



Felipe Martins Olival

Avaliação de algoritmos para otimização topológica de estruturas

Trabalho de Conclusão de Curso

Departamento de Engenharia Mecânica

Programa de Graduação em Engenharia Mecânica

Orientador: Anderson Pereira

Rio de Janeiro,
Novembro 2022

Resumo

A globalização econômica, o mercado extremamente competitivo e avanços tecnológicos inimagináveis, estimularam a busca por meios que tornassem os produtos industriais diferenciados e mais atraentes ao mercado. Nesse cenário, a otimização estrutural, em especial, a otimização topológica, recebeu um forte investimento nas últimas décadas. Diversos artigos, códigos educacionais e abertos, e software comerciais afloraram com o intuito de satisfazer as diferentes necessidades requisitadas pela indústria. A maioria dos software CAD, do inglês *computer aided design*, usados para a otimização topológica são ferramentas comercializadas, e o investimento nesse tipo de ferramenta nem sempre é viável. Portanto, este trabalho visa apresentar a viabilidade de utilizar algoritmos com códigos abertos como possíveis alternativas gratuitas para o *software* comercial. Dessa forma, por meio de um estudo de caso, é analisada a Otimização Topológica em estruturas de suporte para spoiler de carro de corrida. Para isso, são usados o código educacional top88 e top3d, inspirados no top99 de Ole Sigmund (2001) escrito em Matlab, e o *software* Altair Inspire. Os resultados da otimização topológica usando o top88 e top3d são comparados e validados com os obtidos com o Altair Inspire. Por fim, este trabalho também apresenta as principais diferenças, vantagens e desvantagens das ferramentas utilizadas na execução dos processos de otimização.

Palavras-Chave: Otimização topológica, códigos educacionais, software comercial

Evaluation of algorithms for topological optimization of structures

Abstract

Economic globalization, the extremely competitive market and unimaginable technological advances have stimulated the search for ways to make industrial products differentiable and more attractive to the market. In this scenario, structural optimization, in particular topological optimization, has received a strong investment in recent decades. Several articles, educational and open codes, and commercial software emerged in order to satisfy the different needs required by the industry. Most computer aided design (CAD) software used for topological optimization are commercial tools, and investment in this type of tool is not always viable. Therefore, this work aims to present the feasibility of using open-source algorithms as possible free alternatives to commercial software. Thus, through a case study, Topological Optimization is analyzed in support structures for racing car spoilers. For this, the educational code top88 and top3d, inspired by Ole Sigmund's top99 (2001) written in Matlab, and the Altair Inspire software are used. Topological optimization results using top88 and top3d are compared and validated with those obtained using Altair Inspire. Finally, this work also presents the main differences, advantages and disadvantages of the tools used in the execution of the optimization processes.

Keywords: Topology optimization, educational codes, commercial software

LISTA DE FIGURAS

Figura 1 - Publicações de artigos e pesquisas científicas a respeito da Otimização Estrutural ao longo dos anos. Fonte: [1].....	8
Figura 2 - Publicações de artigos e pesquisas científicas por categorias da Otimização Topológica e publicações por métodos da OT. Fonte: [1]	9
Figura 3 - Diferentes tipos de Elementos finitos. Fonte: [14]	12
Figura 4 - Exemplo de Malha de Elementos Finitos. Elementos finitos e Pontos Nodais. Imagem adaptada da Fonte: [11]	12
Figura 5 - Graus de Liberdade de um elemento finito em meio tridimensional. Fonte: [14]	13
Figura 6 - Matriz de rigidez global da estrutura. Fonte: https://www.esss.co/blog/metodos-numericos-para-simulacao-na-engenharia/	13
Figura 7 – (a) Otimização Paramétrica ou dimensional; (b) Otimização de Forma; (c) Otimização Topológica. Fonte: [3].....	15
Figura 8 – Fluxograma do processo de otimização topológica de uma estrutura. Fonte: [20]	17
Figura 9 – Imagem de um Spoiler com seu suporte. Fonte: www.amazon.com.br ...	18
Figura 10 – Suporte universal de spoiler. Fonte: www.amazon.com.br	19
Figura 11 – Vista frontal da peça bruta.....	20
Figura 12 – Desenho técnico da peça	21
Figura 13 – Numeração de elementos e graus de liberdade dos nós. Fonte: [11] ...	24
Figura 14 – Definição das condições de contorno (forças e pontos de fixação) da peça a ser estudada.....	24
Figura 15 - Definição de $x1_{Final}$, $x2_{Final}$ e $y1_{Final}$	25
Figura 16 – Definição de regiões vazias para a topologia final.....	26
Figura 17 – Exemplo da utilização da ferramenta “medida” do Solidworks para definição dos parâmetros “a” e “b” da equação da reta.....	26
Figura 18 – Definição de regiões sólidas para a topologia final.....	27
Figura 19 – Definições importantes do código original (top88).....	28
Figura 20 – Definições de pontos de aplicação da força e pontos de fixação no top3d	29
Figura 21 – Análise estática da peça com resultado de “deslocamento”	30
Figura 22 – Análise estática da peça com resultado de “Tensão de von Mises”	31

Figura 23 – Análise estática da peça com resultado de “Coeficiente de Segurança”	31
Figura 24 - Configurações para OT no Altair Inspire	32
Figura 25 – Otimização Topológica Altair Inspire; Maximização da rigidez; 50% volume; Rmin = 6; Tempo Decorrido: 00:09:13	33
Figura 26 – Revestimento sólido ao redor da topologia da Figura 25 obtida no processo de OT.....	33
Figura 27 - Análise de deslocamento da peça otimizada e pós processada	34
Figura 28 - Análise da Tensão von Mises da peça otimizada e pós processada	35
Figura 29 - Análise do Coeficiente de Segurança da peça otimizada e pós processada	35
Figura 30 – Otimização Topológica top3d	37
Figura 31 - Análise estática por “deslocamento” da topologia obtida com top3d	38
Figura 32 - Análise estática por “tensão de von Mises” da topologia obtida com top3d	38
Figura 33 - Análise estática por “coeficiente de segurança” da topologia obtida com top3d	39
Figura 34 – Otimização Topológica top88	40

