

FIA 2020/22

XII CONGRESSO/CONGRESO IBEROAMERICANO DE ACÚSTICA
XXIX ENCONTRO DA SOCIEDADE BRASILEIRA DE ACÚSTICA - SOBRAC

Florianópolis, SC, Brasil

TOPDyn: um código em Python para suporte ao aprendizado sobre otimização topológica em estruturas vibrantes

Rocha, A. P.¹; Silva, O. M.²

¹ Automation and Systems Engineering, Universidade Federal de Santa Catarina, Florianópolis, SC, Brazil

² Laboratório de Vibrações e Acústica, Universidade Federal de Santa Catarina, Florianópolis, SC, Brazil

Resumo

A otimização topológica estrutural pode ser aplicada no projeto de produtos de maneira a reduzir custos de fabricação e alcançar as características vibroacústicas desejadas. O objetivo deste trabalho é apresentar uma ferramenta em código aberto desenvolvida em Python chamada TOPDyn, voltada a usuários especialistas ou não em programação. O procedimento de otimização topológica é aplicado em estruturas bidimensionais, modeladas sob a hipótese de estado plano de tensões, submetidas a carregamentos harmônicos. Apesar de diversos códigos abertos voltados à otimização topológica, poucos ou nenhum são encontrados com foco na análise de vibrações estruturais. No TOPDyn, o Método dos Elementos Finitos é implementado para a solução do problema estrutural enquanto o processo de otimização tem base no Método das Assíntotas Móveis. O código permite a visualização dos leiautes ótimos, das malhas deformadas, de respostas em frequência, e dos gráficos de convergência das funções de desempenho. Há a possibilidade de minimização de grandezas como flexibilidade dinâmica, potência de entrada, energia potencial, energia cinética, entre outras grandezas energéticas comumente utilizadas em análises vibroacústicas. Este trabalho apresenta, como exemplo, a otimização topológica de uma viga engastada clássica para duas frequências de excitação diferentes, visando minimizar as grandezas acima descritas considerando restrição de volume. Desta forma, a usabilidade e as funcionalidades do TOPDyn são demonstradas: definição da geometria, condições de contorno, propriedades e parâmetros de otimização através de uma interface gráfica de fácil uso. Resultados encontrados em artigos recentes deste ramo do conhecimento são utilizados como referência para validação dos resultados obtidos com o pacote.

Palavras-chave: otimização topológica, vibrações harmônicas, SIMP.

PACS: —.

TOPDyn: open-source Python package for learning topology optimization of vibrating structures considering time-harmonic loads

Abstract

Topology optimization of structures can be used in the design process of products to reduce costs and to achieve desired vibroacoustic characteristics. The purpose of this paper is to present an open-source tool developed in Python called TOPDyn that targets specialist and nonspecialist users. The topology optimization is applied to two-dimensional structures subjected to time-harmonic loads considering plane stress state. Even though in the literature there are some open-source tools for learning this method, none of them have a focus on the analysis of structural vibrations. In TOPDyn, the Finite Element Method is implemented to solve the structural problem, while the optimization process is performed through the Method of Moving Asymptotes. It is also possible to visualize optimized layouts, deformed meshes, frequency responses, and convergence graphs of the objective and constraint functions. The package is capable of minimizing vibration quantities like dynamic compliance, input power, kinetic energy, potential energy, and other energetic relations that are commonly used in vibroacoustic analysis. This work presents the optimization of a classical clamped beam for two different excitation frequencies, aiming to minimize the physical functions listed above considering volume constraint. By this way, the usability and the functionality of TOPDyn are demonstrated to define the problem settings, such as properties and boundary conditions in an easy-to-understand graphical interface. This work also validates the obtained results through comparisons with recently published articles in this field of knowledge.

Keywords: topology optimization, harmonic vibration, SIMP.



1. INTRODUÇÃO

O problema de otimização topológica é utilizado para caracterizar e determinar a distribuição ótima de material no espaço de projeto [1]. A otimização estrutural pode ser dividida em três tipos: tamanho, forma e topologia. Ao contrário dos métodos de otimização de tamanho e forma, o domínio inicial da otimização topológica é uma estrutura homogênea, como uma placa retangular em problemas bidimensionais, e as melhorias na estrutura não são limitadas.

O artigo apresenta o desenvolvimento de uma ferramenta, denominada TOPDyn (<https://github.com/TOPDyn/TOPDyn>), de otimização topológica aplicada a problemas de vibrações, na linguagem Python. Na ferramenta, é implementado o Método dos Elementos Finitos [2] para a solução de tal problema, enquanto o método de otimização empregado é o Método das Assíntotas Móveis [3].

As hipóteses práticas estabelecidas para o desenvolvimento da ferramenta foram: o projeto de componentes do ponto de vista vibroacústico pode ser melhorado através de procedimentos de otimização; a otimização topológica permite a obtenção de formas livres para estruturas dependendo de sua função em determinado projeto; e a implementação de procedimentos *open-source* possibilita uma ampla difusão dos procedimentos de otimização, garantindo que possa ser utilizado por pessoas que não possuam acesso a softwares comerciais.

O objetivo principal deste trabalho é propor, implementar e testar uma ferramenta de otimização topológica que lide com problemas de vibração harmônica no tempo, no qual dependendo da frequência de excitação, as estruturas podem se apresentar mais ou menos rígidas. O projeto foi dividido em três etapas principais: implementação do Método dos Elementos Finitos, da otimização topológica e da interface gráfica.

2. FUNDAMENTAÇÃO TEÓRICA

O Método dos Elementos Finitos (MEF) é uma abordagem na qual as equações diferenciais parciais que descrevem os fenômenos físicos podem ser resolvidas numericamente [2]. A ideia é discretizar a estrutura contínua em elementos (ele-

mentos finitos), onde cada elemento é constituído por um certo número de nós. O deslocamento de um ponto arbitrário dentro do elemento é aproximado ao dos nós por meio de funções predeterminadas, chamadas de funções de forma [4].

Segundo [2], o MEF consiste nas seguintes etapas: 1) Pré-processamento: Subdividir o domínio do problema em elementos finitos; 2) Formulação dos elementos: Desenvolvimento das equações para os elementos; 3) Montagem: Obtenção das equações de todo o sistema a partir das equações dos elementos individuais; 4) Solução: Resolver as equações; 5) Pós-Processamento: Determinação da quantidade de interesse, como deformações, e visualização da resposta.

