

OTIMIZAÇÃO TOPOLÓGICA DE ESTRUTURAS LINEARMENTE ELÁSTICA DISCRETIZADAS COM ELEMENTOS TRIANGULARES

Jorge Alberto Martins de Almeida Junior

Resumo

Neste trabalho são apresentados os conceitos de dois métodos bem consolidados no estudo da Otimização Topológica, o SIMP e o BESO. Estes métodos foram implementados em um programa desenvolvido na linguagem python, utilizando pacotes adicionais que possibilitaram a implementação de uma interface gráfica com diversas funções necessárias para realizar a otimização através do Método dos Elemento Finitos. O programa apresentado deve ser utilizado apenas para fins acadêmicos e pode ser obtido em: <https://github.com/jorgjr/toPy>.

Palavras-chave: Otimização Topológica, Python, SIMP, BESO.

1. Introdução

Os primeiros estudos relacionado ao melhor aproveitamento da matéria-prima em estruturas foram realizados por Maxwell, em 1872, conforme Silva E. C. N. (2001), onde ele procurou calcular as tensões principais de estruturas dado um carregamento e sugeriu conceitualmente que a estrutura ótima seria constituída de materiais alinhados às direções das tensões principais, já que nelas não ocorrem tensões de cisalhamento, posteriormente Michell em 1904, obteve alguns avanços, porém para a época os resultados foram considerados apenas para fins acadêmicos, já que a produção das estruturas obtidas era praticamente impossível. A ideia de Maxwell se provou mais tarde estar correta, para estruturas com um único carregamento em casos de estudo para se obter uma estrutura com máxima rigidez e menor volume de material.

Depois de Michell somente na década de 80, após o surgimento dos computadores e do FEM (Finite Element Method), este problema voltou a ser estudado, principalmente após

a publicação de Bendsøe e Kikuchi (1988), onde foi apresentado a otimização topológica através do método da homogeneização.

Atualmente a otimização topológica vêm sendo utilizada nos estágios iniciais de projetos, procurando reduzir o número de protótipos necessários e também o custo computacional no uso de diversas simulações, já que com o uso das técnicas de otimização é possível obter uma ideia inicial de como será a aparência final do projeto.

O avanço das tecnologias de manufatura durante os últimos anos tornou real a possibilidade de desenvolver estes produtos e estruturas que possuem formas complexas, e isto fez com que este campo de estudos fosse amplamente abordado em diversos estudos.

Diversas abordagens de Otimização Topológica foram desenvolvidas, sendo as principais apresentadas por O. Sigmund (2013), a abordagem de densidade, level-set e as evolucionárias. Neste trabalho será apresentado dois métodos, o método BESO (Bidirectional Evolutionary Software Optimization), e o método SIMP (Solid Isotropic Material with Penalization).

2. Método BESO

O método ESO (Evolutionary Structure Optimization) realiza a remoção dos elementos mais ineficientes do domínio computacional de forma gradativa, porém além da remoção dos elementos do domínio, já a sua variação, o método BESO (Bidirectional Evolutionary Structure Optimization) além de remover também adiciona elementos, dependendo do número da sensibilidade que este elemento apresentar, este número da sensibilidade determina a eficiência de cada elemento. Neste trabalho foi utilizada uma versão soft-kill do método BESO baseada na interpolação de material com penalização, onde o elemento nunca é totalmente removido do domínio, mas sim atribuído um valor de densidade relativa muito baixo, que apenas representa um elemento vazio, esta ideia tem como base o método SIMP. O problema de otimização para esta variação do método BESO pode ser formulado como:

$$\text{Minimizar} \quad C = \sum_{i=1}^N \frac{1}{2} (x_i)^p u_i^T k_i u_i \quad (2.1)$$

$$\text{Sujeito a} \quad V^* - \sum_{i=1}^N V_i x_i = 0 \quad (2.2)$$

$$x_i = x_{min} \text{ ou } 1 \quad (2.3)$$

Onde C é a função objetivo, u é o vetor de deslocamento, k_i é a matriz de rigidez, ambos calculadas através do FEM, N é o número de elementos dentro do domínio computacional discretizado, V^* é o volume determinado que a peça tenha após o processo de otimização, V_i é o volume de cada elemento, e x_i é o valor de densidade relativa (também chamado de vetor da variável de projeto), sendo x_{min} igual a 0.001 para representar elementos vazios e assim evitar problemas durante a análise de elementos finitos, ou 1 para elementos sólidos, já o valor de p é valor da potência de penalidade, mais detalhes sobre a penalidade podem ser encontradas em O. Sigmund (1995).

A cada iteração, ocorre a atualização das propriedades do material de cada elemento:

$$E_{(x_i)} = E_0 x_i \quad (2.4)$$

Onde E_0 é o Módulo de Young, e x_i é o valor da densidade relativa. O volume pretendido que a estrutura terá ao final da iteração atual é dado por:

$$V_k = V_{k-1}(1 - ER) \quad (2.5)$$

Onde V_k é o volume da iteração atual, V_{k-1} é o volume da estrutura na iteração anterior, e ER é a proporção de evolução.