SUMÁRIO

1. Introdução.....	14
1.1 Motivação	14
1.2 Objetivo.....	9
2. Revisão Bibliográfica	11
2.1 Método dos Elementos Finitos (MEF).....	11
2.2 Otimização Estrutural.....	14
2.3 Otimização Topológica	15
3. Apresentação do estudo de caso e Metodologia	18
3.1 Código 88-Linhas top88 (Bidimensional)	21
3.2 Código top3d (Tridimensional).....	28
3.3 Altair Inspire (software comercial).....	29
4. Resultados.....	32
4.1 Altair Inspire	32
4.2 Top3d.....	35
4.3 Top88.....	40
5. Conclusões	41
6. Referências Bibliográfica	42

1. Introdução

1.1 Motivação

A preocupação da humanidade com o perfeccionismo de projetos estruturais tem relatos milenares. Um dos principais exemplos de construções históricas são as famosas Pirâmides do Egito, que compõem a paisagem da região desde antes do nascimento de Cristo. Nessa época, o conhecimento teórico sobre estruturas era extremamente limitado e os projetos eram baseados na experiência. Com o passar do tempo, ainda baseados em experimentos, os padrões de construções foram evoluindo e hoje, estruturas datadas da antiguidade ainda são facilmente encontradas em nosso planeta [21].

Robert Hooke (1660), que dá nome à famosa Lei de Hooke, abre as portas para os métodos analíticos que permitem a solução de problemas mais simples baseados em formulações matemáticas. Contudo, problemas complexos, presentes na maioria dos projetos estruturais reais, só podem ser solucionados com métodos numéricos. Um método numérico vastamente utilizado no ramo da otimização estrutural é o Método dos Elementos Finitos (MEF), que será melhor abordado ao longo deste trabalho.

Atualmente, a demanda e necessidade por estruturas e peças cada vez mais “perfeitas” e complexas, seja por motivos de performance, comerciais, estéticos, ou qualquer outro, juntamente com o avanço tecnológico, impulsionaram o campo da otimização estrutural. A otimização estrutural é um processo numérico/matemático que permite a configuração de uma estrutura com um desempenho ótimo para determinado fim. É dividida em três principais categorias: otimização paramétrica, otimização de forma e otimização topológica (OT)[1]. Dessa forma, o advento da otimização estrutural com a finalidade de aumentar o desempenho de peças industriais numa economia globalizada e competitiva é perceptível pelo aumento de investimento em software comercial, pesquisas científicas e publicações de artigos, muitos deles com algoritmos educacionais e com código aberto de otimização. A Figura 1 evidencia a crescente de publicações de artigos e pesquisas científicas ao longo dos anos.

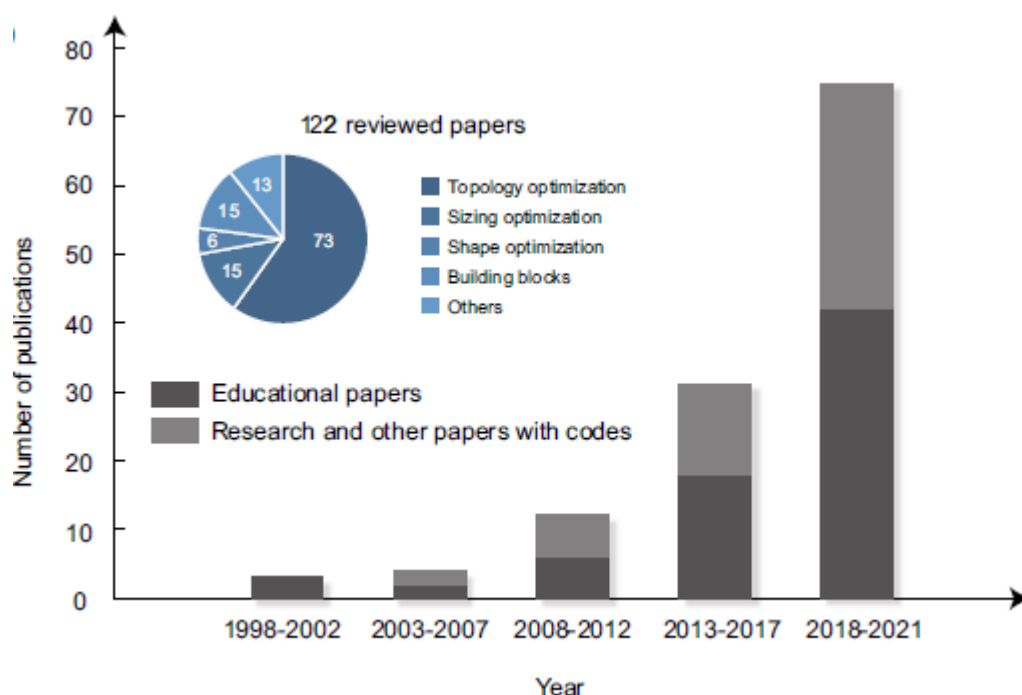


Figura 1 - Publicações de artigos e pesquisas científicas a respeito da Otimização Estrutural ao longo dos anos. Fonte: [1]

Dentre as categorias da otimização, a que mais se destaca é a Otimização Topológica (OT) que, por sua vez, também possui diferentes subcategorias. A mais comum e vastamente aplicada nesse trabalho é o método SIMP, do inglês, *Solid Isotropic Material with Penalization* [22]. Essa relação entre quantidade de publicações acadêmicas e categorias de OT são exemplificadas na Figura 2.