A Figura 1 representa a etapa 1 do MEF e mostra um exemplo de discretização de uma placa bidimensional, na qual é possível notar que os nós conectam os elementos entre si. A etapa 2, formulação dos elementos, é realizada encontrando as matrizes elementares de massa e rigidez do sistema em estudo. Na montagem das equações do problema, etapa 3, são encontradas as matrizes de massa e rigidez globais, sendo escritas, segundo [5], em função das matrizes elementares, calculadas em suas coordenadas locais, de forma que

$$\mathbf{K} = U_e^T \mathbf{K}_e, \quad (1)$$

$$\mathbf{M} = U_e^T \mathbf{M}_e, \quad (2)$$

bem como o vetor de forças

$$\mathbf{F} = U_e^T \mathbf{F}_e \text{ para os elementos de fronteira.} \quad (3)$$

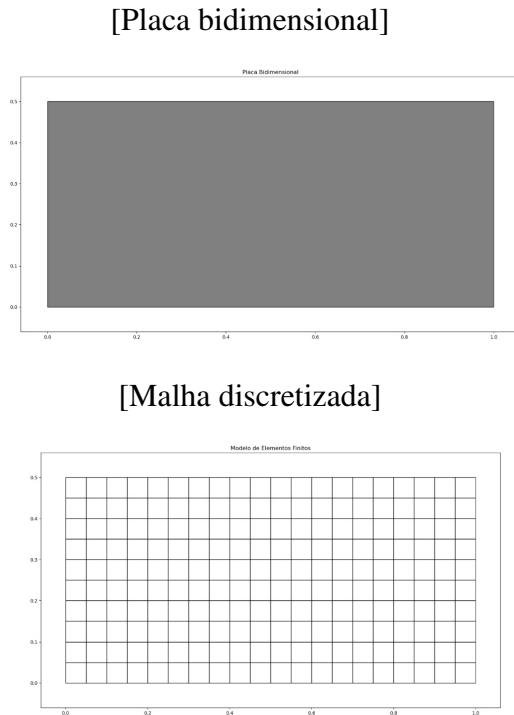
Nas equações 1, 2 e 3, segundo [5], a operação U_e^T representa uma adição matricial restrita aos graus de liberdade aos quais cada elemento e está conectado; $\mathbf{K}$ é matriz de rigidez global, $\mathbf{K}_e$ é a matriz de rigidez elementar, $\mathbf{M}$ é a matriz de massa global, $\mathbf{M}_e$ é a matriz de massa elementar; $\mathbf{F}$ é o vetor de carregamento e $\mathbf{F}_e$ é o carregamento de cada elemento. Ainda de acordo com [5], supondo que o sistema é excitado por um carregamento externo harmônico e para uma dada frequência de excitação ω , é possível representar o sistema de equações a ser resolvido na etapa 4,

como

$$(\mathbf{K} + i\omega\mathbf{C} - \omega^2\mathbf{M})\mathbf{U}(\omega) = \mathbf{F}, \quad (4)$$

sendo $\mathbf{C}$ a matriz de amortecimento global e $\mathbf{U}(\omega)$ o vetor de deslocamento calculado para a frequência ω . A etapa de pós-processamento consiste na visualização da malha deformada com as condições de contorno, sendo $\mathbf{U}(\omega)$ a variável de interesse.

Figura 1: Malha de elementos finitos.



Já com a Otimização Topológica é possível encontrar a distribuição mais adequada de material em uma estrutura a partir da definição de uma região inicial e do conhecimento de informações como carregamento aplicado, restrições de deslocamento e se a estrutura apresenta furos ou áreas sólidas. Essa estrutura é otimizada para uma determinada função solicitada, sendo que essa função pode ser única ou uma soma de funções (função multiobjetivo). A Figura 2 apresenta o resultado da otimização topológica com função objetivo Potência de entrada de uma estrutura retangular. O objetivo do projeto de otimização topológica, segundo [6], é determinar quais pontos do espaço devem ser pontos materiais e quais pontos devem permanecer vazios. Pode-se associar a representação geométrica da estrutura otimizada como

semelhante a uma renderização em preto e branco de uma imagem.

Neste trabalho, o método utilizado na otimização é o Solid Isotropic Material with Penalization (SIMP) que, segundo [7], está associado a um parâmetro espacial que define as propriedades de um material artificial ao longo de um domínio fixo representado por uma malha de elementos, considerando o uso do Método dos Elementos Finitos para resolver o problema físico. O método SIMP, de acordo com [8], utiliza um fator de penalização (PF) para evitar a formação de regiões com densidades intermediárias. Assim, o fator de penalização orienta a solução ótima para conter elementos sólidos ou vazios com densidades 1 ou 0, sendo chamado de vetor de densidades relativas o vetor contendo as densidades dos elementos. Isso é importante, visto que elementos com densidades intermediárias não seriam factíveis para a fabricação ou impressão da estrutura.

Ainda segundo [7], o número de variáveis nesse procedimento de otimização é grande, igual ao número de elementos finitos na malha do domínio. Assim a aplicação de programação convexa sequencial é de interesse, como o Método das Assíntotas Móveis (MMA) e sua versão mais conservadora, o Método Globalmente Convergente de Assíntotas Móveis (GCMMA). Neste trabalho, ambos os métodos foram retirados do código Python *open-source* disponível em [9] e que foram baseados no código GCMMA-MMA escrito em *MATLAB* por [3], sendo o trabalho original retirado de [10] sob o *General Public License* (GNU).

