



OTIMIZAÇÃO TOPOLÓGICA ESTRUTURAL COM MALHA NÃO ESTRUTURADA

Fernanda Bichet Link

Emanuel Moutinho Cesconeto

Ederval de Souza Lisboa

João Baptista Dias Moreira

Walter Jesus Paucar Casas

feulink@gmail.com

emanuelemc@hotmail.com

ederval.lisboa@yahoo.com.br

joaobapdmdias@gmail.com

walter.paucar.casas@ufrgs.br

Departamento de Engenharia Mecânica, Universidade Federal do Rio Grande do Sul

Rua Sarmiento Leite 425, Porto Alegre, 90050-170, Rio Grande do Sul, Brasil.

Abstract. *In the last decades the topological structural optimization method (ESO-Evolutionary Structural Optimization), developed by Xie and Steven, 1993, has been widely discussed emerging several variations of this method. Among them, the method Beso (Bi-directional Evolutionary Structural Optimization), developed by Huang and which allows the removal and addition of material, It is applied in different research in the field of engineering due to its robustness and efficiency. This article aims to extend the BESO (Huang e Xie, 2007) method in optimization problems of a structural system, seeking to minimize flexibility or compliance by applying unstructured and non-regular meshes in order to enable, in future studies, discretization in complex domains. The results obtained so far, present good performance compared to studies in the relevant literature applying the BESO method with regular and structured meshes.*

Keywords: *Topology optimization, FEM, non structured mesh, non-regular mesh*

INTRODUÇÃO

O método dos elementos finitos (MEF) é um método numérico robusto e aplicável a domínios complexos sendo capaz de solucionar problemas nas mais diversas áreas das ciências exatas e biológicas e, devido à sua grande aplicabilidade e eficiência, existem trabalhos com esta metodologia nas diversas especialidades, mais especificamente na área de interesse deste estudo, como transferência de calor, mecânica dos fluidos, mecânica dos sólidos, dentre outras.

Neste sentido, se valendo da grande capacidade do MEF nas últimas décadas diversos pesquisadores se voltaram para o estudo de métodos computacionais para otimização topológica, sendo definida por Olhoff e Taylor (1983) como o estabelecimento racional de um projeto estrutural, que é o melhor dentre todos os projetos possíveis, dentro de um objetivo prescrito e de um dado conjunto de limitações geométricas e/ou comportamentais, aplicável à diversos problemas na engenharia. Não há ainda um método de otimização considerado universalmente superior aos demais, novas formas são frequentemente desenvolvidas e testadas com o intuito de melhorar os métodos existentes.

Dentre os principais métodos destacam-se o método *Solid Isotropic Material with Penalization* (SIMP) (Bendsøe e Sigmund, 2003), o qual trata do método da homogeneização que lida com o problema de distribuição de massa e que pode ser considerado como um problema de síntese de material, o *Level Set* (Sethian e Wiegmann, 2000), que faz o monitoramento de interfaces e formas, envolvendo curvas e superfícies em uma malha cartesiana fixa e, por fim, o *Bidirectional Evolutionary Structural Optimization* (BESO) (Huang e Xie, 2007), que será o método abordado neste trabalho.

Diversos estudos vêm sendo feito com o método BESO, em sua maioria em domínios discretizados com malhas estruturadas, como pode ser visto em Xia *et al.* (2016) e Huang e Xie (2010). O objetivo principal do presente estudo é contribuir com o aprimoramento do método BESO, desenvolvendo e testando em uma abordagem aplicada a um domínio discretizado com malha não estruturada, i.e. que não tenham restrições sobre sua forma, para a minimização da flexibilidade estrutural. Os resultados alcançados são comparados com casos otimizados na literatura corrente utilizando malhas estruturadas uniformes convencionais.

1 MODELAGEM MECÂNICA E NUMÉRICA

O método BESO busca a otimização de uma função objetivo por meio da remoção ou inclusão de elementos de uma estrutura que está sujeita a uma restrição que se opõe à esta função, sendo esta restrição aplicada de forma gradual ao longo de várias iterações. Ele é composto de uma série de módulos, cada um tendo um objetivo definido, mas não uma forma fixa. Detalhes de uma implementação do método podem ser encontrados em Huang e Xie (2010).

1.1 Função objetivo

A função objetivo é uma função aplicada sobre o domínio da estrutura que representa a característica que se deseja minimizar ou maximizar. É utilizada para definir a função de sensibilidade, e como uma condição de parada. As funções mais frequentemente utilizadas são frequência natural (Lisboa *et al.*, 2017) e rigidez (Vicente, 2013), mas inúmeras outras também existem, como resposta em frequência (Yoon *et al.*, 2007), taxa de dissipação de energia devido à viscosidade de um fluido (Hansen *et al.*, 2005) e difusividade efetiva de nutrientes através de uma estrutura porosa (Chang, 2015).

A otimização topológica estrutural para um problema de maximização da rigidez é formulado como a minimização da flexibilidade média da estrutura, e pode ser definida conforme a Eq. (1.1),

$$\begin{aligned} &\text{Encontrar: } \mathbf{x} \\ &\text{Que minimiza: } C(\mathbf{x}_i) = \frac{1}{2} \mathbf{F}^T \mathbf{U} \\ &\text{Sujeito à: } \mathbf{V} * - \sum_{i=1}^{\text{nel}} V_i x_i = 0 \\ &\quad x_i = 0 \vee 1 \end{aligned} \tag{1.1}$$

Onde $\mathbf{x}$ representa o vetor da variável de projeto, V_i o volume individual do i -ésimo elemento, V^* o volume final prescrito da estrutura e x_i a variável da pseudo densidade do material. A função objetivo C do problema equivale à energia total de deformação da estrutura.