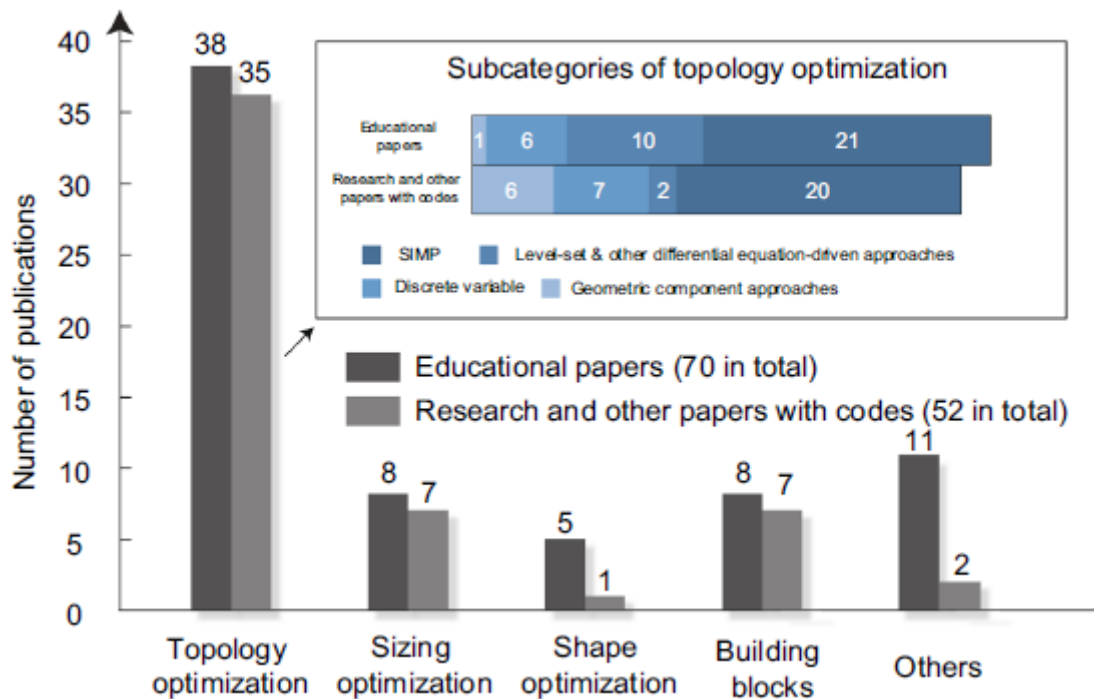


Figura 2 - Publicações de artigos e pesquisas científicas por categorias da Otimização Topológica e publicações por métodos da OT. Fonte: [1]

Como consequência do investimento na área da OT, muitos *software* comerciais e códigos *open-source* foram desenvolvidos com o objetivo de ajudar não apenas aos que estão iniciando na OT, mas também aos pesquisadores que buscam conhecimento sobre os métodos existentes.[1]

Na maioria dos casos, *software* comerciais são ferramentas poderosas e, dependendo da situação, a obtenção desse tipo de programa pode não ser viável.

1.2 Objetivo

Portanto, este trabalho objetiva analisar algoritmos com código aberto para otimização topológica de estruturas como uma possível opção gratuita aos programas comerciais. Para isso, será realizado um estudo de caso em que será executado a OT em estruturas de suporte de spoiler de carro de corrida. Serão utilizados o *software* comercial Altair Inspire [18] e os códigos educacionais top88[2] e top3d[17] escritos em MATLAB. O objetivo com a otimização é minimizar a *compliance* com restrição de volume no domínio das estruturas. Ao final, espera-se obter para os códigos educacionais a mesma topologia obtida com o Altair Inspire. Comprovando, assim, a viabilidade da utilização desses códigos para

resolução de problemas de otimização topológica a fim de auxiliar na confecção de peças e estruturas a serem comercializadas. Além disso, também serão abordadas as principais diferenças, vantagens e desvantagens na execução da OT usando cada ferramenta. Por fim, esse estudo de casos poderá ser útil a engenheiros que buscam compreender melhor a capacidade dos algoritmos *open-source* para a otimização topológica.

2. Revisão Bibliográfica

2.1 Método dos Elementos Finitos (MEF)

Como comentado anteriormente, a maioria dos problemas de engenharia são complexos, sendo inviável a solução por métodos analíticos. Portanto, esses tipos de problemas, como é o caso de projetos estruturais, são resolvidos por meio de métodos numéricos, como o Método dos Elementos Finitos (MEF).

A ideia por trás do método se resume a dividir o domínio de um objeto de estudo, considerado contínuo, em um número discreto de elementos/pequenos corpos. Esses elementos, conhecidos por “Elementos Finitos”, são conectados por nós ou pontos nodais, e podem assumir diferentes geometrias simples, apresentadas na Figura 3. Dessa forma, uma Malha de Elementos Finitos é definida a partir da divisão de um domínio em elementos que se conectam por nós (Figura 4). Logo, ao subdividir o domínio de um problema complexo em elementos finitos, o MEF torna o problema mais simples, reduzindo-o e equacionando-o a um sistema de equações algébricas. Considerando o atual cenário tecnológico e computacional, o processo de discretização, equacionamento e cálculo podem ser facilmente programados. Além disso, vale ressaltar a relação entre a precisão do método e a quantidade de elementos finitos e nós de uma malha. É possível a demonstração de que, quanto menor o tamanho e, consequentemente, maior a quantidade de elementos e nós, mais precisa é a análise dos elementos finitos. Em contrapartida, demanda-se uma maior capacidade computacional para a análise de uma malha com mais elementos finitos. [14]