O valor de x_i é determinado após o cálculo do número de sensibilidade, que é dado por:

$$\alpha_i = \frac{1}{2} (x_i)^{p-1} u_i^T k_i u_i \quad (2.6)$$

Onde α_i é o número de sensibilidade, porém se apenas for realizado o cálculo da sensibilidade irá ocorrer o chamado efeito do “tabuleiro”, que é o surgimento de regiões em que os valores da densidade relativa variam em x_{min} ou 1, conforme a Figura 1.

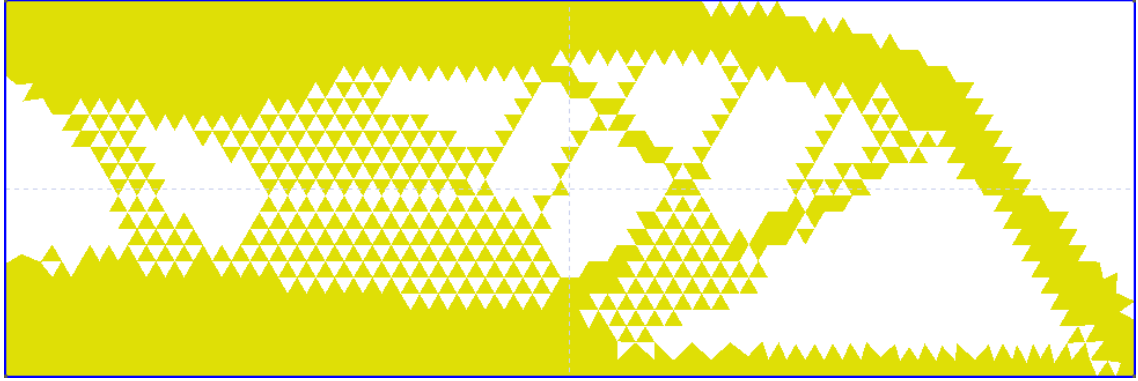


Figura 1: Efeito do "tabuleiro" em uma estrutura do tipo MBB Beam.

Para evitar que este efeito ocorra, após o cálculo de sensibilidade é realizado uma filtragem do valor da sensibilidade, conforme:

$$\alpha_i = \frac{\sum_{j=1}^M w(r_{ij}) \alpha_j}{\sum_{j=1}^M w(r_{ij})} \quad (2.7)$$

Onde M é a quantidade de elementos dentro da área circular limitada pelo raio mínimo, r_{min} , a partir do centróide do elemento, e $w(r_{ij})$ é um fator linear determinado por:

$$w(r_{ij}) = \begin{cases} r_{min} - r_{ij}, & \text{se } r_{ij} < r_{min} \\ 0, & \text{se } r_{ij} \geq r_{min} \end{cases} \quad (2.8)$$

Sendo r_{ij} a distância entre os centróides dos elementos dentro da região limitada. Este filtro utilizado é para o método BESO baseado na interpolação de material, sendo que o filtro do método original possui duas etapas, a primeira é o cálculo da sensibilidade nodal, e em seguida o cálculo da sensibilidade elemental, utilizando como base os valores da sensibilidade nodal.

Após a filtragem da sensibilidade o valor da densidade relativa de cada elemento pode ser atualizado através de uma rotina de otimização:

$$x_i = \begin{cases} 1, & \text{se } \alpha_i > \alpha_{th} \\ x_{min}, & \text{se } \alpha_i \leq \alpha_{th} \end{cases} \quad (2.9)$$

Em seguida atualiza-se as propriedades do material conforme a equação (2.4), e o volume de acordo com a equação (2.5), e assim pode ser realizada a análise de Elementos Finitos e uma nova iteração do ciclo de otimização até que ocorra a convergência:

$$\frac{|\sum_{i=1}^N (C_{k-i+1} - C_{k-N-i+1})|}{\sum_{i=1}^N C_{k-i+1}} \leq \tau \quad (2.10)$$

Onde C é o valor da função, K é o número da iteração atual, N é um valor definido para que o critério de convergência ocorra somente após a estabilização da otimização, τ é o valor de tolerância da convergência. Após a primeira iteração, os valores de sensibilidade também devem ser estabilizados conforme:

$$\alpha_i = \frac{\alpha_{i,k} + \alpha_{i,k-1}}{2} \quad (2.11)$$

Mais detalhes sobre a modelagem do método BESO, e do soft-kill BESO podem ser encontrados em X. Huang (2010) e Vu Truong Vu (2018).

3. Método SIMP

O método SIMP é uma abordagem da otimização topológica baseado na lei da potência, desenvolvida por Bendsøe e Kikuchi (1988), onde deseja-se minimizar o valor da energia de deformação total de um domínio computacional, onde os limites desse domínio determinam as dimensões máximas admissíveis para a geometria final.

