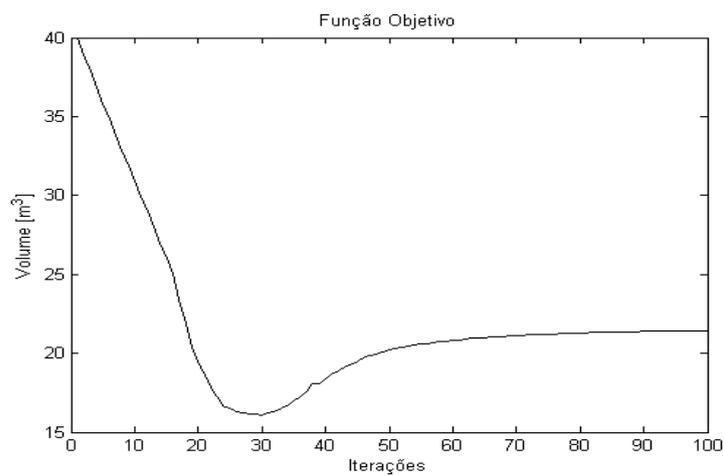


Figura 9.6 – Convergência da função objetivo para malha de 4000 elementos.

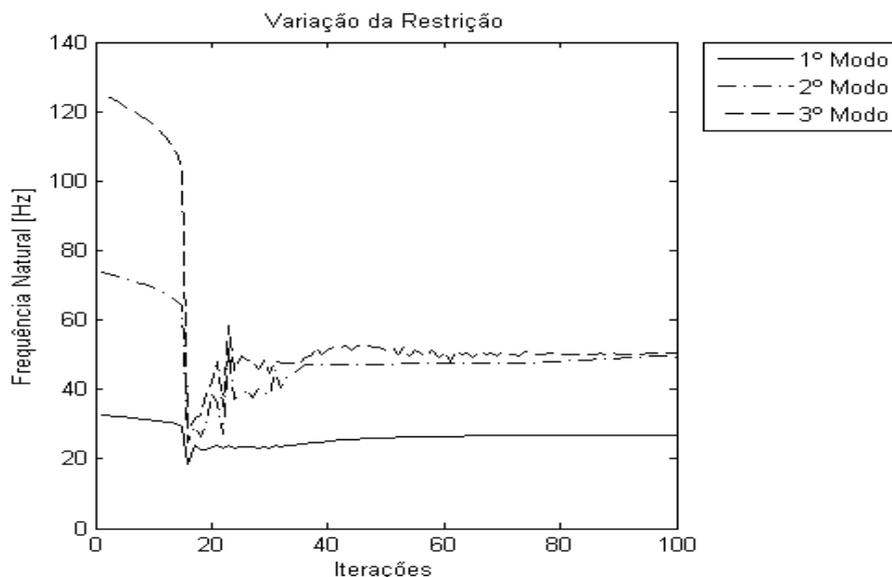


Figura 9.7 – Variação da restrição para malha de 4000 elementos.

Como pode ser visto na Figura 9.6, o volume tem uma drástica redução nas primeiras iterações e após sofre um leve aumento até se estabilizar, terminando em uma redução em torno de 45% com 100 iterações.

Na Figura 9.7 se percebe que no momento em que ocorre uma grande redução no volume, as frequências dos três primeiros modos tendem a se igualar, o que provocaria uma instabilidade e dificultaria a convergência. Porém isto não chega a acontecer neste caso, pois a atuação do filtro começa a definir melhor a topologia, reforçando a estrutura e separando novamente os dois primeiros modos, enquanto o terceiro termina tendo a mesma frequência do segundo modo sem influenciar na restrição.

O processo de otimização dos dois casos foi realizado utilizando-se a mesma capacidade computacional. Enquanto no primeiro caso foram necessários menos de cinco minutos para a convergência, no segundo caso a convergência foi obtida depois de três horas de simulação. Nota-se que o número de variáveis é em torno de quinze vezes maior no segundo caso e o tempo gasto em torno de trinta e seis vezes maior, sendo que o tempo necessário na solução modal não se altera tão significativamente, demonstrando a dependência do tempo necessário para a otimização em relação ao número de variáveis.