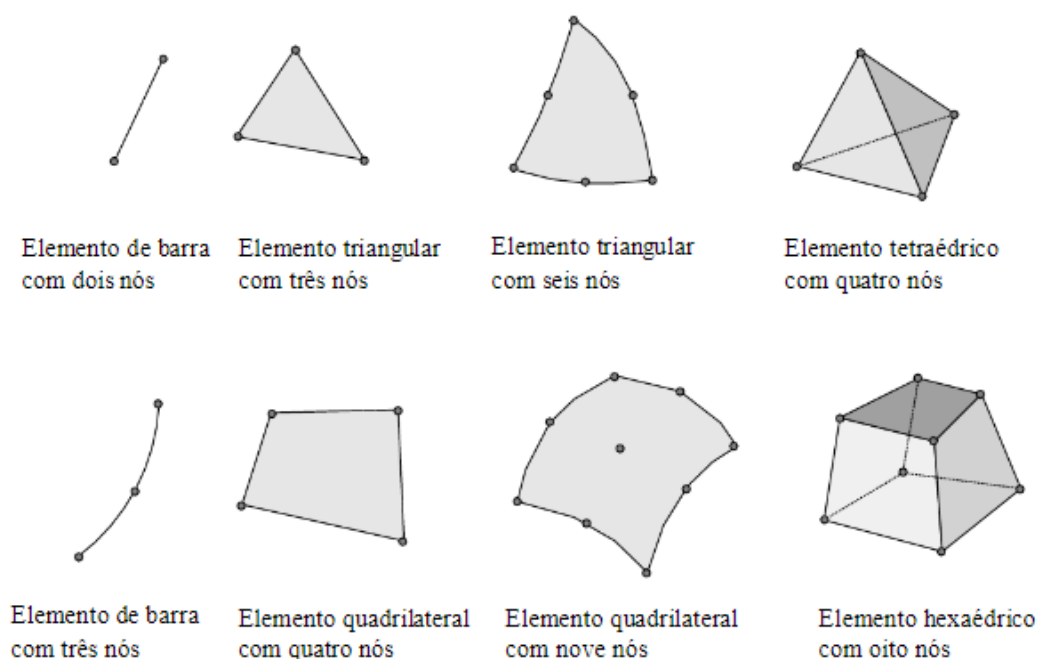


Figura 3 - Diferentes tipos de Elementos finitos. Fonte: [14]

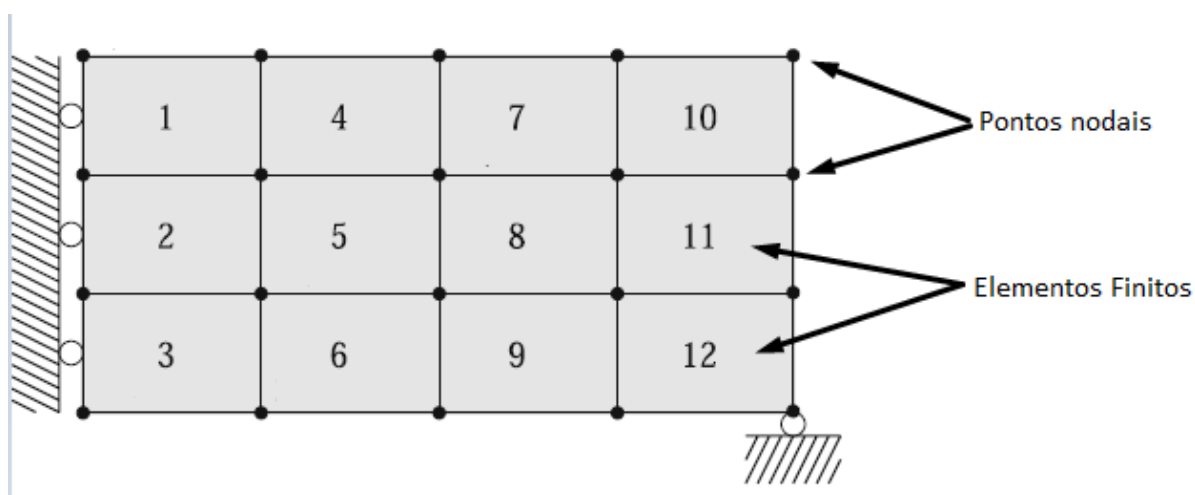


Figura 4 - Exemplo de Malha de Elementos Finitos. Elementos finitos e Pontos Nodais. Imagem adaptada da Fonte: [11]

Um conceito totalmente relacionado aos pontos nodais e elementos finitos no MEF é o conceito de Graus de Liberdade (GDL) ou, do inglês, *Degrees of Freedom* (DOF). O GDL indica as direções que um dado nó pode se deslocar. Por exemplo, em meio tridimensional, um elemento finito pode ter movimentos de translação e rotação nas três direções principais X, Y e Z. Ou seja, totalizando 6 graus de liberdade [14].

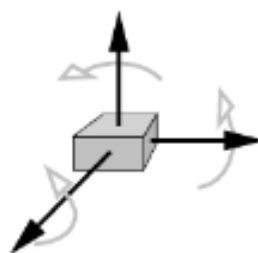


Figura 5 - Graus de Liberdade de um elemento finito em meio tridimensional. Fonte: [14]

Pelo fato do MEF transformar problemas complexos em problemas simples, a solução de sistemas de equações diferenciais pode se resumir a uma equação de estado que garante o equilíbrio. Considerando, por exemplo, o método analítico linear elástico da Lei de Hooke, a partir do conhecimento dos GDL, das propriedades do material do objeto de estudo e da análise de cada elemento finito, é possível determinar a rigidez de cada elemento finito e, conseqüentemente, uma matriz (K) de rigidez global da estrutura, ilustrada na Figura 6, onde “N” representa o número de graus de liberdade