O problema de otimização para o método SIMP pode ser formulado como:

$$\text{Minimizar} \quad C = U^T K U = \sum_{i=1}^N (x_i)^p u_i^T k_i u_i \quad (3.1)$$

$$\text{Sujeito a} \quad \frac{V(x)}{V_0} = f \quad (3.2)$$

$$0 < x_{min} \leq x \leq 1 \quad (3.3)$$

Onde $V(x)$ é o volume atual da geometria, V_0 é o volume total do domínio computacional e f é a fração de volume desejada para a geometria final. A cada iteração, ocorre a atualização das propriedades do material de cada elemento:

$$E_{(x_i)} = E_0 x_i \quad (3.4)$$

Onde E_0 é o Módulo de Young, e x_i é o valor da densidade relativa pode ser determinada conforme:

$$x_i = \begin{cases} \max(x_{min}, x_i - m), & \text{se } x_i B_i^\eta \leq \max(x_{min}, x_i - m) \\ x_i B_i^\eta, & \text{se } \max(x_{min}, x_i - m) < x_i B_i^\eta < \min(1, x_i + m) \\ \min(1, x_i + m), & \text{se } \min(1, x_i + m) \leq x_i B_i^\eta \end{cases} \quad (3.5)$$

Sendo m um valor de movimento-limite, e η é um coeficiente de amortecimento, sendo eles iguais a 0.2 e 0.5, respectivamente, mais informações sobre eles estão descritas em Bendsøe e Sigmund (2003). B_i é obtido através do Critério de Otimalidade que depende do número de sensibilidade que no método SIMP é dado por:

$$\frac{\partial C}{\partial x_i} = -p(x_i)^{p-1} u_i^T k_i u_i \quad (3.6)$$

Então B_i pode ser encontrado por:

$$B_i = \frac{-\frac{\partial C}{\partial x_i}}{\lambda \frac{\partial V}{\partial x_i}} \quad (3.7)$$

Onde λ é um multiplicador de Lagrange que pode ser obtido através de um algoritmo de biseção. Sendo que o número de sensibilidade deve antes passar por um filtro para evitar que ocorra o efeito do “tabuleiro” e que não tenham uma dependência da malha:

$$\frac{\partial \widehat{C}}{\partial x_i} = \frac{1}{x_i \sum_{f=1}^M \widehat{H}_f} \sum_{f=1}^M \widehat{H}_f x_f \frac{\partial C}{\partial x_f} \quad (3.8)$$

Onde M é o número de elementos pertencentes à uma região limitada pelo raio mínimo r_{min} , e \widehat{H}_f é um fator peso dado por:

$$\widehat{H}_f = r_{min} - dist(i, f) \quad (3.9)$$

Sendo $dist(i, f)$ a distância entre os centróides dos elementos dentro da região a ser analisada. Mais detalhes sobre a modelagem do método SIMP, podem ser encontrados em Bendsøe e Kikuchi (1988), O. Sigmund (2001) e Bendsøe e Sigmund (2003).

4. Programa

A ideia do desenvolvimento deste programa foi de aproximar os códigos top.m apresentado por Sigmund O. (2001), o código softbeso.m apresentado por X. Huang (2010), e as rotinas para a análise dos elementos finitos apresentadas por Kattan (2008) para um ambiente similar ao de softwares comerciais do tipo CAE (Computer Aided Engineering), que dão a possibilidade do usuário realizar algumas funções básicas, como importar um domínio computacional (geometria), adicionar as condições de contorno diretamente no domínio computacional, bem como visualizar a malha gerada e os resultados obtidos.

O programa diferentemente dos códigos, foi desenvolvido utilizando a linguagem python, com o auxílio de três pacotes: Gmsh, NumPy e SciPy. O pacote Gmsh traz diversas funções sendo as principais utilizadas a importação de arquivos, a geração de malha, e a criação de grupos físicos, além dessas funções ele possui uma interface gráfica que pode ser modificada. Os pacotes NumPy e SciPy são utilizados para resolver as diversas rotinas necessárias para a análise dos Elementos Finitos. Na figura 2 é apresentada a GUI modificada.

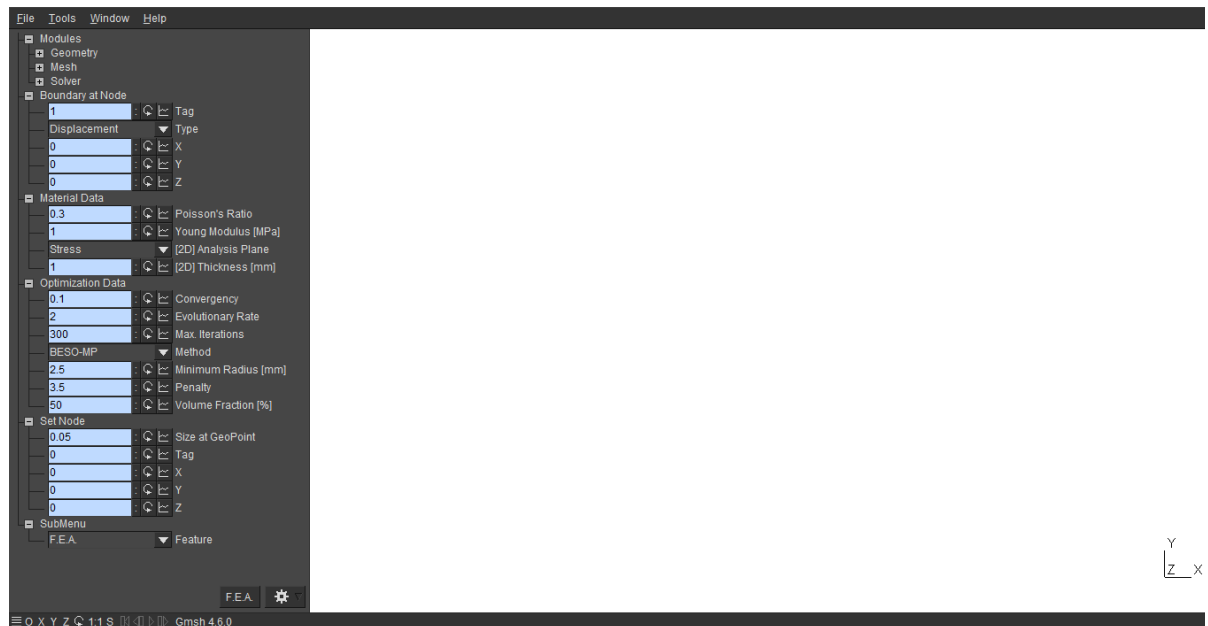


Figura 2: Interface gráfica modificada.