9.3. Geometria Interpretada

Depois de feita a otimização, foi realizada uma interpretação da topologia obtida para a verificação dos resultados. A geometria interpretada foi modelada em um software CAD para corresponder fielmente a topologia obtida com a malha de 4000 elementos. Contudo, na região onde havia a presença de instabilidade de tabuleiro, foi interpretado que seria necessária a adição de um reforço. Após, foi feita uma nova análise modal via elementos finitos, onde o resultado pode ser visualizado na Figura 9.8.

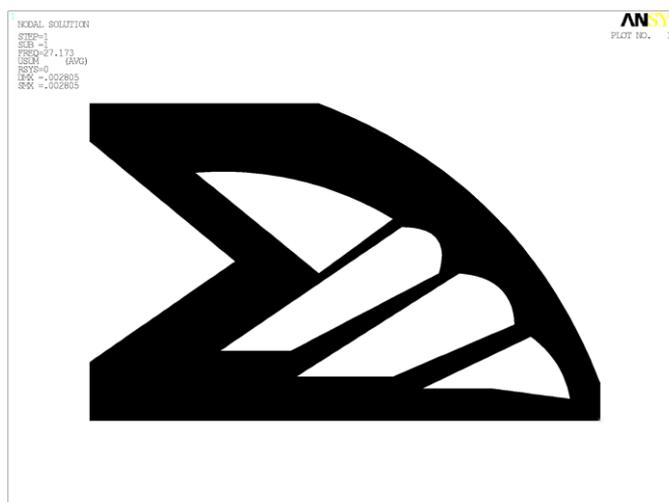


Figura 9.8 – Geometria interpretada.

Esta geometria apresentou um volume de $20,77 \text{ m}^3$ e a análise modal resultou em uma frequência de 27,173 Hz para o primeiro modo, se mostrando condizente com os resultados obtidos pela otimização.

10. CONCLUSÕES

Os resultados obtidos pela utilização da programação de otimização adotada se mostraram satisfatórios no que diz respeito ao alcance do objetivo proposto no caso estudado. Porém se observou certa dificuldade em se obter resultados sem a presença de instabilidade de tabuleiro e densidades intermediárias. Estes problemas são comuns em otimização topológica, mas poderiam ser amenizados ainda mais com o desenvolvimento de uma técnica de filtragem mais precisa.

Outro fator observado está relacionado com o tempo computacional gasto em otimização. Em problemas com poucas variáveis (poucos elementos) o tempo necessário para a análise por elementos finitos é maior que o necessário para a otimização por programação linear, no entanto esta relação se inverte para muitas variáveis de forma que o tempo gasto na otimização se torna muito maior que o necessário para análise de finitos. Isso ocorre devido ao fato de que a análise pode ser realizada na utilização de múltiplos processadores trabalhando em paralelo, enquanto o processo de otimização adotado não permite isto, pois depende de soluções seriadas.

Um estudo para verificar a possibilidade de implementação do processo de otimização no ambiente do software de finitos foi realizada para diminuir o tempo gasto com transferência de dados. Contudo, o software em questão não apresenta uma ferramenta de otimização de problemas lineares como o Simplex, o que exigiria a sua implementação e tornaria o trabalho inviável.

De uma forma geral, a otimização topológica se mostrou bastante útil para a diminuição de custo em problemas de engenharia, sendo uma ferramenta fundamental para realização de projetos que envolvam alto custo na fabricação de componentes, eliminando a necessidade de testes em vários protótipos.

Em trabalhos futuros a programação desenvolvida poderia ser facilmente estendida para casos tridimensionais, se tornando mais generalizada e aplicável a casos mais complexos. Fica também como sugestão, a implementação de outros tipos de restrição, como restrições de tensão, por exemplo, que são fundamentais em projetos de engenharia.