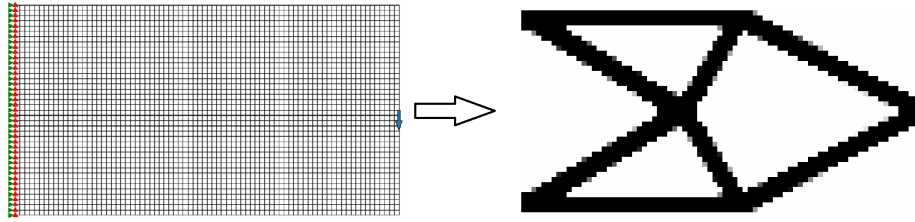
Segundo [3], são considerados problemas sequenciais de otimização na forma mostrada na Equação 5, na qual as variáveis são $x = (x_1, \dots, x_n)^T \in \mathbb{R}^n$, $y = (y_1, \dots, y_m)^T \in \mathbb{R}^m$, e $z \in \mathbb{R}$.

$$\begin{aligned} &\text{minimize } f_0(x) + a_0 z + \sum_{i=1}^m (c_i y_i + \frac{1}{2} d_i y_i^2) \\ &\text{subject to } f_i(x) - a_i z - y_i \leq 0, i = 1, \dots, m \\ &\quad x \in X, y \geq 0, z \geq 0, \end{aligned} \quad (5)$$

Ainda na Equação 5, $X = \{x \in \mathbb{R}^n \mid x_j^{\min} \leq x_j \leq x_j^{\max}, j = 1, \dots, n\}$, sendo $x_j^{\min}$ e $x_j^{\max}$ números reais que satisfazem $x_j^{\min} < x_j^{\max}$ para todo j ; $f_0, f_1, \dots, f_m$ são funções contínuas diferenciáveis de valor real em X ; a_0, a_i, c_i e d_i são números reais que satisfazem



Figura 2: Estrutura otimizada com função objetivo Potência de entrada.



$a_0 > 0$, $a_i \geq 0$, $c_i \geq 0$, $d_i \geq 0$ e $c_i + d_i > 0$ para todo i , e também $a_i c_i > a_0$ para todo i com $a_i > 0$. Para a otimização foram adotados três critérios de convergência, sendo eles: número máximo de iterações; Condições de Karush Kuhn Tucker; se o valor da função objetivo após 10 iterações permanecer constante durante 5 iterações.

2.1 Funções objetivo e análise de sensibilidade

Para a aplicação do MMA e GCMMA na otimização do vetor de densidades relativas é necessário o cálculo das funções de restrição e das funções objetivo envolvidas, além dos seus respectivos gradientes. A função objetivo em um problema de otimização topológica é utilizada para definir o objetivo da otimização, sendo um valor escalar que é estabelecido de acordo com as definições do problema. Na ferramenta TOPDyn foram empregadas 8 funções objetivo, sendo que elas podem ser combinadas em pares para minimizar ou maximizar ambas. Assim, é proposta a função objetivo G da Equação 6:

$$G = n \cdot G_1 + (1 - |n|) \cdot G_2, \quad (6)$$

na qual G_1 e G_2 são as funções objetivo, e n é o fator de ponderação, sendo o seu valor padrão normalmente 1. Ao alterar o valor de ponderação é possível equilibrar o efeito de uma função objetivo que está dominando a otimização. Ao minimizar a função objetivo o propósito é obter estruturas com baixos níveis de vibração. Enquanto que, ao maximizar uma função objetivo, pretende-se criar estruturas ressonantes para uso, por exemplo, em *Energy Harvesting*. A formulação das funções objetivo utilizadas na ferramenta TOPDyn foi baseada nos trabalhos [7], [11], [12], sendo 3 funções objetivo locais e 5 globais. As funções globais são: Flexibilidade; Potência de entrada; Energia potencial elástica; Energia ci-

nética; Razão de energias. Enquanto as locais são: Razão local de energias; Energia potencial elástica local; Energia cinética local.

As possíveis funções de restrição são: Razão de energias, Flexibilidade, Energia cinética local, energia potencial elástica local, razão local de energias e com o acréscimo da função de restrição de área. Com as funções de restrição é possível impedir que a otimização retorne soluções triviais. Por exemplo, para maximizar a rigidez da estrutura, a otimização retornaria uma peça com todas as densidades relativas iguais a 1. Contudo, ao aplicar uma restrição, a otimização é forçada a buscar outra solução que satisfaça a maximização da função objetivo e a restrição.

O procedimento de cálculo do gradiente das funções objetivo e de restrição envolvidas em relação às variáveis de projeto, segundo [5], é conhecido em procedimentos de otimização como análise de sensibilidade. Ainda segundo [5], as funções objetivo e a análise de sensibilidade necessitam de informações obtidas a partir da solução da equação de equilíbrio, que no problema estrutural resulta no campo de deslocamento $\mathbf{U}$, e de informações intrínsecas da estrutura discretizada, como a matriz de rigidez e o volume.

2.2 Filtros

Para evitar que a estrutura otimizada apresente soluções não desejadas com descontinuidades entre elementos, um filtro é aplicado no vetor de densidades relativas e consequentemente nas derivadas das grandezas vibratórias do problema. Neste trabalho, dois filtros podem ser utilizados: filtro de densidade e de sensibilidade. O filtro de densidade atua diretamente na densidade, sendo definido, de acordo com [7], [11], [12], pela Equação

ção 7:

$$\frac{df}{dx_e} = \sum_{i=1}^{N_e} \left(\frac{1}{\sum_{j=1}^{N_i} H_j} \cdot H_i \cdot \frac{df}{dx_i} \right), \quad (7)$$

na qual N_e é o número de elementos na vizinhança do elemento e dentro de um raio R ; N_i são os elementos i vizinhos do elemento e , e $H_i = R - \text{dist}(e, i)$. Sendo $\text{dist}(e, i)$ a distância entre os centroides dos elementos e e i . Enquanto o filtro de sensibilidade, ainda segundo [7], [11], [12], é definido pela Equação 8:

$$\frac{df}{dx_e} = \frac{1}{x_e \cdot \sum_{i=1}^{N_r} H_i} \cdot \sum_{i=1}^{N_r} (H_i \cdot x_i \cdot \frac{df}{dx_i}), \quad (8)$$

na qual N_r é o número de elementos dentro do raio R , e $H_i = R - \text{dist}(e, i)$.