1.1 Método dos elementos finitos

As equações base para o domínio estrutural são as de movimento de corpo contínuo. Como hipóteses considera-se pequenas deformações e material homogêneo e isotrópico. São considerados apenas os esforços sobre dois eixos principais, portanto um domínio bidimensional. Com isso pode ser usada a Eq. (1.2) de elasticidade linear, como segue:

$$\sigma_s = D_s \varepsilon_s \tag{1.2}$$

Onde $\mathbf{D}_s$ é a matriz constitutiva do material, que depende de seu módulo de elasticidade (E) e coeficiente de Poisson (ν), e o subscrito s se refere ao nome do domínio, vindo do inglês *structural*. O domínio está sujeito às condições de contorno de um meio elástico contínuo, descritas pela equação de contorno do domínio, as condições de Dirichlet, e as condições de Neumann. A condição de contorno do domínio é dada pela Eq. (1.3) a baixo:

$$\tilde{\nabla}^T \sigma_s + b_s = \rho_s \frac{\partial^2 u_s}{\partial t^2} \quad (1.3)$$

que para casos estáticos podem ser reduzida a seguinte equação:

$$\tilde{\nabla}^T \sigma_s + b_s = 0 \quad (1.4)$$

Onde, $\tilde{\nabla}^T$ é a matriz transposta do operador diferencial e b_s são as forças de corpo. As condições de Dirichlet dizem que o deslocamento na fronteira do domínio deve ser igual ao deslocamento prescrito nela, e as condições de Neumann dizem que os esforços na fronteira devem ser iguais aos esforços prescritos nela. Das Equações (1.2) e (1.4), bem como das condições de contorno, é possível calcular o campo de tensões σ_s e de deslocamentos u_s do domínio.

No entanto, a resolução analítica destas equações só é possível para casos simples, portanto é utilizado o método dos elementos finitos. Utilizam-se equações ponderadoras lineares que descrevem o campo de deslocamentos e tensões em um corpo em função de valores calculados em pontos específicos, chamados de nós. Para o caso estudado são utilizados elementos quadriláteros de quatro nós, conforme Fig. 1, portanto existem quatro funções ponderadoras chamadas funções de forma. Estas são escolhidas tal que seu valor seja um no seu nó referente, decaindo linearmente para zero sobre os outros nós. O deslocamento dentro do corpo é, portanto, dado pela soma destas quatro funções, cada uma delas multiplicada pelo deslocamento de seu nó referente. São considerados apenas os deslocamentos sobre os dois eixos principais, portanto dois graus de liberdade por nó. Isto pode ser escrito conforme a Eq. (1.5).

$$u_s^e = N_s^e \tilde{u}_s^e \quad (1.5)$$

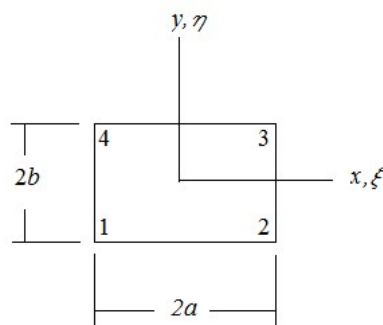


Figura 1. Geometria de um elemento retangular $\xi = x/a, \eta = y/b$

Onde N_s^e são as funções de forma organizadas em um vetor linha, u_s^e são os deslocamentos no domínio do elemento e $\tilde{u}_s^e$ são os valores dos deslocamentos nodais, ambos

organizados em um vetor coluna. Os componentes de N_s^e e u_s^e são funções de x e y , enquanto que os componentes de $\tilde{u}_s^e$ são constantes.

Para calcular os esforços e deslocamentos da estrutura, é utilizada a equação do movimento de corpos elásticos, com aceleração e amortecimento iguais a zero, como segue:

$$K u = F \quad (1.6)$$

Para resolver a Eq. (1.6), é necessário termos a matriz de rigidez global da estrutura K . Esta matriz é montada com base nas matrizes de rigidez de cada elemento. A matriz de rigidez K^e elementar é dada pela Eq. (1.7).

$$K^e = \int (\nabla N^e)^T D_S (\nabla N^e) dv \quad (1.7)$$

Seu desenvolvimento pode ser encontrado em Petyt (1990). Os diferenciais das funções de forma podem ser calculados analiticamente, pois elas são apenas funções bi-lineares, mas a integral deve ser calculada numericamente utilizando-se integração de Gauss. Como a equação envolve a multiplicação das funções o domínio pode estar sendo descrito por uma função bi-quadrática, ou bi-cúbica dependendo do sistema de coordenadas. Em ambos os casos é suficiente a utilização de quatro pontos sobre o domínio.