REFERÊNCIAS

- ANSYS; **“ANSYS Multiphysics Documentation”**, 2007.
- ARORA, J. S.; **“Introduction to Optimum Design”**, Elsevier Academic Press, 2nd edition, 1989.
- BENDSØE, M. P.; **“Optimization of Structural Topology, Shape, and Material”**, Springer-Verlag Berlin Heidelberg, 1995.
- CARDOSO, E. L.; **“Controle de Complexidade na Otimização Topológica de Estruturas Contínuas”**, Dissertação de Mestrado, Universidade Federal do Rio Grande do Sul, 2000.
- HAFTKA, R. T., GÜRDAL, Z.; **“Elements of Structural Optimization”**, Kluwer Academic publishers, 3rd edition, 1992.
- MOSMANN, R. M.; **“Otimização Topológica de Estruturas Contínuas Submetidas a Restrições de Flexibilidade, Volume e Frequência Natural”**, Dissertação de Mestrado, Universidade Federal do Rio Grande do Sul, 2003.
- SIGMUND, O., PETERSSON, J.; **“Numerical Instabilities in Topology Optimization: A Survey on Procedures Dealing with Checkerboards, Mesh-dependencies and Local Minima”**, Structural Optimization, vol. 16, pp. 68-75, 1998.

APÊNDICE A - Segundo problema analisado

Foi analisado o problema ilustrado na Figura A.1 a seguir, com dimensões de $2 \times 1 \times 0,001$ metros, utilizando os mesmos elementos finitos e propriedades de material do problema analisado anteriormente. Esta geometria possui a base engastada e massas concentradas de 10 kg nos pontos indicados. Neste caso objetiva-se a criação de reforços para a estrutura básica representada pelos contornos em preto.

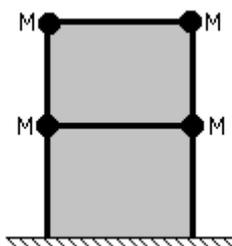


Figura A.1 – Segunda estrutura analisada.

A malha utilizada foi de 1250 elementos, expoente de penalização igual a $1/3$, variação de densidades entre 10^{-6} e 1, limites móveis iniciais iguais a 0,05, tolerância para a convergência de 10^{-7} e restrição de frequência de 40 Hz. A topologia obtida é apresentada na Figura A.2.

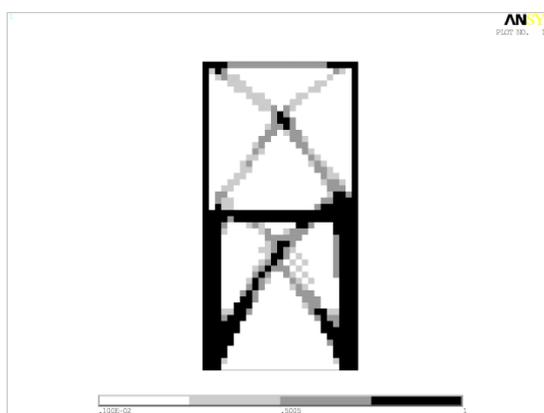


Figura A.2 – Topologia obtida no segundo problema.

Neste problema a redução de volume chegou a 65%, como pode ser visto na Figura A.3 que apresenta a convergência da função objetivo. Na Figura A.4 é apresentada a variação da restrição.

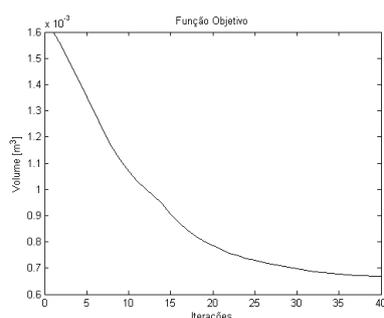


Figura A.3 – Convergência da função objetivo para o segundo problema.

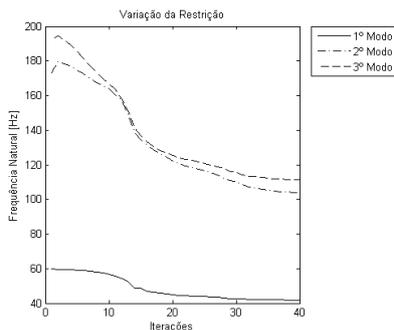


Figura A.4 – Variação da restrição para o segundo problema.

A interpretação para este caso apresentou um volume de $0,66 \times 10^{-3} \text{ m}^3$ e frequência de 43 Hz para o primeiro modo. A geometria interpretada pode ser visualizada na Figura A.5.