2.3 Elementos passivos

Em alguns casos, segundo [6], alguns elementos podem ser obrigados a assumir um valor mínimo ou máximo de densidade, por exemplo, um furo para um tubo ou uma região que suporta um objeto. Quando se é definido um valor mínimo, os elementos no vetor de densidades relativas são definidos como 1 e no vetor de densidades relativas mínimas são fixados como 0.99. A Figura 3(a) apresenta uma estrutura de tamanho $0.5 \text{ m} \times 1 \text{ m}$, na qual os elementos na região $0 \text{ m} \leq X \leq 0.5 \text{ m} \cap 0.95 \text{ m} \leq Y \leq 1 \text{ m}$ não alteram suas densidades relativas. Enquanto, quando se é definido um valor máximo, os elementos no vetor de densidades relativas são definidos como 0 e no vetor de densidades relativas mínimas são fixados como 0.0001. A Figura 3(b) apresenta uma estrutura de tamanho $1 \text{ m} \times 0.5 \text{ m}$, na qual os elementos na região $0.8 \text{ m} \leq X \leq 1 \text{ m} \cap 0.30 \text{ m} \leq Y \leq 0.5 \text{ m}$ não alteram suas densidades relativas.

3. PROGRAMAÇÃO

3.1 Python

De acordo com [13], Python é uma linguagem de programação de uso geral e definida como uma linguagem de *script* orientada a objetos. Também conforme o site oficial da linguagem [14] ressalta-se que Python é desenvolvida sob uma licença de código aberto aprovada pela *Open System Interconnection* (OSI) e administrada pela *Python Software Foundation*, tornando-a livremente uti-

Figura 3: Estrutura otimizada com elementos passivos.

[Elementos fixos como 1 no vetor de densidades relativas]



[Elementos fixos como 0 no vetor de densidades relativas]



lizável e distribuível, mesmo para uso comercial. Assim, a linguagem possui a contribuição da comunidade para a criação de bibliotecas que aumentam as suas funcionalidades e possibilidades de uso. Com a imersão da comunidade externa na criação de bibliotecas, também é gerada uma coletividade para auxílio em problemas complexos e resolução de dúvidas.

Para o desenvolvimento da ferramenta de otimização topológica, os fatores primordiais para o uso da linguagem de programação Python foram a sua capacidade matemática e diversidade de bibliotecas que poderiam auxiliar no aperfeiçoamento da ferramenta.

A interação do usuário com o *software* é feita com uma interface gráfica do usuário (GUI - *Graphic User Interface*). Foi utilizada a biblioteca *PyQt5* [15] que permite utilizar o *framework* Qt, sendo este escrito em C++. Assim, ao criar as interfa-



ces, a velocidade de execução do código não é comprometida excessivamente.

Após a entrada dos parâmetros e execução do processo selecionado, é realizada a visualização dos gráficos. Foi estabelecido que o gráfico da estrutura otimizada e o gráfico das funções objetivo e de restrição seriam mostrados a cada iteração que a otimização topológica fosse executada, pois a execução do processo é consideravelmente demorada para que o usuário fique sem nenhum tipo de visualização do progresso da otimização. Assim, ao visualizar parte dos dados do processo, é possível acompanhar como as densidades relativas estão sendo alteradas.

Para a visualização da peça deformada utilizou-se a biblioteca *Matplotlib* [16], visto que esta possui personalizações poderosas para a plotagem de gráficos 2D e é possível integrar os gráficos na interface gráfica. Enquanto para a estrutura otimizada foi usada a biblioteca *PyQtGraph* [17], que possui como qualidades o desenho rápido para a exibição de dados em tempo real e exibição de imagens que facilitam a geração do gráfico do vetor de densidades relativas.

3.2 Tratamento dos dados de entrada

O tratamento dos dados de entrada engloba a validação das entradas passadas pelo usuário e a transmissão dos dados da interface para o back-end. Dessa forma, é evitado que o algoritmo "quebre" devido a um valor passado erroneamente. Na validação de cada dado fornecido pelo usuário, se a entrada fornecida não for de um dos tipos obrigatórios, é gerado um aviso no formato texto, que é apresentado como um *pop-up* na interface, informando os parâmetros que estão incorretos e qual o tipo esperado para cada um deles.

A transmissão dos dados, para a execução do algoritmo de otimização, é feita em duas etapas: agrupamentos dos dados e exportação. Para o agrupamento é gerado um dicionário com os dados numéricos, booleanos e do tipo lista do processo selecionado, além de dois vetores bidimensionais do tipo *numpy array* com o carregamento e com os nós com deslocamento restrito. Cada linha desses vetores é um carregamento ou restrição definido pelo usuário, sendo exemplos de vetores os mostrados na Equação 9. O item 1 informa que o deslocamento do nó na posição (1, 0.5) está

restrito nas direções X e Y; e o item 2 informa que é adicionado um carregamento negativo de módulo 100 N no grau de liberdade, na direção Y, de todos os nós que possuem coordenada X=0 ou estejam próximos da coordenada X=0 com uma margem de erro $1e-6$.

$$\begin{cases} 1) [[1, 0.5, 1, 1]] \\ 2) [[0, 1, 1e-6, 0, -1, 100]] \end{cases} \quad (9)$$

Enquanto na transmissão é criada uma pasta temporária que é utilizada para salvar o dicionário e os vetores das condições de contorno contendo os parâmetros do problema. Esses dados são importados nos arquivos de execução do processo. Essa pasta também é utilizada para salvar os dados gerados no decorrer do processo e que serão adicionados na interface em forma de gráfico ou dado bruto. É considerada uma pasta temporária, pois esta é excluída do diretório, assim que o processo for finalizado ou o botão *stop* for pressionado.

3.3 Geração da malha

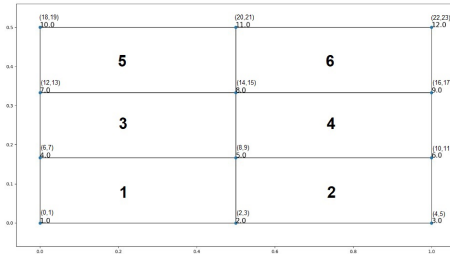
Para a geração da malha é considerado o número de elementos que se deseja na direção X (**nelx**) e na direção Y (**nely**), além do comprimento (**lx**) e largura (**ly**) da estrutura. Baseado nessas informações, é gerado um *grid*, no qual os elementos são quadrados ou retangulares. Assim, são obtidas as matrizes de coordenadas dos nós e de conectividade dos elementos.