As funções de forma podem ser formuladas em outro sistema de coordenadas que é então mapeado ao sistema utilizado na estrutura. Neste caso deve-se usar um método para transformar as derivadas locais das funções de forma em derivadas no sistema global. Para tal é utilizada uma matriz Jacobiana, conforme Eq. (1.8). Os detalhes de sua formulação podem ser encontrados em Petyt, 1990.

$$J = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \quad (1.8)$$

Para o cálculo das matrizes de rigidez elementares usando funções de forma normalizadas é então usada a seguinte equação:

$$K^e = \int (J^{-1} \nabla N^e)^T D_S (J^{-1} \nabla N^e) \det(J) dv \quad (1.9)$$

Ao contrário de estruturas com elementos retangulares, não é possível a obtenção de uma matriz de rigidez elementar que dependa apenas de parâmetros do elemento como altura ou largura; o procedimento de integração numérica deve ser realizado para cada elemento para obtenção de sua matriz de rigidez.

1.2 Malha não estruturada

Existem dois tipos de malhas, a estruturada e a não estruturada. A malha estruturada refere uma malha na qual os elementos tem igual número de elementos vizinhos. Esta é implementada geralmente pelo método das diferenças finitas e apresenta-se muito conveniente para geometrias de domínio simples, além de requerer um gasto computacional aceitável. Na malha não estruturada não necessariamente cada elemento apresenta igual número de elementos vizinhos, geralmente está é implementada pelo método de elementos finitos. Esta última apresenta vantagens como flexibilidade para contornos irregulares de geometrias complexas, mas sua solução requer um gasto computacional maior em comparação com o da malha estruturada.

Uma malha estruturada é uma malha em que a numeração dos elementos segue uma ordem definida. Como exemplo, considere uma malha de quadriláteros, com 5 elementos na horizontal e 4 na vertical. Exemplos de malhas com estas características são mostrados na coluna esquerda da Fig. 2.

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20

(a)

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20

(b)

5	12	11	20	1
4	7	8	13	9
19	18	2	6	3
15	14	10	16	17

(c)

5	12	11	20	1
4	7	8	13	9
19	18	2	6	3
15	14	10	16	17

(d)

Figura 2. Exemplo de malhas: a) Malha regular e estruturada b) Malha não regular e estruturada c) Malha regular e não estruturada d) Malha não regular e não estruturada

Para um dado elemento interno i , considerando as malhas escritas na Fig. 2 (a) e (b), podemos dizer que o elemento à sua esquerda possui número $i-1$, o à sua direita $i+1$, acima $i-5$

e abaixo $i+5$. A vantagem disto é que não são necessárias informações extras para determinar a conectividade dos elementos; malhas não estruturadas dependem de uma matriz de conectividade que determina quais elementos estão ligados entre si, o que aumenta a complexidade da implementação e aumenta um pouco o custo computacional. Uma malha regular é uma malha onde todos os elementos são idênticos, conforme Fig. 2 (a) e (c). A grande vantagem de se trabalhar com malhas deste tipo é que só é necessário o cálculo e armazenamento de uma matriz de rigidez elementar; malhas não regulares necessitam do cálculo da matriz elementar de cada elemento, o que aumenta significativamente o custo computacional. Geralmente a otimização topológica de um sistema é realizada discretizando-o com uma malha estruturada e regular. Isto diminui o custo computacional e simplifica a implementação, mas impõe limitações quanto ao tipo de sistema que se pode otimizar. A proposta deste trabalho é implementar o método BESO com malhas do tipo apresentadas na Fig. 2 (d), ou seja, não estruturadas e não regulares, a fim de possibilitar seu uso em casos com geometrias cujo contorno são mais complexos. Uma alternativa ao uso deste tipo de malha é o uso de malhas convencionais mais refinadas; o aumento do número de elementos possibilita a discretização de curvas e outras características através de "escadas". Porém, o uso de mais elementos causa um aumento do custo computacional da otimização ao elevar significativamente o número de variáveis.

1.3 Discretização do domínio e variáveis otimizadas

O domínio deve ser discretizado em vários elementos, formando uma malha para análise em elementos finitos. A malha elementar permanece constante em uma dada execução do método. No método BESO cada elemento pode assumir um valor de dois valores possíveis:

$$x_i = \begin{cases} 1 \\ x_{\min} \end{cases} \quad x_i = \begin{cases} 1 & \text{se o elemento } i \text{ estiver presente} \\ x_{\min} & \text{se o elemento } i \text{ estiver ausente} \end{cases} \quad (1.10)$$

Este vetor é utilizado para calcular um novo coeficiente de elasticidade E_i para o elemento i , com base na elasticidade original, da seguinte forma:

$$E_i = (x_i)^p E_i^0$$

Onde p é um expoente chamado de fator de penalidade, como descrito por Huang e Xie, 2010.

1.4 Sensibilidade

Neste trabalho é usada a rigidez estrutural como característica a ser maximizada e, portanto a otimização busca a minimização da energia absorvida pela estrutura. Assim, devemos minimizar a seguinte função objetivo,

$$C = \frac{1}{2} \mathbf{f}^T \mathbf{u} \quad (1.11)$$

Onde $\mathbf{u}$ é o vetor de deslocamentos da estrutura e $\mathbf{f}$ é o seu vetor de forças. Este vetor pode ser escrito em função do deslocamento e da matriz de rigidez do sistema, $\mathbf{f} = \mathbf{K}\mathbf{u}$. Esta matriz de rigidez global é constituída pelas matrizes de rigidez elementares, $\mathbf{K}_i$. Estas dependem do coeficiente de elasticidade E_i de cada elemento estrutural, que por sua vez depende da variável de projeto x_i como mostrado na Eq.(1.10). Assim, pode-se escrever a matriz $\mathbf{K}$ da seguinte forma:

$$\mathbf{K} = \sum_{i=1}^n (x_i)^p \mathbf{k}_i^{0e} \quad (1.12)$$

Onde $\mathbf{k}_i^{0e}$ é a matriz de rigidez elementar do elemento i em função de E_i^0 ao invés de E_i e N é o número de elementos.