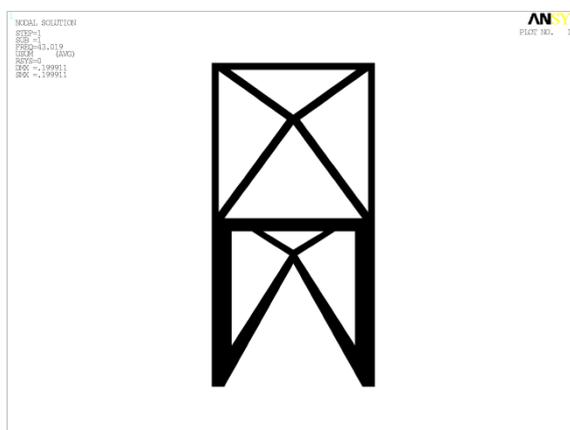


Figura A.5 – Geometria interpretada para o segundo problema.

APÊNDICE B - Programação realizada no software Matlab

A seguir é apresentada uma cópia da programação feita no software Matlab para a execução de otimização do problema analisado neste trabalho:

```

*****EXECUTA ROTINA DE OTIMIZAÇÃO*****
*****PARA MINIMIZAÇÃO DE VOLUME*****
*****COM RESTRIÇÃO DE FREQUÊNCIA NATURAL*****

%-----Retira dados iniciais para pré-processamento-----
clear
clc
! "C:\Program Files\ANSYS Inc\vi21\ANSYS\bin\winx64\ansys121.exe" -b -i
"C:\Caminho\inicia.dat" -o "C:\Caminho\probl.out"

%#####Dados de entrada

Frcmin = 27; %Frequência mínima em Hz
pen = 1/5; %Expoente de penalização da função objetivo
raio_busca = 0.11; %Raio do filtro
ajuste_peso = 0.3; %Ajuste de peso para o filtro
limmax = 1; %Valor máximo das densidades
limmin = 1e-3; %Valor mínimo das densidades
limite = 0.05; %Valor do limite móvel inicial
tol = 1e-3;
nelem = load ('nelem.dat'); %Arquivo com número total de elementos
nnos = load ('nnos.dat'); %Arquivo com número total de nós
A = load ('elems.dat'); %Arquivo com conectividade dos elementos
B = load ('nodes.dat'); %Arquivo com posições dos nós
C = load ('centroides.dat'); %Arquivo com centroides dos elementos

nnoselem = 4; %Número de nós do elemento utilizado

%Funções de interpolação do elemento utilizado
syms s t real
N1=(1/4)*(1-s)*(1-t);
N2=(1/4)*(1+s)*(1-t);
N3=(1/4)*(1+s)*(1+t);
N4=(1/4)*(1-s)*(1+t);
Nv=[N1;N2;N3;N4];

%Matriz constitutiva
E = 210e9; %Módulo de elasticidade em Pa
ni = 0.3; %Coeficiente de Poisson
C1 = E/(1-ni^2);

C2 = (E*ni)/(1-ni^2);
C3 = E/(2*(1+ni));
D = zeros(3);
D(1,1) = C1;
D(2,2) = C1;
D(2,1) = C2;
D(1,2) = C2;
D(3,3) = C3;

%#####Calcula a matriz de rigidez de cada elemento

XG = zeros(nnoselem,2);

for i = 1:1
disp('Calculando matriz de rigidez do elemento:')
disp(i)
l = 1;
for j = 2:nnoselem+1
for k = 1:nnos
if A(i,j) == B(k,1)
XG(l,1) = B(k,2);
XG(l,2) = B(k,3);
l = l+1;
end
end
end
X = XG'*Nv;
Jb(1,1) = diff(X(1,1),s);
Jb(1,2) = diff(X(1,1),t);
Jb(2,1) = diff(X(2,1),s);
Jb(2,2) = diff(X(2,1),t);
DNX = jacobian(Nv, [s t])*Jb^(-1); %Componentes da Matriz de
deformações Be
J = det(Jb);
Be(1,1) = DNX(1,1);
Be(3,2) = DNX(1,1);
Be(2,2) = DNX(1,2);
Be(3,1) = DNX(1,2);
Be(1,3) = DNX(2,1);
Be(2,4) = DNX(2,1);
Be(2,4) = DNX(2,2);
Be(3,3) = DNX(2,2);