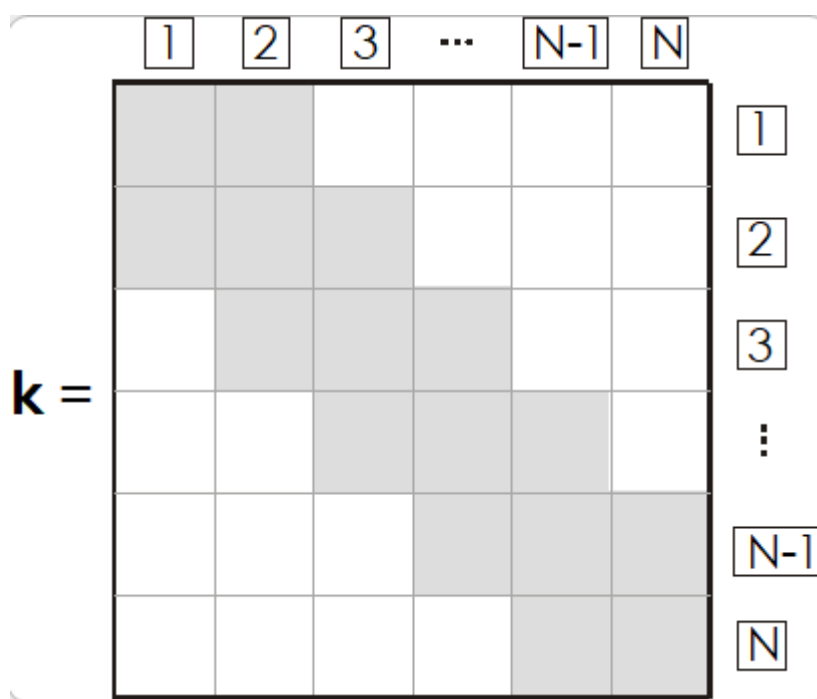


Figura 6 - Matriz de rigidez global da estrutura. Fonte: <https://www.esss.co/blog/metodos-numericos-para-simulacao-na-engenharia/>

Escrevendo a equação resultante da aplicação do MEF, temos:

$$\mathbf{F} = \mathbf{K}\mathbf{u}$$

onde,

\mathbf{F} = Vetor de Esforços

\mathbf{K} = Matriz de rigidez global

\mathbf{u} = Vetor de deslocamentos

2.2 Otimização Estrutural

Segundo o dicionário Michaelis, a definição do verbo “otimizar” é: “tornar ótimo” / “determinar o valor ótimo de uma grandeza”. Portanto, otimizar uma estrutura mecânica implica buscar por um melhor desempenho possível, respeitando restrições impostas como, por exemplo, a quantidade de material disponível. Limitar a quantidade de material pode significar reduzir drasticamente o custo de produção de uma peça confeccionada em escala industrial, como também, significa reduzir o peso de uma peça de veículos de competições automobilísticas, o que pode ser um fator determinante para a vitória.

Dessa forma, nesse período de grande investimento na OT por consequência de tantas motivações e da atual capacidade de processamento computacional, surge o código de 99 linhas em MATLAB de Sigmund (2001) [2] e seu algoritmo sucessor, o código de 88 linhas (Andreassen et al. 2011) [11]. Uma alternativa para a linguagem Python aos códigos de 99 e 88 linhas escritos em MATLAB, é o algoritmo de 165 linhas de Niels Aage e Villads Egede Johansen de 2013, com contribuição de Arjen Deetman [10]. Além do 88 linhas (MATLAB) e 165 linhas (Python), também existem códigos escritos em outras linguagens, como por exemplo a biblioteca TopOpt [7, 8] desenvolvida em Julia.

Como comentado rapidamente na introdução, a otimização estrutural é dividida em três principais categorias: paramétrica ou de dimensões, forma e topológica. A otimização paramétrica se preocupa com as dimensões das seções transversais de uma estrutura que minimiza, ou maximiza, a função objetivo respeitando as restrições. Já a otimização de forma, se atenta às formas no domínio de uma estrutura para atender a função objetivo. Por fim, a otimização topológica, foco deste trabalho, visa encontrar uma distribuição de material dentro do domínio de uma estrutura que minimize a função objetivo sujeita a um conjunto de restrições.

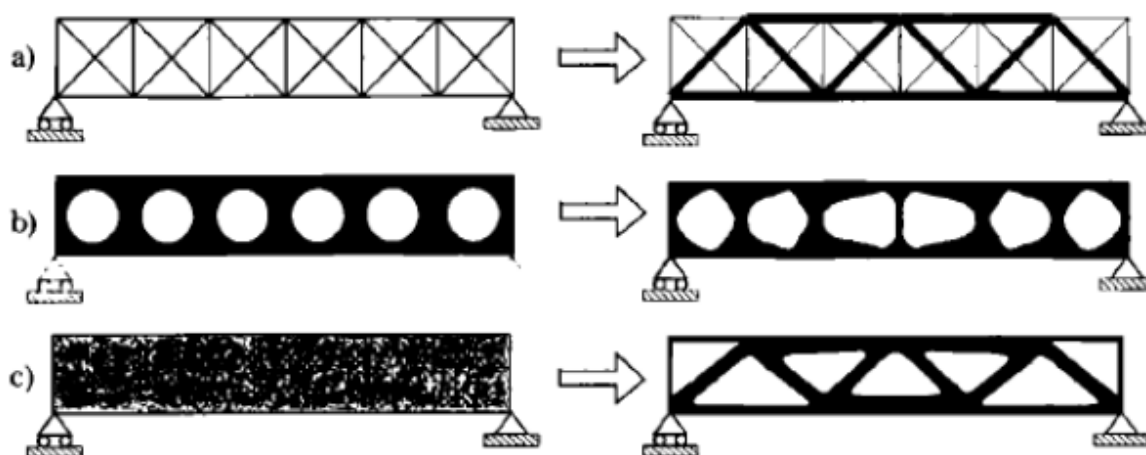


Figura 7 – (a) Otimização Paramétrica ou dimensional; (b) Otimização de Forma; (c) Otimização Topológica. Fonte: [3]

2.3 Otimização Topológica

Em poucas palavras, otimização topológica se resume em encontrar uma melhor distribuição do material dentro de um domínio (espaço de projeto). Por sua vez, o domínio é limitado pelas condições de contorno (restrições de deslocamento/ pontos de apoio), caracterizado por possuir regiões de aplicação de cargas e definido por suas demais variáveis de projeto. A distribuição ótima do material também depende da função objetivo que se deseja solucionar e das restrições impostas. Alguns exemplos de funções objetivas comumente usadas em otimização topológica são: maximização da rigidez, ou minimização da *compliance* (usado no estudo de caso presente nesse trabalho); minimização da massa; minimização da deformação; minimização da tensão; minimização da vibração, entre outras.