Ao se derivar $\mathbf{K}$ por um componente x_i específico temos como resultado apenas a matriz elementar $\mathbf{k}_i^{0e}$ do elemento i , pois $\frac{\partial x_j}{\partial x_i} = 0$ para $j \neq i$. Assim, derivando-se a função objetivo em função de x_i obtemos a seguinte equação:

$$\alpha_i = p(x_i)^{(p-1)} \frac{1}{2} (\mathbf{u}_i^e)^T \mathbf{k}_i^{0e} \mathbf{u}_i^e \quad (1.13)$$

Onde o índice e indica que estes são valores elementares. Os vetores $\mathbf{u}$ não dependem de x_i e, portanto permanecem iguais, mas todos os seus componentes que não se referem aos graus de liberdade do elemento i são multiplicados por zeros na matriz de rigidez e, portanto, podem ser descartados, restando o vetor de deslocamentos elementar $\mathbf{u}_i^e$. p só é relevante para métodos *soft-kill*; no caso *hard-kill* p deve ser igual e a equação é simplificada.

1.5 Filtro de estabilização

Este filtro modifica o valor de sensibilidade de cada elemento, levando em consideração os valores que cada um deles possuía em iterações anteriores. Um exemplo é a média simples, onde a sensibilidade da iteração atual de um elemento é dada pela média do valor calculado na iteração presente e do calculado na iteração anterior:

$$\alpha_i^f = \frac{(\alpha_i + \alpha_{i-1}^f)}{2} \quad (1.14)$$

Onde α_i^f é o valor de sensibilidade filtrado para a iteração i atual de um elemento, α_i é o valor não filtrado calculado para o mesmo elemento na mesma iteração e α_{i-1}^f é a sensibilidade filtrada dele na iteração anterior. Assim, o histórico de valores calculados para um dado elemento influencia no valor atual dele, com valores em iterações mais distantes tendo influências cada vez menores. Com isso, mudanças bruscas no domínio não causam

impactos tão grandes na otimização, que pode continuar convergindo de uma forma mais suave.

2 METODOLOGIA

Foi programado um software em MATLAB[®] que implementa um método BESO *hard-kill* 2D. A escolha deste método sobre um *soft-kill* se deve ao menor custo computacional do primeiro, que quando aliado a métodos de filtragem adequados não possui desvantagem frente ao segundo. Os específicos da implementação são descritos nesta seção. O procedimento geral do programa é mostrado na Fig. 3:

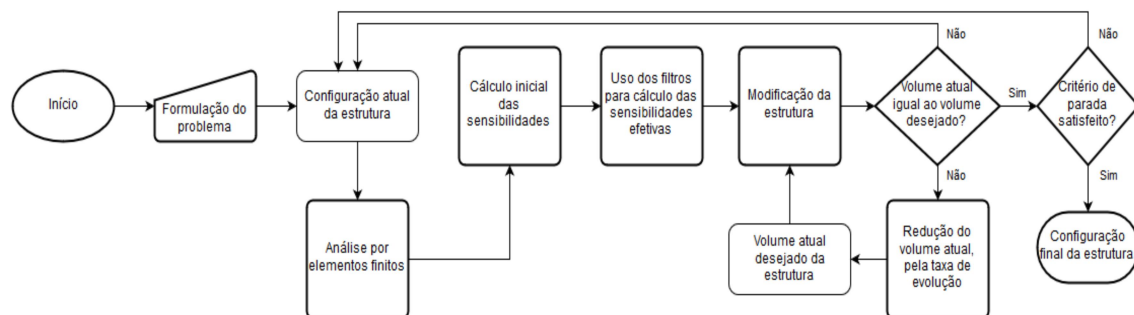


Figure 3. Esquema do procedimento geral de otimização pelo método BESO usado

2.1 Formulação do problema

Os problemas analisados consistem da otimização de um sistema estrutural, buscando-se a minimização da flexibilidade ou *compliance*, que equivale à maximização da rigidez. A malha dos sistemas não é modificada durante a otimização. Todos os problemas são primeiramente descritos por uma série de dados referentes à sua malha (posição dos nós e matriz de conectividade), condições de contorno (forças e deslocamentos prescritos sobre determinados graus de liberdade), propriedades do material estrutural (coeficiente de Poisson e módulo de Young). Também se informa a fração de volume final desejada e a taxa de evolução.

2.2 Cálculo inicial das sensibilidades

O cálculo das sensibilidades é primeiramente dado pela Eq.(1.13) com $p = 1$. Ela é calculada utilizando a matriz de rigidez estrutural elementar e as componentes do vetor de deslocamentos referentes aos graus de liberdade de cada elemento, caso este faça atualmente parte da estrutura. Isto resulta em um valor por elemento ativo.