No projeto são estudadas placas bidimensionais; os elementos possuem dois graus de liberdade por nó, ou seja, o número mínimo de coordenadas independentes para especificar a posição do nó são duas. A Figura 4 apresenta como os elementos, os nós e seus graus de liberdade são numerados. Além disso, a ordem que os nós são dispostos na matriz de conectividade para a malha da Figura 4 é mostrada na Tabela 1. Essa sequência de numeração foi escolhida para gerar a formulação das matrizes elementares de massa e rigidez.

Tabela 1: Ordem dos nós na matriz de conectividade para a malha da Figura 4.

Elemento	1 ^o nó	2 ^o nó	3 ^o nó	4 ^o nó
1	1	2	5	4
2	2	3	6	5
3	4	5	8	7
4	5	6	9	8
5	7	8	11	10
6	8	9	12	11

Figura 4: Numeração dos nós e dos elementos.



3.4 Definição das condições de contorno

Com a geração da malha é possível definir quais nós recebem o carregamento e quais nós não apresentam restrição no deslocamento. A Figura 5 apresenta o fluxograma da execução da definição das condições de contorno. Para encontrar o(s) nó(s), caso tenham sido fornecidas as coordenadas, é utilizado o algoritmo árvore k-d que encontra os vizinhos mais próximos de qualquer ponto. Logo, caso a coordenada fornecida pelo usuário não exista na matriz de coordenadas, é selecionado um nó que esteja próximo do desejado. Ao final, é retornado o número dos nós que possuem carregamento.

A partir dos nós são encontrados os seus índices dos graus de liberdade. Com essa informação é declarado o vetor de carregamento, sendo formado inicialmente de zeros com tamanho igual ao número de graus de liberdade (NGL), sendo NGL dado pela Equação 10:

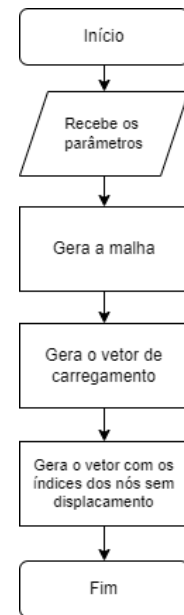
$$NGL = 2 \cdot ((nelx + 1) \cdot (nely + 1)), \quad (10)$$

e adicionado o carregamento nos nós obtidos

pelo algoritmo. Por exemplo, um carregamento positivo de 100 N no grau de liberdade, na direção X, do nó na posição (1, 0.25) na malha da Figura 4 seria dado pelo vetor bidimensional $[[1, 0.25, 1, 0, 100]]$. O nó associado a essa coordenada é o 9 e o seu grau de liberdade é o 17.

O algoritmo árvore k-d também é utilizado para obter os nós com deslocamentos restritos, todavia esses nós são excluídos e é gerado um vetor com os graus de liberdade livres. Por exemplo, o vetor que representa a restrição de todos os graus de liberdade dos nós que possuem coordenada $X = 0$ da malha da Figura 4 é $[[0, 1, 1, 1]]$. A partir dele são obtidos os nós 1, 4, 7, 10 e, com essa informação, o vetor com os graus de liberdade livres é dado por $[2, 3, 4, 5, 8, 9, 10, 11, 14, 15, 16, 17, 20, 21, 22, 23]$.

Figura 5: Fluxograma da definição das condições de contorno.

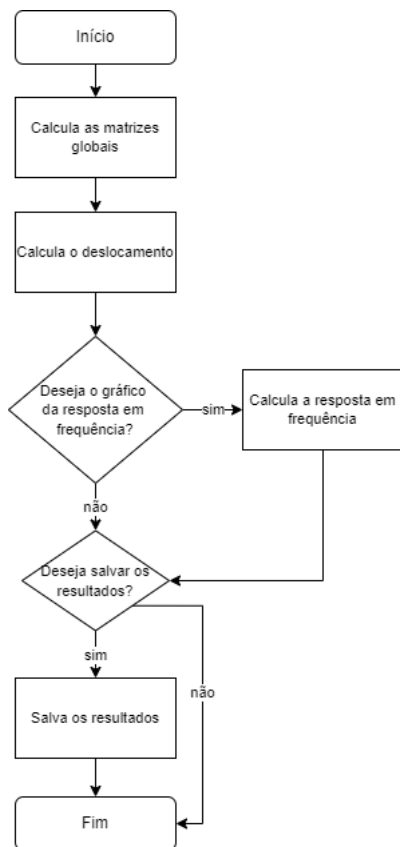


Com a geração da malha e dos vetores das condições de contorno de acordo com o fluxograma da Figura 5 é possível executar o Método dos Elementos Finitos e a Otimização Topológica. Os passos do procedimento MEF são apresentados no fluxograma da Figura 6, no qual para a geração das matrizes globais utilizou-se matrizes esparsas, visto que a maioria dos elementos são preenchidos por zeros. Para calcular o vetor de deslocamentos utilizou-se a solução harmônica direta; e na resposta em frequência um laço de



repetição *for* foi usado para calcular $U(\omega)$ para cada ω contido no vetor de frequências fornecido pelo usuário.

Figura 6: Fluxograma do Método dos Elementos Finitos.



Os passos para a execução do procedimento de otimização topológica são mostrado na Figura 7. O deslocamento pode ser calculado pelo método da superposição modal, que usa as frequências naturais e formas de modo da análise modal, ou por uma análise harmônica obtida pelo método direto. Também é necessário normalizar as funções objetivo e suas derivadas, bem como a função de restrição, visto que, segundo [3], a otimização matemática opera melhor em valores próximos de 100. Assim, o valor da função é dado como:

$$f'(x) = 100 \cdot \frac{f(x)}{f_0(x)}, \quad (11)$$

sendo $f'(x)$ o valor normalizado da função, $f(x)$ o valor atual da função e $f_0(x)$ o valor da função na primeira iteração do processo. Quando se utiliza a função multiobjetivo, a normalização é realizada antes de somar as funções de acordo com

a Equação 6. As derivadas também são normalizadas pelo valor da função objetivo na primeira iteração. Além disso, para diminuir o tempo de processamento, o filtro é aplicado em todas as derivadas (derivadas das funções objetivo e de restrição) ao mesmo tempo, visto que de acordo com as Equações 7 e 8 são realizados dois somatórios no cálculo do filtro. Assim, esses somatórios só são realizados uma vez para todas as derivadas. Ainda em relação às derivadas, primeiro é aplicado o filtro, depois é efetuada a normalização e, por último, caso a função multiobjetivo seja utilizada, as derivadas da função objetivo são somadas.