Nesse contexto, em um processo de OT, encontrar uma distribuição de material dentro do domínio significa identificar regiões cheias/sólidas no espaço, em que o material é indispensável para o objetivo funcional da peça ou estrutura, além das regiões não essenciais (“vazias”) e que, portanto, podem ter o material retirado. Com essa finalidade, são utilizados métodos de diferentes tipos como, por exemplo, os baseados na densidade dos elementos finitos [4, 6], os métodos “Level Set” e outras abordagens de equações diferenciais e, por fim, os métodos de abordagens de componentes geométricos. [1]

Neste trabalho, serão resolvidos problemas de OT utilizando um dos métodos baseados na densidade, o método SIMP [22]. Nas abordagens baseadas em

densidade, como o SIMP, o domínio da estrutura é discretizada por uma malha de elementos finitos, onde a densidade de cada elemento será otimizada. A metodologia SIMP se caracteriza por assumir como constante as propriedades do material dentro de cada elemento usado para discretizar o domínio, além de considerar as densidades dos elementos como variáveis de projeto. As densidades dos elementos variam de 0 (vazio, ausência de material) a 1 (sólido), com penalização dos valores intermediários. [1, 2]

A Figura 8 apresenta o fluxograma de um processo de otimização topológica que inicia com a estimativa das variáveis de projeto que definem o domínio a ser discretizado em uma malha de elementos finitos. As próximas etapas são a análise dos elementos finitos para cálculo dos deslocamentos e análise da sensibilidade das variáveis de projeto. Após esses passos, inicia-se o processo de otimização que buscará pela melhor distribuição do material dentro do espaço do projeto, respeitando a função objetiva. Caso o processo de otimização dentro de uma tolerância pré-definida, o processo é finalizado. Caso contrário, o procedimento se repete.

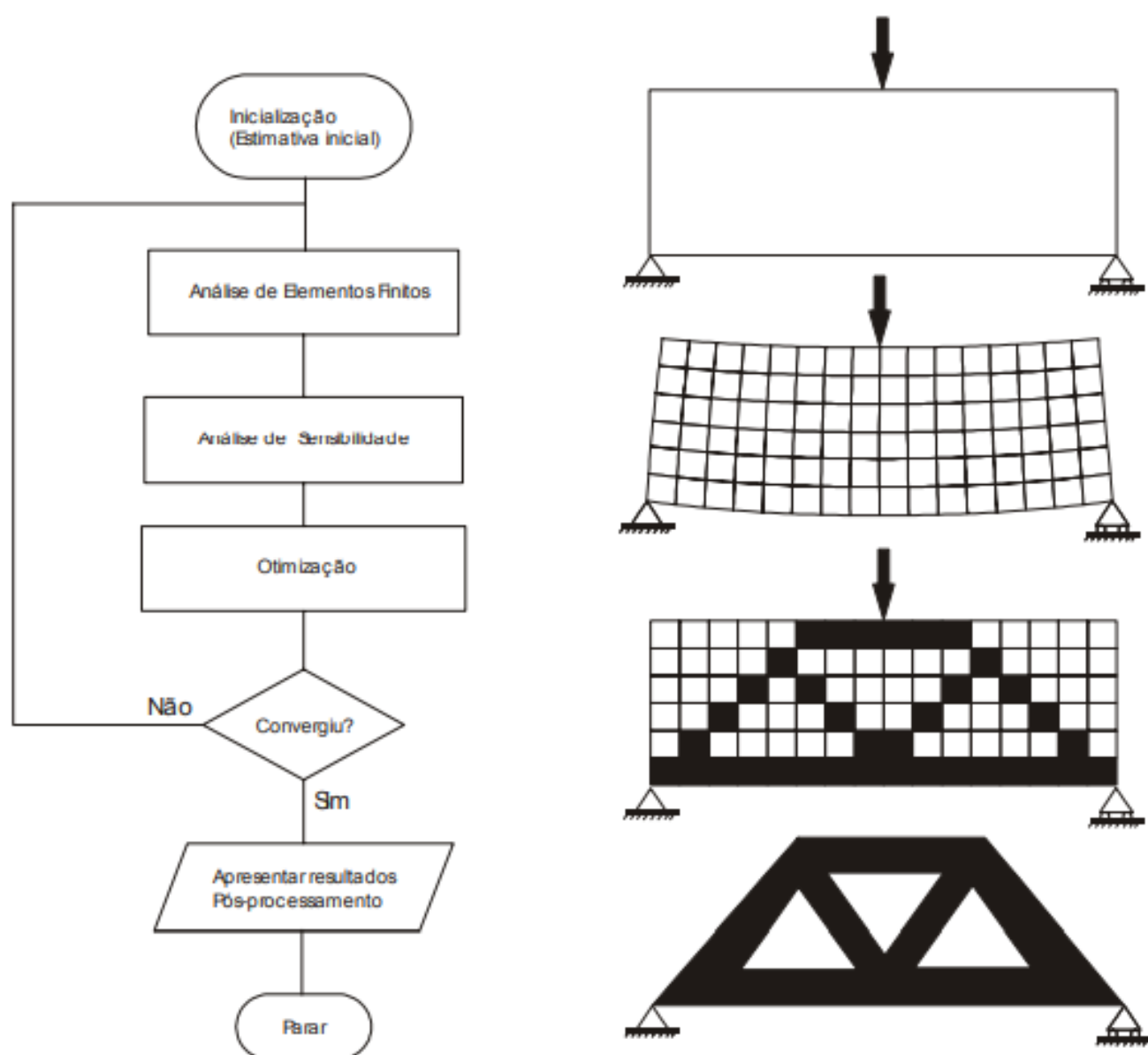


Figura 8 – Fluxograma do processo de otimização topológica de uma estrutura.
Fonte: [20]

3. Apresentação do estudo de caso e Metodologia

Em corridas automobilísticas a aerodinâmica é uma das principais preocupações das equipes. O grau de importância no assunto é tão alto que sutis diferenças no comportamento aerodinâmico entre um carro e outro pode ser um fator decisivo no resultado de uma corrida. Em altas velocidades, é formada uma área de baixa pressão em cima do veículo causada pela passagem de ar. A diferença de pressão entre as partes inferior e superior de um veículo pode reduzir a aderência do veículo ao solo, comprometendo o desempenho na corrida.