A sensibilidade é então calculada segundo a Eq. (2.1), com base na utilizada por Vicente, (2013), mas modificada para considerar o volume dos elementos conforme Huang e Xie (2010).

$$\alpha_i = \frac{\partial C}{\partial x_i} = \begin{cases} \frac{[\frac{1}{2}(u_s)^T K_s^i u_s + (u_s)^T L_{sf}^i p_f]}{V_e} & x_i = 1 \\ 0 & x_i \neq 1 \end{cases} \quad (2.1)$$

2.3 Filtragem dos valores de sensibilidade

2.3.1 Filtro de independência de malha

Para o cálculo da sensibilidade nodal, primeiramente se determina os nós dentro de um certo raio, chamado de $r_{\min}$, ao redor do centroide de cada elemento. Idealmente este deve ser escolhido de modo a incluir mais do que um elemento. Para estes nós é calculada a distância entre o centroide e o nó. Este processo é exemplificado pela Fig.4.

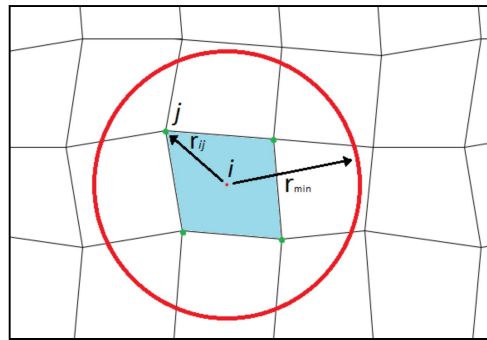


Figure 4. Visualização do processo de determinação dos nós próximos à um elemento e do cálculo de sua distância

Feito isso, a sensibilidade de cada nó é dada por uma média ponderada da sensibilidade de todos os elementos próximos, segundo a seguinte equação:

$$\alpha_j^n = \sum_{i=1}^M \omega_{ij} \alpha_i^e \quad (2.2)$$

Onde ω_i são os pesos de ponderação correspondente ao elemento i para o nó j , tal que $\sum_{i=1}^M \omega_{ij} = 1$, e M é o número de elementos próximo ao nó. Os pesos são dados pela seguinte equação:

$$\omega_{ij} = \frac{1}{M-1} \left(1 - \frac{r_{ij}}{\sum_{i=1}^M r_{ij}} \right) \quad (2.3)$$

Onde r_{ij} é a distância entre o nó j e o elemento i .

Com os valores de sensibilidade nodal calculados, o valor de sensibilidade elementar filtrado é dado por:

$$\alpha_i = \frac{\sum_{j=1}^M \beta_{ij} \alpha_j^n}{\sum_{j=1}^K \beta_{ij}} \quad (2.4)$$

Onde K é o número de nós próximos ao elemento i e β_{ij} é igual à $r_{min} - r_{ij}$

É proposta então uma nova forma de aplicar o filtro. Após o cálculo dos pesos de ponderação, o método é constituído apenas de multiplicações simples e somatórios. Assim, é possível escrevê-lo em forma vetorial:

$$\alpha^n = \omega \alpha^e \quad (2.5)$$

$$\alpha = \beta \alpha^n \quad (2.6)$$

Onde os α^n , α^e e α possuem as sensibilidades nodais, elementares não filtradas e elementares filtradas respectivamente, ω possui os pesos de ponderação nodais e β possui os pesos de ponderação elementar. α^e e α são vetores $N \times 1$, onde N é o número de elementos, α^n é um vetor $Y \times 1$, onde Y é o número de nós, ω é uma matriz $Y \times N$ e β é uma matriz $N \times Y$. Combinando as Eqs. (2.5) e (2.6) podemos escrever:

$$\alpha = \beta \omega \alpha^e \quad (2.7)$$

que pode ser reescrita como,

$$\alpha = \mathbf{H} \alpha^e \quad (2.8)$$

Onde $\mathbf{H}$ é a matriz $N \times N$ nomeada de matriz de suavização. Desta forma não é necessário calcular ou armazenar valores de sensibilidade nodais, nem duas matrizes separadas. Esta matriz ainda é computacionalmente cara para ser gerada, mas depende apenas dos valores das coordenadas dos nós e elementos e, portanto, é fixa para um dado problema. Assim, ela só precisa ser calculada na primeira iteração, sendo então reutilizada nas iterações seguintes.

Como este filtro não leva em consideração a diferença de volume, ou área, dos elementos, sua utilização em sistemas com muita diferença de área entre elementos vizinhos pode provocar

resultados indesejados. Para amenizar este efeito, ao invés de se dividir a energia do elemento pela sua área antes, ou depois, da filtragem, se divide pela raiz quadrada da área duas vezes: antes e depois de ser aplicado o filtro. Computacionalmente, no entanto, só são mantidos os valores da iteração atual e da anterior.