O domínio computacional pode ser importado através de diversos formatos, como .step ou .igs, ou podem ser criados dentro da própria interface gráfica utilizando as funções embutidas no pacote Gmsh.

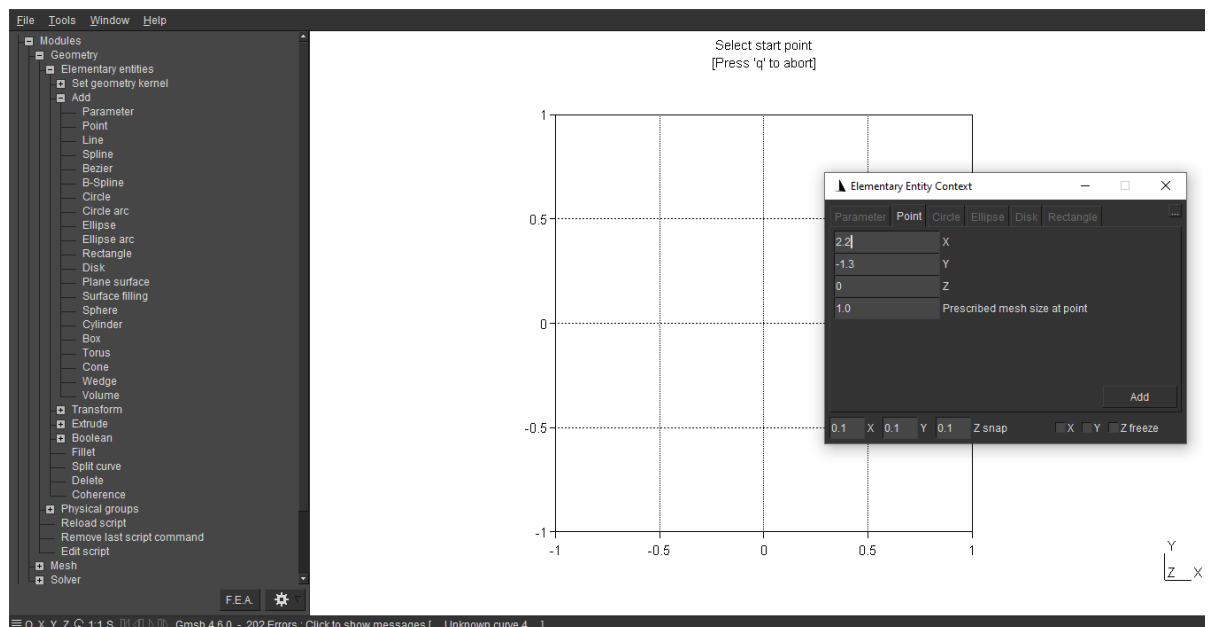


Figura 3: Funções embutidas para criação de geometria dentro da interface gráfica.

A inclusão das condições de contorno deve ser realizada antes da geração da malha, utilizando as funções de grupos físicos do pacote Gmsh, pois com estas funções é possível

obter todos os nós pertencentes à uma determinada região, e com isto as condições podem ser facilmente atribuídas, conforme a Figura 4. Para aplicar as condições de contorno deve-se:

- 1) Escolher o tipo de grupo físico (ponto, curva, face ou corpo);
- 2) Selecionar os elementos físicos da geometria que pertencem ao mesmo tipo de grupo físico;
- 3) Agora deve ser decidido o tipo da condição de contorno:
 - a. Para uma condição de deslocamento, o nome do grupo físico deve ser *displacement_x_a_y_b_z_c*, onde a é o valor correspondente à componente no eixo x, b é o valor correspondente à componente no eixo y, e c é o valor correspondente à componente no eixo z, no caso 2D o valor de c não possui nenhum efeito nos cálculos;
 - b. Para uma condição de força, o nome do grupo físico deve ser *force_x_a_y_b_z_c*, onde a, b e c são os valores correspondentes às componentes nos eixos x, y, e z, respectivamente;
 - c. Para informar uma condição de que os elementos em contato com os nós do grupo físico não podem ser removidos, o nome do grupo de ser *fixel_xx*, onde xx pode possuir qualquer valor alfanumérico para organizar os grupos de elementos fixo.