Dessa forma, o spoiler surge como um acessório cujo objetivo é aumentar a aderência do veículo ao solo, garantindo estabilidade. Devido à sua forma/*design*, quando o ar atinge o spoiler é gerado um esforço para baixo, também conhecido pelo termo em inglês “*downforce*”. A força atuando sobre o spoiler é transmitida ao veículo pelas hastes de suporte, objetos de estudo desse trabalho. Portanto, será executada uma otimização topológica em uma estrutura de braço de suporte de spoiler a fim de se obter a peça mais leve possível, ao mesmo tempo que seja resistente o suficiente para absorver a *downforce*.

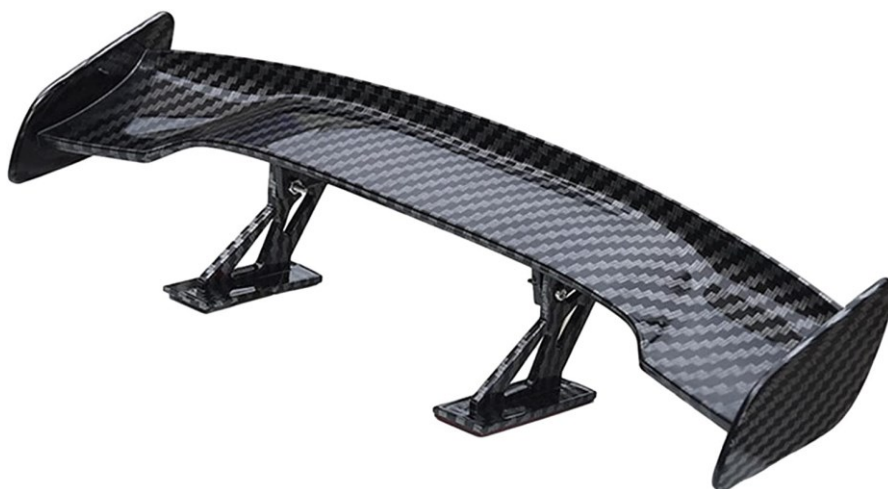


Figura 9 – Imagem de um Spoiler com seu suporte. Fonte: www.amazon.com.br



Figura 10 – Suporte universal de spoiler. Fonte: www.amazon.com.br

A Figura 10 ilustra um modelo universal para o suporte de spoiler facilmente encontrado no mercado. Assumindo esse modelo como exemplo, é esboçado no software CAD *Solidworks* uma peça bruta contendo o domínio inicial da peça em que será realizado o processo de otimização topológica. Vale ressaltar que apesar das versões mais recentes do *software Solidworks* possuir ferramentas de otimização topológica, a utilização desse programa no trabalho foi unicamente com o objetivo de auxiliar nas configurações da peça e uso dos demais *software*. A vista frontal e o desenho técnico da peça podem ser visualizadas nas Figuras 11 e 12.