```

```

Be(1,5) = DNX(3,1);
Be(3,6) = DNX(3,1);
Be(2,6) = DNX(3,2);
Be(3,5) = DNX(3,2);
Be(1,7) = DNX(4,1);
Be(3,8) = DNX(4,1);
Be(2,8) = DNX(4,2);
Be(3,7) = DNX(4,2);
R = Be'*D*Be*J;
for m = 1:8
    for n = 1:8
        F = matlabFunction(R(m,n));
        Ke(m,n,i) = dblquad(F,-1,1,-1,1);
    end
end
end
disp('Pré-processamento concluído com sucesso!')

densidade = ones(nelem,1);
save densidades.dat densidade -ascii -double

%-----Inicia a otimização-----
iter = 0;
convergencia = 10;
limite = limite*ones(nelem,1);
while convergencia > tol

!\"C:\Program Files\ANSYS Inc\v121\ANSYS\bin\winx64\ansys121.exe\" -b -i
\"C:\Caminho\otimizacao.dat\" -o \"C:\Caminho\probl.out\"

iter = iter+1;
disp('Calculando iteração:')
disp(iter)
dens(:,iter) = densidade;

G = load('volumes.dat'); %Volume dos elementos
M1 = load('mode01.dat'); %Frequência do 1º modo
M2 = load('mode02.dat'); %Frequência do 2º modo
M3 = load('mode03.dat'); %Frequência do 3º modo
V = load('displacements_mode01.dat'); %Deslocamentos do 1º modo

obj = ((pen*(densidade(:,1).^pen)).*G)';
vol = G';
lambda = M1(1,2)^2; %Autovalor do 1º modo
Dlambda = zeros(1,nelem); %Vetor com derivadas do autovalor
avet = zeros(8,1); %Autovetor do elemento
densidadeold = densidade;
%obj = zeros(3,1);
liminf = zeros(nelem,1);
limsup = zeros(nelem,1);
for i = 1:nelem
    %Cálculo das Derivadas
    l = 1;
    for j = 2:nnoselem+1
        for k = 1:nnos
            if A(i,j) == V(k,1)
                avet(l,1) = V(k,2);
                avet(l+1,1) = V(k,3);
                l = l+2;
            end
        end
    end
    DKe = densidadeold(i,1)*Ke(:,l,1); %Derivada da matriz de rigidez do
elemento
Dlambda(l,i) = avet'*DKe*avet;
%Filtragem
c = 0;
d = 0;
cont = 0;
peso = 0;
peso_medio = 0;
for m = 1:nelem
    if i ~= m
        dx = C(m,2)-C(i,2);
        dy = C(m,3)-C(i,3);
        raio = sqrt(dx^2+dy^2);
        if raio <= raio_busca
            c = c+dens(m,iter)*G(m);
            d = d+G(m);
            peso = peso+(raio_busca-raio)/raio_busca;
            cont = cont+1;
        end
    end
end
if cont > 0
    peso_medio = ajuste_peso*(peso/cont);
    dens(i,iter) =
(dens(i,iter)*G(i)+peso_medio*c)/(peso_medio*d+G(i));
end
%Alteração dos limites móveis
if iter > 2
    dif1 = dens(i,iter)-dens(i,iter-1);
    dif2 = dens(i,iter-1)-dens(i,iter-2);
    sinal = dif1*dif2;
    if sinal > 0
        if dif1 && dif2 < 0
            limite(i,1) = limite(i,1)*1.05;
            liminf(i,1) = (dens(i,iter))-limite(i,1);
            limsup(i,1) = (dens(i,iter))+limite(i,1);
            if liminf(i,1) < limmin
                liminf(i,1) = limmin;
            end
            if limsup(i,1) > limmax
                limsup(i,1) = limmax;
            end
        end
    else
        limite(i,1) = limite(i,1)*0.95;
        liminf(i,1) = (dens(i,iter))-limite(i,1);
        limsup(i,1) = (dens(i,iter))+limite(i,1);
        if liminf(i,1) < limmin
            liminf(i,1) = limmin;
        end
        if limsup(i,1) > limmax
            limsup(i,1) = limmax;
        end
    end
end
else
    limite(i,1) = limite(i,1)*0.95;
    liminf(i,1) = (dens(i,iter))-limite(i,1);
    limsup(i,1) = (dens(i,iter))+limite(i,1);
end
end
end
end
end

if liminf(i,1) < limmin
    liminf(i,1) = limmin;
end
if limsup(i,1) > limmax
    limsup(i,1) = limmax;
end
end
else
    liminf(i,1) = (dens(i,iter))-limite(i,1);
    limsup(i,1) = (dens(i,iter))+limite(i,1);
    if liminf(i,1) < limmin
        liminf(i,1) = limmin;
    end
    if limsup(i,1) > limmax
        limsup(i,1) = limmax;
    end
end
end
end

Arest = -1*Dlambda;
breast = -1*(Frqmin^2)+lambda-Dlambda*densidadeold;
options = optimset('LargeScale','off','Simplex','on');
densidade = linprog(obj,Arest,breast,[],[],liminf,limsup,[],options);
fobj(iter) = vol*densidadeold;
freq1(iter) = M1(1,2);
freq2(iter) = M2(1,2);
freq3(iter) = M3(1,2);
save densidades.dat densidade -ascii -double
convergencia = abs(vol*densidadeold-vol*densidade);

end

!\"C:\Program Files\ANSYS Inc\v121\ANSYS\bin\winx64\ansys121.exe\" -b -i
\"C:\Caminho\plotagem.dat\" -o \"C:\Caminho\probl.out\"

disp('Otimização concluída!')
p = plot(fobj)
xlabel('Iterações')
ylabel('Volume [m^3]')
title('Função Objetivo')

set(0,'DefaultAxesColorOrder',[0 0 0],...
'DefaultAxesLineStyleOrder','-|-.|--|:');
plot(freq1)
hold all
plot(freq2)
hold all
plot(freq3)
xlabel('Iterações')
ylabel('Frequência Natural [Hz]')
title('Variação da Restrição')
h = legend('1º Modo','2º Modo','3º Modo',-1);
set(h,'Interpreter','none')