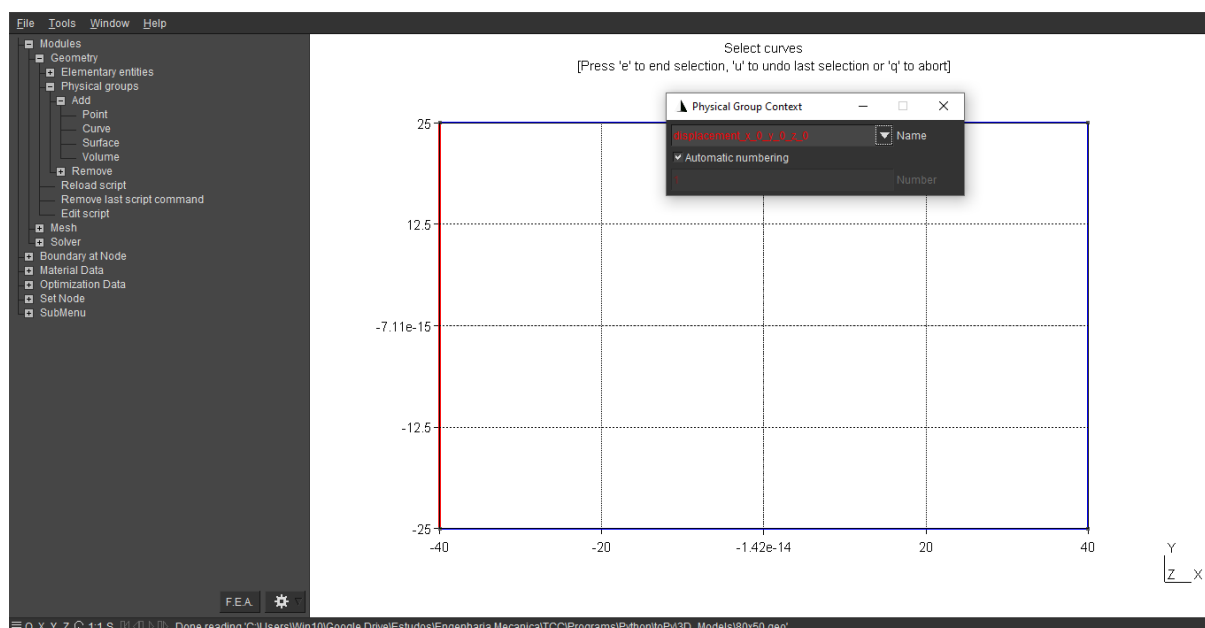


Figura 4: Aplicação das Condições de Contorno.

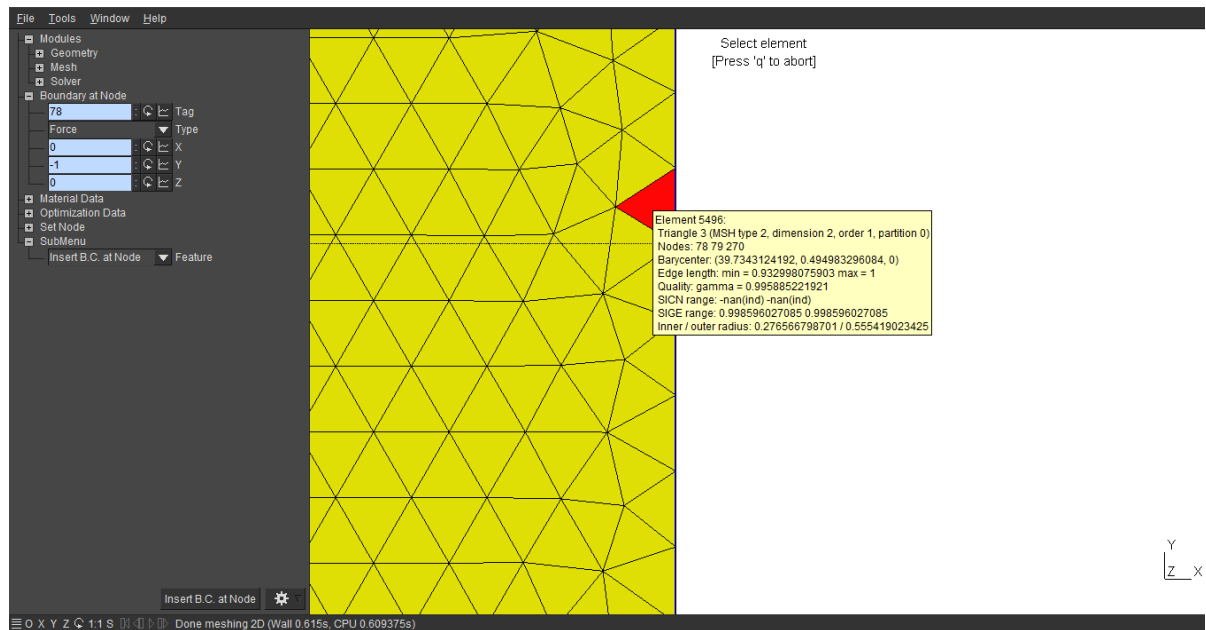


Figura 5: Aplicação de Condição de Contorno em um nó específico.