2.3.2 Filtro de estabilização

A proposta é de um filtro que faz a média dos valores de sensibilidade elementar da iteração atual com os da iteração anterior. Se escrito em forma expandida, é possível ver que todos os valores anteriores influenciam nos valores atuais:

$$\alpha_i^f = \frac{\alpha_i}{2} + \frac{\alpha_{i-1}}{4} + \frac{\alpha_{i-2}}{8} + \dots \quad (2.9)$$

2.4 Condição restritiva e taxa de evolução

A condição restritiva utilizada é uma restrição de volume, implementada como fração do volume inicial. Assim, a estrutura deve satisfazer a seguinte condição ao longo das iterações:

$$V_i = \begin{cases} V_0(1 - iER), & \text{se } V_0(1 - iER) > V^* \\ V^* & \text{se } V_0(1 - iER) < V^* \end{cases} \quad (2.10)$$

Onde V_i é o volume da estrutura do sistema na iteração i , V_0 é o seu volume inicial, ER é a taxa de evolução e V^* é o volume final desejado, que pode ser dado por V_{0vp} onde v_p é a fração do volume inicial que se deseja que a estrutura otimizada final possua. Por meio desta condição o volume da estrutura é gradativamente reduzido a cada iteração, até ser atingida a fração do volume inicial desejada.

2.5 Modificação da estrutura

Para implementar estas características, primeiramente devem-se determinar quais elementos previamente removidos poderão vir a ser reincluídos na estrutura. Para tal, é usado o método da bisseção para determinar um valor de corte da sensibilidade. Este método é usado para encontrar um valor x' para uma função $y(x)$ tal que $y(x') = 0$. Neste caso, x' será um valor de corte para a sensibilidade elementar, e $y(x)$ será a soma do volume de todos os elementos ativos que possuam valores de sensibilidade maiores do que o valor de entrada x . Isto pode ser escrito da seguinte forma:

$$\text{Encontre: } x' \mid y(x') = 0 \quad (2.11)$$

$$y(x) = \sum_{j=1}^N V_j'^e(x) - V_{ad} \quad (2.12)$$

$$V_j^e(x) = \begin{cases} V_j^e & \text{caso } j \text{ se refira a um elemento ativo } \alpha_j > x \\ 0 & \text{caso contrário} \end{cases} \quad (2.13)$$

Onde N é o número de elementos, V_{ad} é o volume que pode ser adicionado a cada iteração, V_e^j é a área do elemento j e α_j é o valor de sensibilidade filtrada dele. Os elementos ativos são nesse caso os elementos não estruturais. Os elementos que podem ser adicionados são então os elementos ativos com valores de sensibilidade maiores do que x' .

Determinados quais elementos podem ser re-adicionados, repete-se o mesmo processo, mas desta vez os elementos ativos serão os elementos estruturais, e V_{ad} é substituído por V_i , que é a fração de volume definida pela condição restritiva para a iteração i atual. No caso 2D as áreas são para efeito de cálculo equivalentes ao volume, então no algoritmo são usadas áreas ao invés de volumes. Além disto, alguns elementos estruturais podem ser definidos como obrigatórios, e não podem ser removidos, e a eles são efetivamente dados valores de sensibilidade infinitos para este teste.

2.6 Condição de parada

Para o fim da otimização são considerados três quesitos. Primeiramente a condição restritiva, que é aplicada gradativamente, deve estar sendo obedecida por completo para que a otimização seja considerada terminada.

Após o primeiro quesito ser alcançado, algumas iterações ainda são executadas até que a mudança no valor da função objetivo, no caso a *compliance* da estrutura, seja menor do que um valor de tolerância. Durante estas iterações o volume da estrutura não varia muito, mas sua configuração pode mudar enquanto elementos são adicionados e removidos. Esta condição é necessária porque como mencionado os diferenciais usados para modificar a estrutura são apenas aproximações, que embora suficientes para guiar a redução do volume durante a primeira parte da otimização não são capazes de chegar à uma posição ótima em apenas uma iteração. Assim, durante várias iterações extras a estrutura se estabiliza e converge para sua forma final.

Concorrentemente com o segundo quesito, o valor da função objetivo da iteração atual é comparado com o melhor valor encontrado desde o alcance do primeiro quesito. Caso este seja melhor, ele é armazenado como o melhor valor, e a configuração da estrutura também são salvos. Caso o processo de otimização falhe em alcançar uma melhora em certo número de iterações ele é finalizado. Matematicamente, estas condições são representadas pelas seguintes equações:

$$V_i = V^* \quad (2.14)$$

$$abs\left(\frac{C_i - C_{i-1}}{C_i}\right) < tol \text{ ou } C_{(i-K)...i} > C_{min} \quad (2.15)$$

Onde V_i é o valor do volume da estrutura na iteração i , V^* é o volume final desejado, C_i é o valor da função objetivo na iteração i , tol é a variação aceitável, em porcentagem, para considerar que a forma da estrutura está convergida, e K é o número de iterações que podem ter valores de C piores do que C_{min} , o menor valor encontrado, antes da otimização ser interrompida. Ambos V^* , tol e K são dados de entrada da otimização.

3 VERIFICAÇÃO DOS RESULTADOS

Realizou-se uma comparação entre os resultados de literatura e os obtidos pelo algoritmo desenvolvido. Foi considerado uma viga 2D engatada-livre com uma carga aplicada na extremidade livre, conforme Fig. 5. O domínio é retangular com as dimensões de 160 mm $\times$ 40 mm, discretizado com 160 $\times$ 40 elementos de tensão plana com quatro nós cada. O objetivo é maximizar a frequência fundamental com uma fração de volume prescrito de 50%. A Fig. 6 mostra as geometrias alcançadas através do método Hard-kill BESO e Soft-kill BESO, ambos com malha estruturada e regular.