```

APÊNDICE C - Macros para extração de dados do software Ansys

A seguir é apresentada uma cópia das macros desenvolvidas para extração dos dados do software Ansys, necessários para o processo de otimização:

```

!=====Retira dados iniciais para pre-processamento=====

/nopr
/prep7
nfile = 'nelem.dat'
nfile = 'nnos.dat'
efile = 'elems.dat'
nfile = 'nodes.dat'
centfile = 'centroides.dat'

!-----Conta o numero de elementos e exporta-----

alls
esel,u,type,,2 !Exclui o elemento de massa concentrada
*get,nelem,elem,,count

*cfopen,%nfile%
*vwrite,nelem
%10d
*cfclose
*msg,,nfile
--- Numero de elementos escrito em %s

!-----Conta o numero de nós e exporta-----

alls
*get,nnos,node,,count

*cfopen,%nfile%
*vwrite,nnos
%10d
*cfclose
*msg,,nfile
--- Numero de nos escrito em %s

!-----Cria matriz com as posições dos nós e exporta-----

alls
*get,mxnd,node,,num,max
*dim,nmsk,,mxnd
*dim,nds,,mxnd,3
*vfill,nds(1,1),ramp,1,1
*vget,nmsk(1),node,1,nsel

*vmask,nmsk(1)
*vget,nds(1,2),node,1,loc,x
*vmask,nmsk(1)
*vget,nds(1,3),node,1,loc,y
!*vmask,nmsk(1)
!*vget,nds(1,4),node,1,loc,z

*cfopen,%nfile%
*vmask,nmsk(1)
*vwrite,nds(1,1),nds(1,2),nds(1,3)
%10d %16.9g %16.9g
*cfclose
*msg,,nfile
--- Nos escritos em %s

!-----Cria matriz de conectividade dos elementos e exporta-----

alls
esel,u,type,,2 !Exclui o elemento de massa concentrada
*get,mxel,elem,,num,max
*dim,elms,,mxel,5
*dim,emsk,,mxel
*vfill,elms(1,1),ramp,1,1
*vget,emsk(1),elem,1,esel

*vmask,emsk(1)
*vget,elms(1,2),elem,1,node,1
*vmask,emsk(1)
*vget,elms(1,3),elem,1,node,2
*vmask,emsk(1)
*vget,elms(1,4),elem,1,node,3
*vmask,emsk(1)
*vget,elms(1,5),elem,1,node,4

*cfopen,%efile%
*vmask,emsk(1)
*vwrite,elms(1,1),elms(1,2),elms(1,3),elms(1,4),elms(1,5)
%10d %10d %10d %10d %10d
*cfclose
*msg,,efile
--- Elementos escritos em %s

!-----Cria matriz com centróide dos elementos e exporta-----

alls
esel,u,type,,2 !Exclui o elemento de massa concentrada
*dim,ecent,,mxel,3
*vfill,ecent(1,1),ramp,1,1

*vmask,emsk(1)
*vget,ecent(1,2),elem,1,cent,x
*vmask,emsk(1)
*vget,ecent(1,3),elem,1,cent,y
!*vmask,emsk(1)
!*vget,ecent(1,4),elem,1,cent,z

*cfopen,%centfile%
*vmask,emsk(1)
*vwrite,ecent(1,1),ecent(1,2),ecent(1,3)
%10d %16.9g %16.9g