Caso seja necessário realizar a inclusão de uma condição de contorno em um nó específico, pode ser utilizada a função inspect, para descobrir qual o índice do nó, e em seguida ser colocadas as condições desejadas no campo modificado, demonstrado na Figura

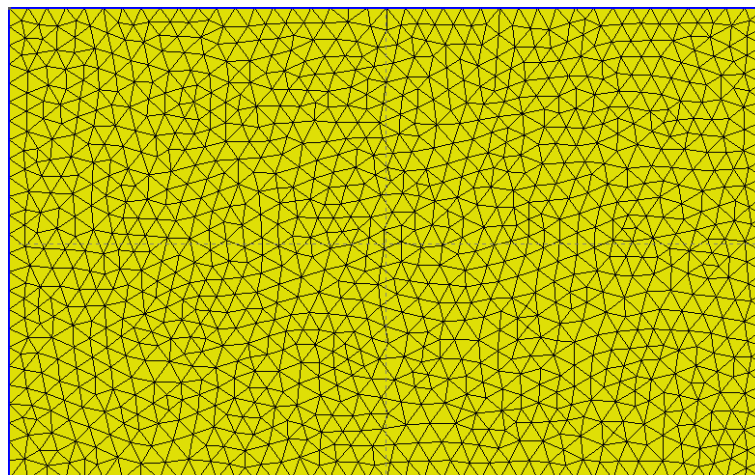


Figura 6: Exemplo de um domínio computacional discretizado não-estruturado em elementos triangulares.

A discretização do domínio computacional nos códigos de Otimização Topológica geralmente utilizam elementos estruturados quadrangulares ou hexaédricos. Neste programa a otimização topológica foi implementada em elementos triangulares, afim de estudar o comportamento dos algoritmos com elementos triangulares e também quando não há uma

malha estruturada. Conforme X. Huang, quando se utiliza um malha não-uniforme podemos utilizar um elemento de densidade de deformação, e_i , ao invés do número de sensibilidade:

$$\alpha_i = e_i = \left(\frac{1}{2} (x_i)^{p-1} u_i^T k_i u_i \right) / V_i \quad (4.1)$$

Onde V_i é o volume do elemento.

5. Resultados

Os resultados apresentados a seguir foram obtidos em casos em que ambos os métodos convergiram e que não houve a necessidade de um grande número de iterações. Vale ressaltar que o método SIMP pode aparentar possuir menos volume, porém isto ocorre porque ao fim da otimização alguns elementos tiveram um valor da densidade relativa intermediário entre 0.001 e 1, e no programa desenvolvido apenas elementos sólidos, ou seja, com a densidade relativa igual a 1 são apresentados.

5.1. MBB Beam:

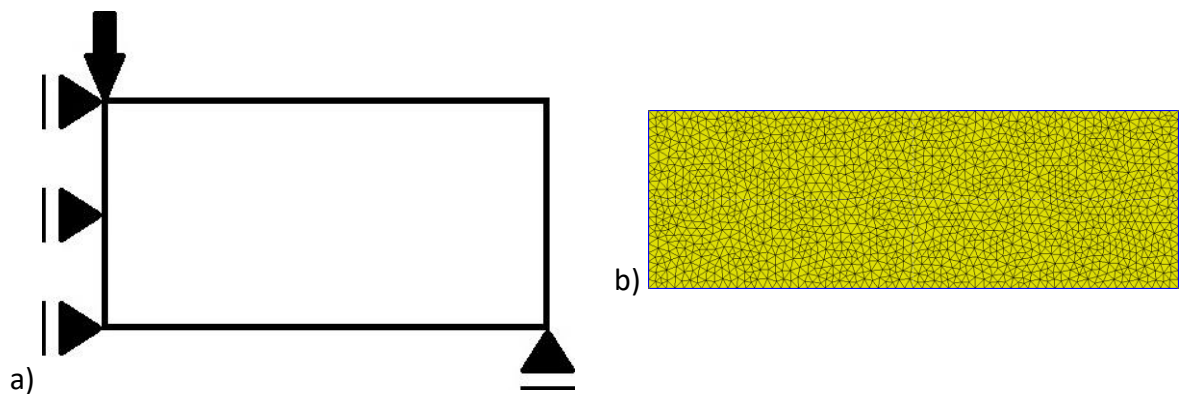


Figura 8 a) Condições de contorno para uma estrutura do tipo MBB Beam. b) Discretização do domínio com 3086 elementos.

Os valores obtidos da função custo, foram 85.58[N.mm] e 180.12[N.mm], para o método BESO e o método SIMP, respectivamente, utilizando uma superfície de 60[mm] x 20[mm] para a importação do domínio, para a geração da malha foi utilizado os valores de 0[mm] para o tamanho mínimo do elemento e 1[mm] para o tamanho máximo. A força

aplicada foi de 1N, e os parâmetros para a otimização sendo, a penalidade de 3.5, um raio mínimo de 1.5[mm], uma fração de volume de 55%, o critério de convergência de 0.1% e 1%, para o método BESO-MP e o método SIMP, respectivamente, e para o caso do método BESO uma proporção de evolução de 2%. O algoritmo utilizado para geração da malha foi o Delaunay.

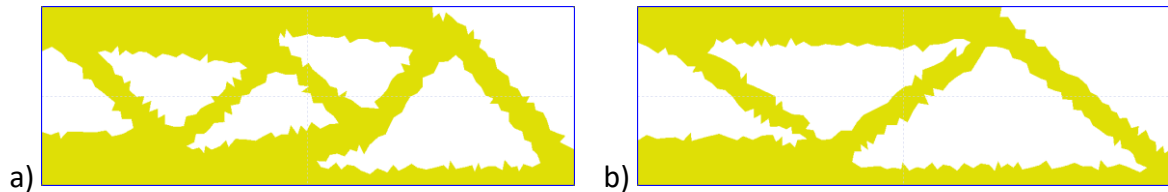


Figura 9: Geometrias obtidas após a otimização topológica. a) Método BESO. b) Método SIMP.