4. EXEMPLOS DE APLICAÇÃO

Nesta seção, são descritos os testes realizados para verificar o funcionamento da ferramenta desenvolvida e seus resultados. O objetivo é avaliar se os resultados obtidos na ferramenta são condizentes com a literatura. Os testes foram realizados com os métodos MMA e GCMMA, mas os resultados mostrados são apenas do método MMA, devido a semelhança da resposta.

5. MINIMIZAÇÃO DA POTÊNCIA DE ENTRADA COM RESTRIÇÃO DE VOLUME

A análise do resultado da potência de entrada foi baseada no trabalho [5], no qual foi utilizada a estrutura apresentada na Figura 8. A estrutura apresenta comprimento de 1 m e largura de 0.5 m, e uma das extremidades laterais é engastada, sendo a outra submetida a um carregamento harmônico de 10000 N aplicado no nó central da estrutura (coordenada (1, 0.25)). Ainda de acordo com [5], nesta análise será utilizado o modelo SIMP padrão, com $p = 3$ e $q = 1$, também é adotado o filtro de densidades com raio mínimo de 0.031 m, e o vetor de densidades relativas é definido como 0.49 para todos os elementos da malha. O único coeficiente de amortecimento diferente de 0 será o proporcional à rigidez, sendo adotado $Beta = 1e - 6$, o coeficiente de Poisson adotado será de 0.3 e o módulo de elasticidade será 210e9 Pa.

A Figura 9 apresenta a resposta em frequência da Potência de entrada para a faixa de frequências entre 5 e 500 Hz, com discretização de 1 Hz, considerando as configurações definidas anterior-

Figura 7: Fluxograma do processo de otimização topológica.

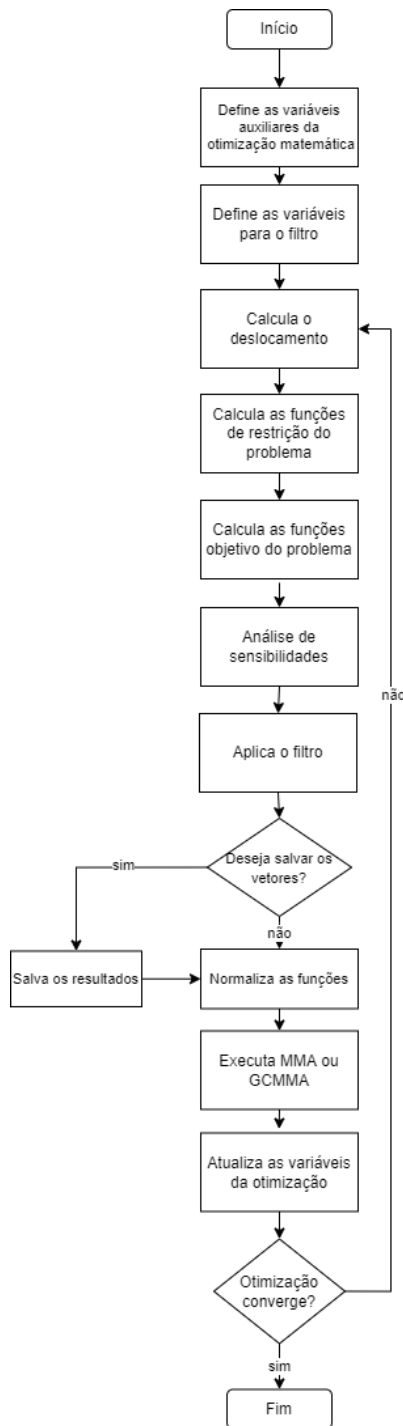
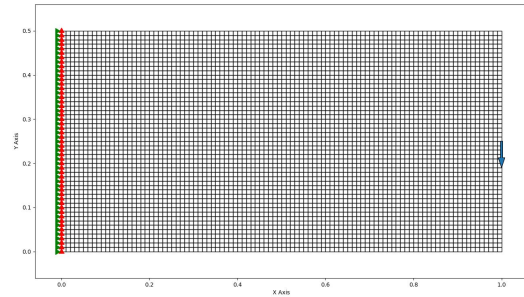


Figura 8: Estrutura exemplo para a análise de Potência de entrada.

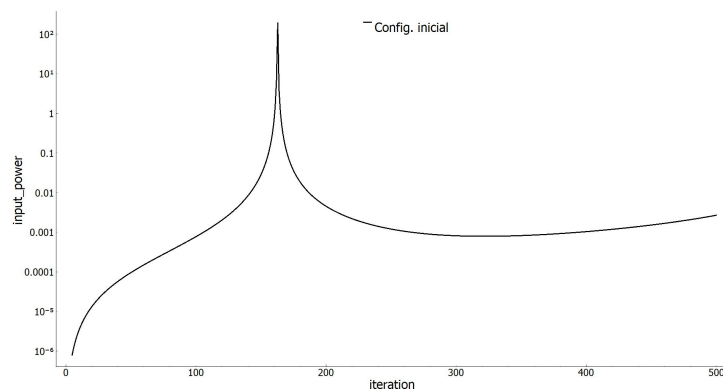


mente e utilizando uma análise harmônica para obtenção do deslocamento. Será desenvolvida uma análise em torno da primeira ressonância, ou seja, em 175 Hz. Para isso, considerou-se as frequências antes e depois da primeira ressonância, sendo 160 Hz e 185 Hz.