```

```

*cfclose
*msg,,centfile
--- Centroides escritos em %s

alls

!-----Apaga as variáveis-----

nfile=
nelem=
nfile=
nnos=
nfile=
efile=
mxnd=
nmsk=
nds=
mxel=
elms=
emsk=
centfile=
ecent=

*msg,,
--- Concluido
finish
/go

```

```

!=====Retira dados para pós-processamento e otimização=====

/nopr
mfilpre = 'displacements_mode'
mfilpst = 'dat'
nome = 'mode'
ext = 'dat'
volufile = 'volumes.dat'

!-----Cria vetor com volume dos elementos e exporta-----

alls
esel,u,type,,2 !Exclui o elemento de massa concentrada
*get,nelem,elem,,count
*DIM,numelem,ARRAY,nelem
*dim,volumes,,nelem,1
*DO,j,1,nelem
    *GET,numelem(j),elem,0,NUM,MIN
    *get,volumes(j),elem,numelem(j),volu
    esel,U,elem,,numelem(j)
*ENDDO
*cfopen,%volufile%
!*vmask,emsk(1)
*vwrite,volumes(1,1)
%16.9g
*cfclose
alls
*msg,,volufile
--- Volumes escritos em %s

!-----Cria matriz com deslocamentos nodais e vetor com frequências-----

/post1
set,last
allsel
esel,u,type,,2 !Exclui o elemento de massa concentrada

*get,lsbst,active,,set,sbst
*get,mxnd,node,,num,max

*dim,igvct,,mxnd,3
*dim,nmsk,,mxnd

*vfill,igvct(1,1),ramp,1,1
*vget,nmsk(1),node,1,nsel

*do,i,1,lsbst
set,i
*get,frq,active,,set,freq
*vmask,nmsk(1)
*vget,igvct(1,2),node,1,u,x
*vmask,nmsk(1)
*vget,igvct(1,3),node,1,u,y
!*vmask,nmsk(1)
!*vget,igvct(1,4),node,1,u,z

*if,i,lt,10,then
*cfopen,%mfilpre%0%i%.%mfilpst%
*else
*cfopen,%mfilpre%%i%.%mfilpst%
*endif

*vmask,nmsk(1)
*vwrite,igvct(1,1),igvct(1,2),igvct(1,3)
%16.9g %16.9g %16.9g
*cfclose

*if,i,lt,10,then
*cfopen,%nome%0%i%.%ext%
*else
*cfopen,%nome%%i%.%ext%
*endif
*vwrite,i,frq,
(F10.4,F10.4)
*cfclose

*msg,,i,frq
--- Modo %d, %10.2F Hz escrito
*enddo

alls

!----- Apaga as variáveis-----

igvct=
nmsk=
lsbst=
frq=
i=
mfilpre=
mfilpst=
mxnd=
nome=
ext=
volufile=
nelem=
numelem=
volumes=
j=

*msg,,
--- Concluido
finish
/go

```