5.2. Formato “L”:

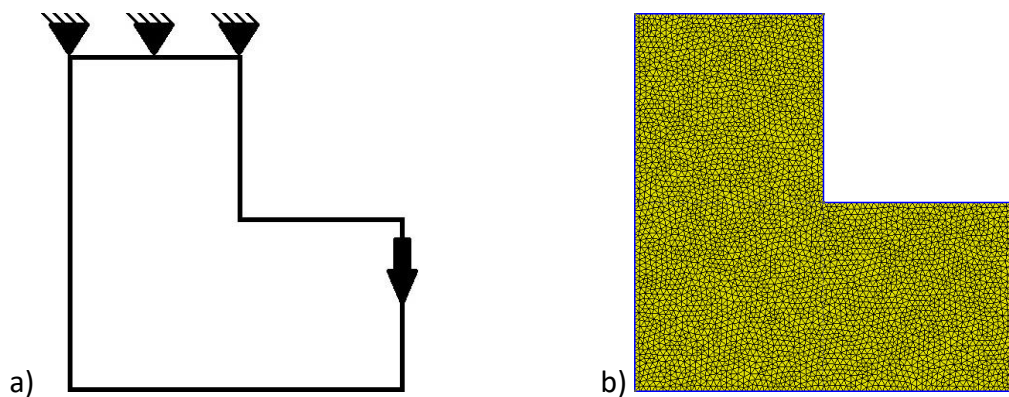


Figura 10: a) Condições de contorno para uma estrutura “L”. b) Discretização do domínio com 8954 elementos.

Os valores obtidos da função custo, foram 32.92[N.mm] e 67.37[N.mm], para o método BESO e o método SIMP, respectivamente, utilizando uma superfície do tipo L com dimensões máximas de 100[mm] x 100[mm] para a importação do domínio, para a geração da malha foi utilizado os valores de 0[mm] para o tamanho mínimo do elemento e 1.5[mm] para o tamanho máximo. A força aplicada foi de 1N, e os parâmetros para a otimização sendo, a penalidade de 3.5, um raio mínimo de 2[mm], uma fração de volume de 55%, o critério de convergência de 0.1% e 1%, para o método BESO-MP e o método SIMP, respectivamente, e para o caso do método BESO uma proporção de evolução de 2%. O algoritmo utilizado para geração da malha foi o Delaunay.

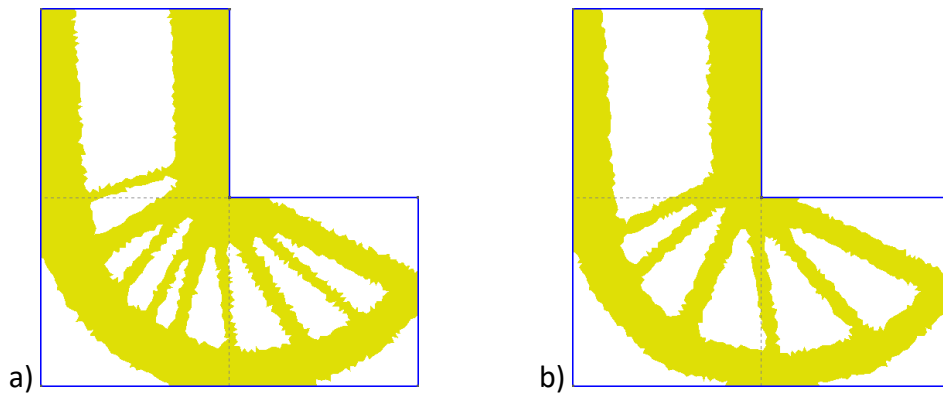


Figura 11: Geometrias obtidas após a otimização topológica. a) Método BESO. b) Método SIMP.

5.3. Viga Engastada:

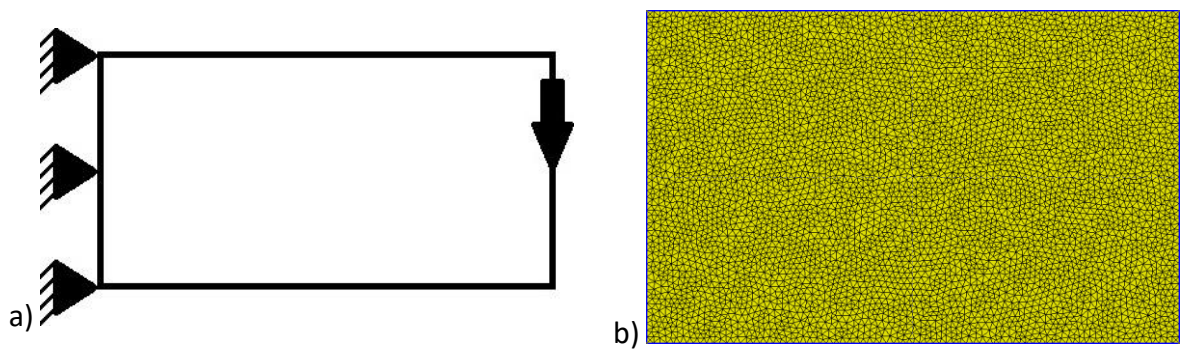


Figura 12: a) Condições de contorno para uma viga engastada. b) Discretização do domínio com 10518 elementos