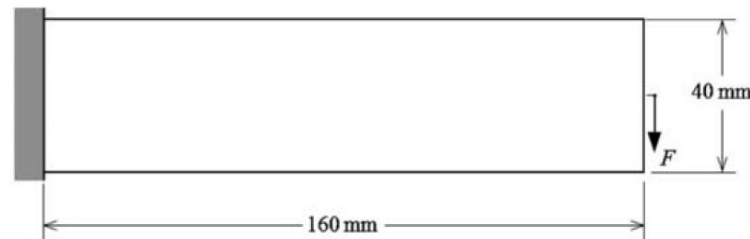


Figura 5. Exemplo utilizado



Figura 6. Geometrias alcançadas com (a) Hard-kill BESO tradicional, e (b) Soft-kill BESO

Nas Figuras 7 e 8 são apresentados os resultados obtidos considerando a implementação com malhas não estruturadas e não regulares, e elementos com incremento de raio de irregularidade de 0.3 e 0.5, respectivamente, com seus devidos históricos.

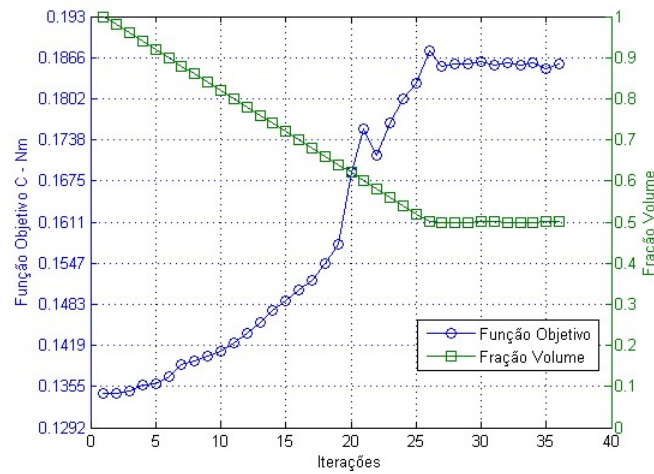
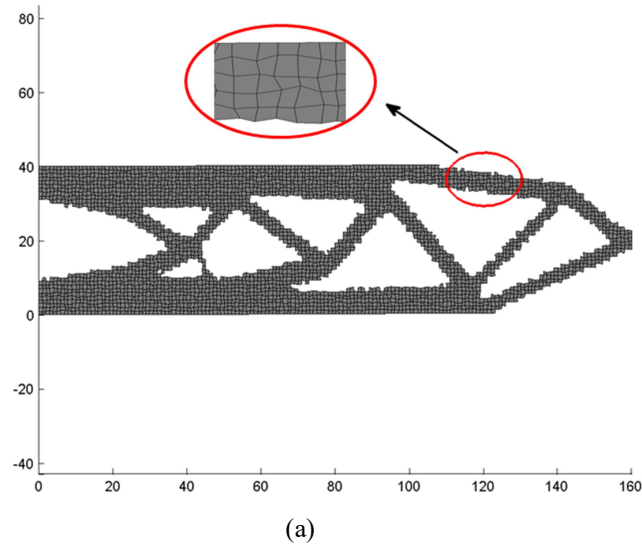


Figura 7. Elementos discretizados com um incremento de 0.3: a) Malha não regular e não estruturada 160×40 e respectivo domínio otimizado b) Relação da distribuição Distribuição da *Compliance* em relação ao número de iterações para fração de volume estipulada em 50%

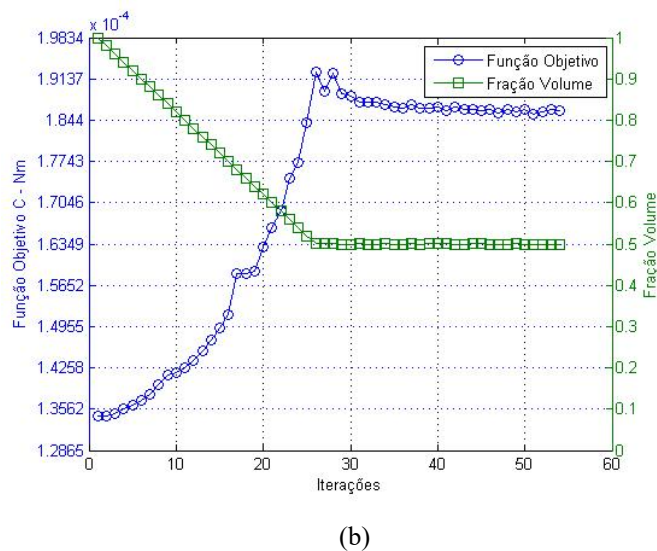
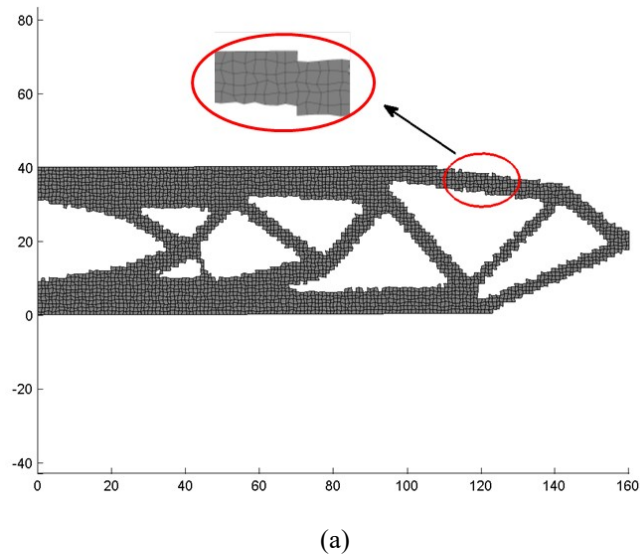