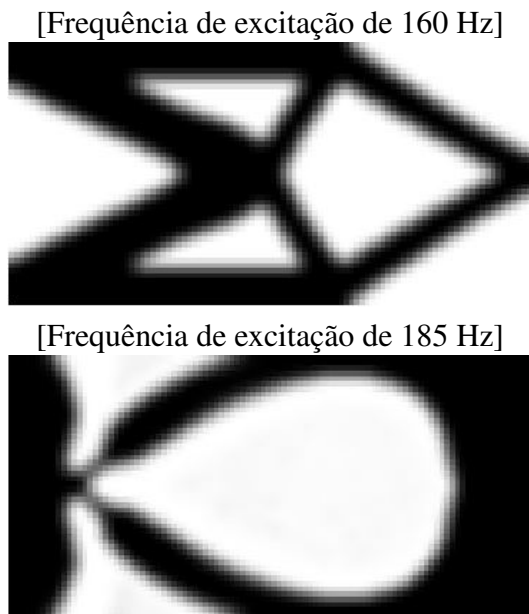
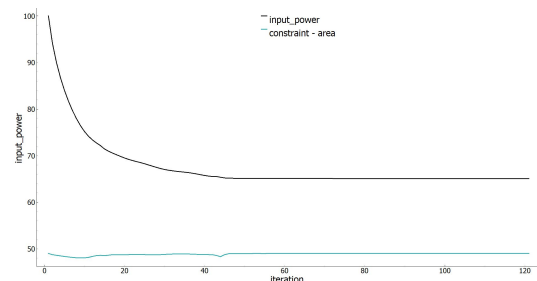
A estrutura otimizada, obtida para a frequência de 160 Hz, é mostrada na Figura 10(a). Como a frequência analisada está à esquerda da ressonância, a otimização busca mover o pico da estrutura otimizada para frequências alcançáveis mais altas, com o objetivo de minimizar a função em 160 Hz. Enquanto a estrutura otimizada para a frequência de excitação em 185 Hz (à direita da ressonância) é mostrada na Figura 10(b), sendo que esta apresenta uma massa grande e uma rigidez pequena. Nesse caso, a otimização busca mover o pico da estrutura otimizada para frequências mais baixas possíveis, assim, é criada uma mola flexível com o objetivo de manter a ressonância em baixas frequências.

Esses comportamentos estão em concordância com [5], que afirma para a estrutura da Figura 10(a) que o local de aplicação do carregamento apresenta material e é consistente em sua forma final; e para a estrutura da Figura 10(b) que o método encontrou uma solução, na qual é gerado próximo do engaste um vínculo muito flexível com a base, propiciando um movimento de rotação da maior parte do corpo.

Os valores da função objetivo, normalizada pelo valor inicial, e da função de restrição para cada iteração da otimização para a frequência de excitação de 160 Hz são mostrados na Figura 11. Observa-se que a função objetivo foi minimizada e a restrição de volume foi obedecida, seus valo-

**Figura 9:** Resposta em frequência da Potência de entrada.

res finais são, respectivamente, 65 Nm/s e 49%. O gráfico para a frequência de excitação de 185 Hz foi omitido devido à similaridade, sendo que o último valor para a função de restrição foi o mesmo para 160 Hz e o valor da função objetivo foi 154 Nm/s. Novamente os resultados estão em conformidade com [5].

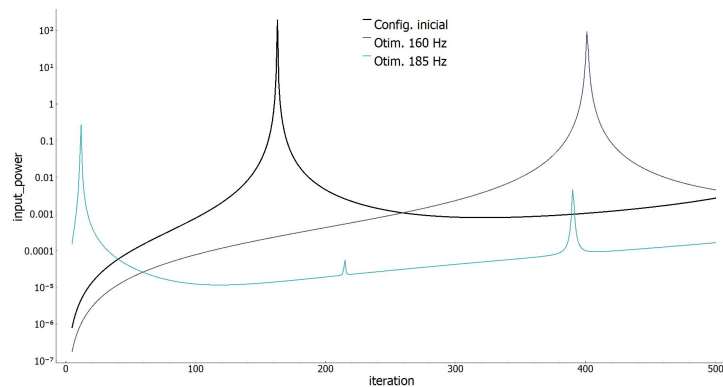
Figura 10: Estrutura otimizada para função objetivo Potência de entrada.**Figura 11:** Valores da função objetivo normalizada pelo valor inicial e da função de restrição para cada iteração da otimização.

Para evidenciar o comportamento da estrutura é mostrada, na Figura 12, a resposta em frequência da Potência de entrada para a configuração original (todos os valores do vetor de densidades relativas igual a 0.49) e para as estruturas otimizadas com frequências de excitação de 165 Hz e 185 Hz.

6. OTIMIZAÇÃO MULTIOBJETIVO COM RESTRIÇÃO DE VOLUME

Dada a estrutura otimizada da Figura 10(b) e segundo [5], para frequências de excitação acima da primeira ressonância da configuração inicial, a minimização da Potência de entrada leva a uma

Figura 12: Resposta em frequência da Potência de entrada para a configuração original e para as estruturas otimizadas.



minimização da energia de deformação, assim o design final tem uma tendência a se desvincular da base engastada, criando um movimento de corpo rígido. Para solucionar esse problema será usada uma função multiobjetivo para a otimização, sendo considerada a Potência de entrada e a Flexibilidade, e o peso associado a cada uma é, respectivamente, 0.9 e 0.1. Dessa forma, segundo [5], ao adicionar uma função de controle do comportamento estático da estrutura, é garantido que a peça não terá grande movimentação e, com isto, é possível realizar otimizações em frequências de excitação mais altas, mantendo uma forma fabricável e com redução de vibrações na frequência de interesse. Como o objetivo é restringir a Flexibilidade estática, será adotada a frequência de excitação de 0 Hz para essa função. A Figura 13 apresenta a estrutura otimizada com a função multiobjetivo. Essa peça possui massa na região de engaste, sendo que quanto menor o peso adotado para a Flexibilidade, menor a região engastada.

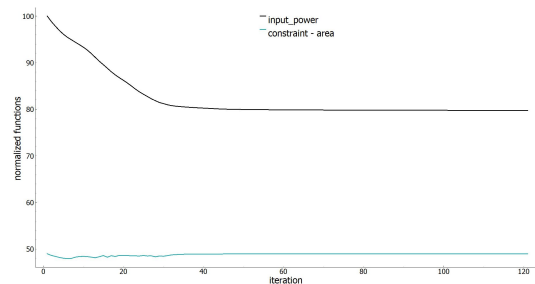
Figura 13: Estrutura otimizada com função multiobjetivo: Potência de entrada e Flexibilidade.