Os valores obtidos da função custo, foram 18.29[N.mm] e 37.87[N.mm], para o método BESO e o método SIMP, respectivamente, utilizando uma superfície de 80[mm] x 50[mm] para a importação do domínio, para a geração da malha foi utilizado os valores de 0[mm] para o tamanho mínimo do elemento e 1[mm] para o tamanho máximo. A força aplicada foi de 1N, e os parâmetros para a otimização sendo, a penalidade de 3.5, um raio mínimo de 2[mm], uma fração de volume de 50%, o critério de convergência de 0.1% e 1%, para o método BESO-MP e o método SIMP, respectivamente, e para o caso do método BESO uma proporção de evolução de 2%. O algoritmo utilizado para geração da malha foi o Delaunay.

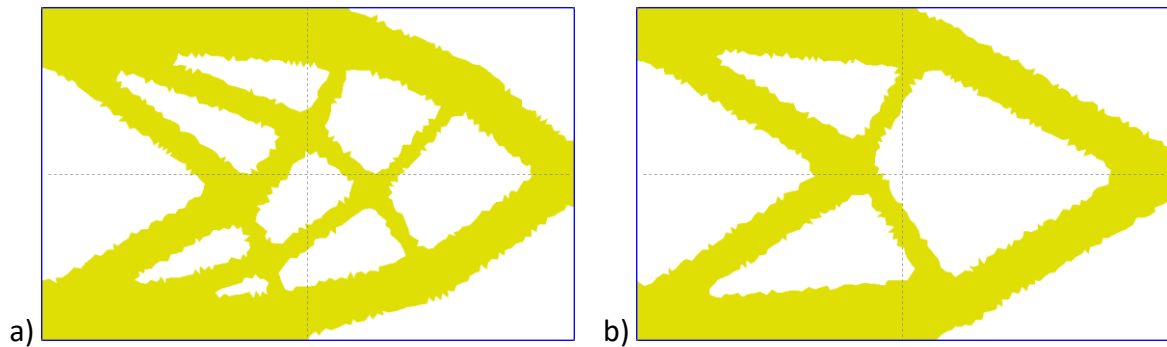


Figura 13: Geometrias obtidas após a otimização topológica. a) Método BESO. b) Método SIMP.

6. Conclusão

A implementação dos métodos de Otimização Topológica e dos Elementos Finitos em um programa similar aos programas do tipo CAE comerciais facilita o entendimento dos conceitos apresentados, e também é possível verificar como uma Black Box pode ser construída.

O uso da Otimização Topológica no software ampliou as possibilidades de realizar testes com condições e domínios diferentes sem a necessidade de realizar modificações no algoritmo do programa.

Foi possível notar como ambos os métodos, SIMP e BESO para materiais com penalização, podem gerar layouts com detalhes similares, mesmo com uma malha não-estruturada de elementos triangulares, e os valores das funções custos obtidos foram satisfatórios.

7. Bibliografia

Bendsøe, M.P.; Kikuchi, N. 1988: Generating optimal topologies in optimal design using a homogenization method. *Comp. Meth. Appl. Mech. Engrg.* 71, 197–224

Bendsøe M. P., Sigmund O. (2003) *Topology Optimization: Theory, Methods and Applications*. Springer, Berlin

Chapra S., C. Raymond. (2008) *Métodos Numéricos para Engenharia – Quinta Edição*. AMGH Editora Ltda. São Paulo

Fernandes M. F. E. (2008) *Optimização topológica multi-objectivo de estruturas lineares elásticas tridimensionais*. Universidade de Aveiro, Departamento de Engenharia Mecânica

Kattan P. (2008) *MATLAB Guide to Finite Elements: An Interactive Approach – Second Edition*. Springer, Berlin

Shure, L. (2007) *Creating Sparse Finite-Element Matrices in MATLAB*. Disponível em <<https://blogs.mathworks.com/loren/2007/03/01/creating-sparse-finite-element-matrices-in-matlab/>>. Acesso em 20 de jun. de 2020

Sigmund O. (2001) A 99line topology optimization code written in Matlab. *Struct Multidisc Optim* 21, 120-127. DOI 10.1007/s001580050176

Sigmund O., Maute K. (2013) Topology optimization approaches: A comparative review. *Struct Multidisc Optim* 48:1031–1055. DOI 10.1007/s00158-013-0978-6

Silva E. C. N. (2001) *Técnicas de Otimização Aplicadas no Projeto de Peças Mecânicas*. Escola Politécnica da USP, São Paulo

Vu Truong Vu (2018) *Bidirectional Evolutionary Structural Optimization with Stationary Steps*. Conference: 2018 Multidisciplinary Analysis and Optimization Conference. DOI 10.2514/6.2018-4059

X. Huang, Y. M. Xie (2010) *Evolutionary Topology Optimization of Continuum Structures*. Wiley, Chichester