Figura 8. Elementos discretizados com um incremento de 0.5: a) Malha não regular e não estruturada 160×40 e respectivo domínio otimizado b) Relação da distribuição Distribuição da *Compliance* em relação ao número de iterações para fração de volume estipulada em 50%

Tabela 1: Comparativo das flexibilidades obtidas.

Métodos	<i>Compliance</i> (Nmm)	Iterações
Hard-kill BESO	181.79	52
Soft-kill BESO	181.71	46

SIMP	201.70	44
Continuation	180.69	36
Hard-kill BESO não estruturada (0.3)	185.68	34
Hard-kill BESO não estruturada (0.5)	186.01	36

4 CONCLUSÃO

Na validação a comparação entre os casos demonstra que o método implementado é capaz de otimizar casos com malha regular ou não regular. O algoritmo desenvolvido contém uma implementação do método FEM como uma de suas partes. Com isso, não há dependência de outros programas comerciais para realizar otimizações, apesar de que é relativamente simples fazer uma interface com um programa como o ANSYS, por exemplo, mas foi escolhida a criação de um módulo FEM. Dessa forma, o aprimoramento do método BESO foi possível implementando um código de otimização topológica com malha não estruturada em MATLAB[®] sem nenhuma dependência à programas comerciais externos. O algoritmo implementado foi validado com um caso da literatura para demonstrar sua capacidade e robustez. Assim, foi atingido o objetivo de implementar uma melhoria ao método BESO. Cabe ainda destacar que durante o desenvolvimento do trabalho uma nova proposta para a aplicação do filtro foi apresentada, e na comparação dos resultados obtidos com o de bibliografias existentes, se mostrou plenamente eficiente.

AGRADECIMENTOS

A pesquisa descrita neste artigo foi apoiada financeiramente por agências de financiamento nacionais brasileiras, através do CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico (concessões 140081 / 2015-1, 148895 / 2016-6 e 148887 / 2016-3) do Ministério da Ciência Tecnologia e Inovação; e a Coordenação para a Melhoria do Pessoal de Educação Superior CAPES (concessão 3686) do Ministério da Educação.

REFERÊNCIAS

- Bendsøe, M. P.; Sigmund, 2003. *Topology optimization: theory, methods, and applications*. Berlin: Springer-Verlag, 370 p.
- Chang, C. , 2015. *Design and Topology Optimization of Tissue Scaffolds*. Tese de doutorado, University of Sydney, Australia.

Hansen, A. G., Sigmund, O., Haber, R. B., 2005. *Topology optimization of channel flow problems*. Struct. Multidisc. Optim., USA, v.30, p.181–192.

Huang, X., Xie, Y.M., 2007. *Convergent and mesh-independent solutions for the bi-directional evolutionary structural optimization method*. Finite Elements in Analysis and Design, v. 43, 1039–1049.

Huang, X., Xie, Y. M., 2010 *Evolutionary Topology Optimization Of Continuum Structures: methods and applications*. 1a edição.ed. Chichester, UK: John Wiley & Sons, Ltd. 237p.

Lisboa, E.S., Moreira, J. B. D., Cesconeto, E. M., Medeiros, J. E. G., Ribeiro, T. S. and Casas, W. J. P., 2017. *Optimization of dynamic parameters with bidirectional evolutionary structural method of continuum structures*. “XVII International Symposium on Dynamic Problems on Mechanics”, São Sebastião, Brazil, March.

Olhoff, N., Taylor J. E. , 1983. *On structural optimization*. Journal of Applied Mechanics, v.50, pp.1139-1151.

Petyt, M. 1990, *Introduction to Finite Element Vibration Analysis*, Cambridge University Press, New York, NY, USA.

Ramm, E., 1981. Strategies for tracing the nonlinear response near limit points. In Wunderlich, W., Stein, E., & Bathe, K. J., eds, *Nonlinear Finite Element Analysis in Structural Mechanics*, pp. 63–89. Springer-Verlag.

Sethian, J., Wiegmann, A., 2000. *Structural boundary design via level set and immersed interface methods*. Journal of Computational Physics, v. 163, n. 2, 489 – 528.

Vicente, W.M., 2013. *Otimização topológica evolucionária aplicada a sistemas fluido-estrutura*. Tese (Doutorado). Universidade Estadual de Campinas.

Xia, L., Xia, Q., Huang, X., Xie, Y. M., 2016. *Bi-directional Evolutionary Structural Optimization on Advanced Structures and Materials: A Comprehensive Review*. Archives of Computational Methods in Engineering. pp. 1-42.

Yoon, G. H., Jensen, J. S., Sigmund, O. 2007. *Topology optimization of acoustic–structure interaction problems using a mixed finite element formulation*. International Journal for Numerical Methods in Engineerings, USA, v.70, p.1049–1075.