Os valores da função multiobjetivo normalizada

pelo valor inicial e da função de restrição para cada iteração da otimização para a frequência de excitação de 185 Hz são mostrados na Figura 14. Observa-se também que a restrição de volume foi obedecida, sendo finalizada em 49%.

Figura 14: Convergência da função multiobjetivo e da função de restrição.



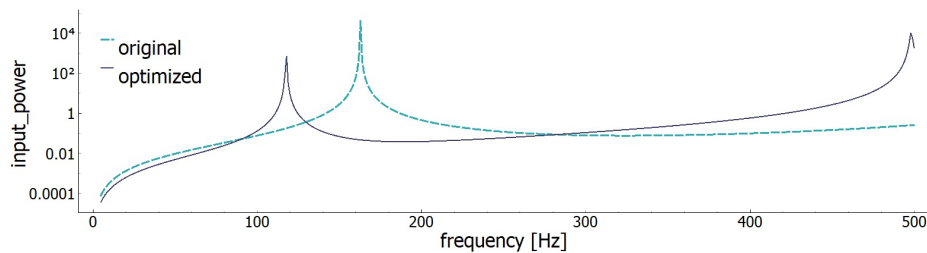
Portanto, de acordo com [5], o uso da restrição de flexibilidade estática resulta em uma estrutura contínua, vinculada à base, e com redução da Potência de entrada. Essa redução pode ser observada pelo gráfico da resposta em frequência da função objetivo Potência de entrada com o vetor de densidades relativas, obtido com a função multiobjetivo, mostrada na Figura 15.

7. CONSIDERAÇÕES FINAIS

O objetivo deste trabalho é a criação de um *software* de fácil entendimento e usabilidade e que fosse uma alternativa para pessoas que não possuem acesso a *softwares* comerciais e estudantes que estão aprendendo Otimização Topológica. Neste contexto, o objetivo foi alcançado de forma satisfatória no desenvolvimento da ferramenta,



Figura 15: Resposta em frequência da Potência de entrada com o vetor de densidades relativas, obtido com a função multiobjetivo.



que permite o usuário fornecer os parâmetros para representar seu problema de maneira metódica (passo a passo), utilização dos métodos matemáticos Método dos Elementos Finitos, MMA e GCMMA, além de uma gama de funções objetivo e de restrição que permitem abranger cenários variados.

REFERÊNCIAS

- [1] Coutinho, Karilany; Júnior, João; Krajnc Alves, Marcelo; Neto, Custódio e Wanderley, Caroline. Otimização topológica 3d sob restrição de tensão. 3, 02 2014. doi: [10.18816/r-bits.v3i4.4919](https://doi.org/10.18816/r-bits.v3i4.4919).
- [2] Jacob Fish, Ted Belytschko. *A first course in finite element analysis*. Wiley, 2007.
- [3] Svanberg, Krister. Mma and gcmma – two methods for nonlinear optimization, [s.d]. Disponível em: <https://people.kth.se/~krille/mmagcmma.pdf>.
- [4] Ce Zhang, Jianming Yang, Zongyu Chang. *Machinery Dynamics*. 1ª ed. Academic Press, 2021. ISBN 0128157852.
- [5] Silva Junior, Olavo Mecias da. *Otimização topológica em problemas de vibrações harmônicas forçadas utilizando potência de entrada*. Tese de doutorado, Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia Mecânica, Florianópolis, 2017.
- [6] M.P.Bendsøe e O.Sigmund. *Topology Optimization: Theory, Methods and Applications*. 2ª ed. Springer-Verlag Berlin Heidelberg GmbH, 2009. ISBN 978-3-662-05086-6.
- [7] Silva, Olavo M.; Neves, Miguel M. e Lenzi, Arcanjo. A critical analysis of using the dynamic compliance as objective function in topology optimization of one-material structures considering steady-state forced vibration problems. *Journal of Sound and Vibration*, 444:1–20, 2019. ISSN 0022-460X. doi: <https://doi.org/10.1016/j.jsv.2018.12.030>. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0022460X18308563>.
- [8] Ozkapici, Damla e Yaman, Ulas. Tiling of microstructures according to the density values of simp topology optimization. *Procedia Manufacturing*, 51:1033–1037, 01 2020. doi: [10.1016/j.promfg.2020.10.145](https://doi.org/10.1016/j.promfg.2020.10.145).
- [9] Deetman, Arjen. Python code of the method of moving asymptotes, 2020. Disponível em: <https://github.com/arjendeetman/GCMMA-MMA-Python>.
- [10] Svanberg, Krister. Mma and gcmma matlab code, [s.d]. Disponível em: <http://www.smoptit.se/>.
- [11] Silva, Olavo M.; Neves, Miguel M. e Lenzi, Arcanjo. On the use of active and reactive input power in topology optimization of one-material structures considering steady-state forced vibration problems. *Journal of Sound and Vibration*, 464:114989, 2020. ISSN 0022-460X. doi: <https://doi.org/10.1016/j.jsv.2019.114989>. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0022460X19305516>.
- [12] Silva, Olavo M.; Valentini, Fernando e Cardoso, Eduardo L. Shape and position preserving design of vibrating structures by controlling local energies through topology optimization. *Journal of Sound and Vibration*, 515:116478, 2021. ISSN 0022-460X. doi: <https://doi.org/10.1016/j.jsv.2021.116478>. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0022460X21005150>.
- [13] Lutz, Mark. *Learning Python*. 4ª ed. O'reilly, 2009. ISBN 978-0-596-15806-4.
- [14] Python official website, [s.d]. Disponível em: <https://www.python.org/>.
- [15] Limited, Riverbank Computing. Qt for python, [s.d]. Disponível em: <https://www.riverbankcomputing.com/software/pyqt/>.
- [16] John D. Hunter, Michael Droettboom. Matplotlib - matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in python., [s.d]. Disponível em: <https://matplotlib.org/>.
- [17] Campagnola, Luke. Pyqtgraph is a pure-python graphics and gui library built on pyqt5/pyside2 and numpy., [s.d]. Disponível em: <https://www.pyqtgraph.org/>.