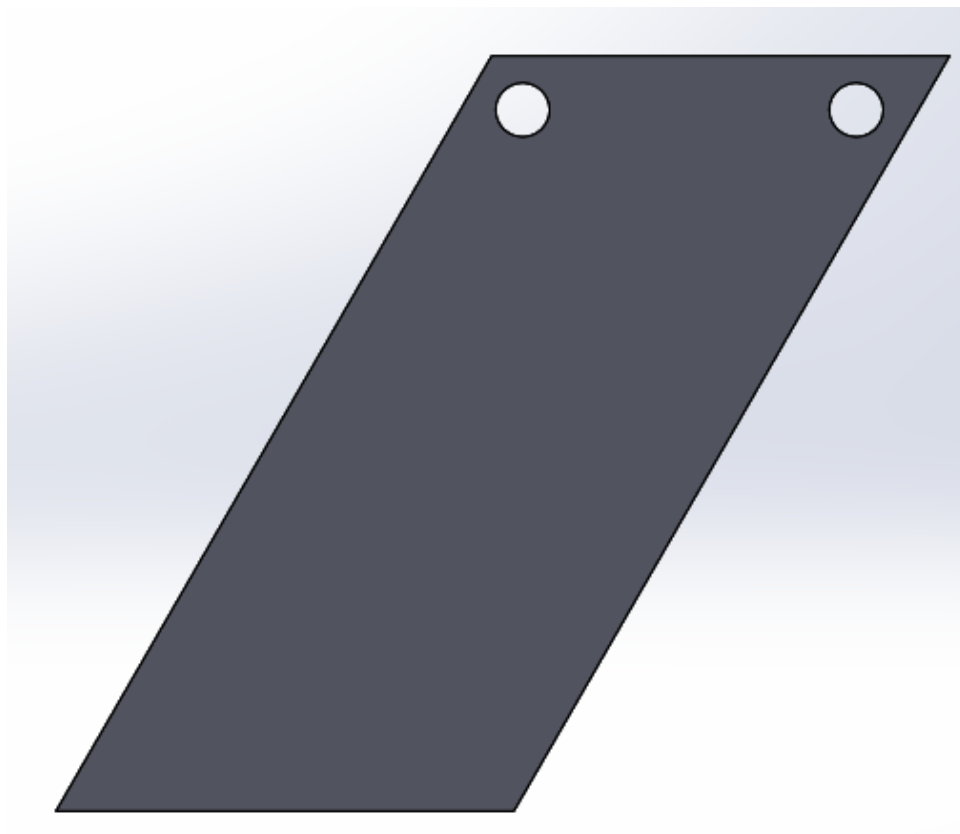


Figura 11 – Vista frontal da peça bruta

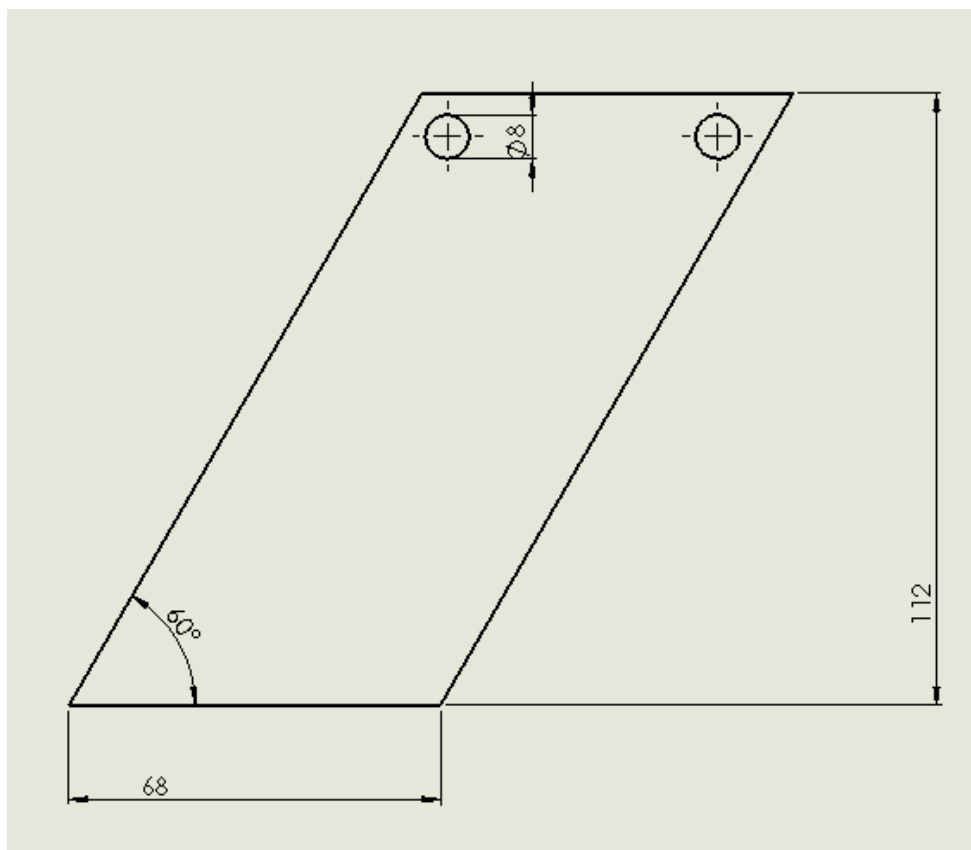


Figura 12 – Desenho técnico da peça

Nas próximas sessões serão apresentados os códigos educacionais e o software Altair Inspire usados neste trabalho para a resolução do clássico problema de OT que busca a minimização da *compliance* (flexibilidade), considerada como a função objetivo. Ou seja, maximização da rigidez. A otimização respeitará a restrição de 50% do volume original da estrutura. A peça é fixada na superfície inferior, e duas forças trativas na direção vertical, para baixo, são aplicadas nos furos.

3.1 Código 88-Linhas top88 (Bidimensional)

Antes do top88, o código top99 desenvolvido em MATLAB por Ole Sigmund em 2001 [2] é considerado um dos pioneiros e principais referências de códigos educacionais na área da OT. Das 99 linhas, 36 são destinadas ao programa principal, 12 linhas para o “otimizador” baseado em OC, 16 linhas para o filtro de independência de malha e 35 linhas para o código de Elementos Finitos (EF). [2] O método baseado em densidades utilizado no algoritmo é o *Solid Isotropic Material with Penalization* (SIMP). [4]

O código top88, publicado em 2011, é visto como sucessor direto do top99, trazendo pequenas modificações que contribuíram para a melhoria da eficiência do código [1, 11].

As chamadas para as funções dos códigos top99 e top88 são, respectivamente:

```
top(nelx,nely,volfrac,penal,rmin)
```

```
top88(nelx,nely,volfrac,penal,rmin,ft)
```

onde, **nelx** e **nely** são os números de elementos da malha na direção horizontal (x) e vertical (y), **volfrac** é a fração volumétrica da malha que se pretende restringir, **penal** é o expoente de penalização, **rmin** é o raio mínimo de espessura e **ft** representa a escolha do filtro para a realização da otimização no top88 (ft = 1 - filtro de sensibilidade; ft = 2 - filtro de densidade).

É importante observar que, nos códigos educacionais, o domínio da estrutura a ser otimizada encontra-se presente dentro da malha de elementos finitos. Além disso, a chamada da restrição de volume (**volfrac**) leva em consideração o volume da malha de elementos finitos. Dessa forma, para otimizar a peça com restrição de 50% do volume original, é preciso considerar o volume total da malha, que é maior que o domínio da peça. Consequentemente, o valor de **volfrac** a ser informado deverá ser menor que 50%.

Considerando o caso de estrutura linear elástica, a formulação do problema de minimização da *compliance* usando método SIMP, aplicado nos códigos top99 e top88, é escrito como: [2]

$$\left\{ \begin{array}{ll} \min_x: c(x) = U^T K U = \sum_{e=1}^N (x_e)^p u_e^T k_0 u_e & \\ \text{Sujeito a} & : \frac{V(x)}{V_0} = f \\ & : K U = F \\ & : 0 < x_{min} \leq x \leq 1 \end{array} \right. \quad (1)$$

onde, u_e e k_e são, respectivamente as matrizes de deslocamento e rigidez do elemento e , x é o vetor de variáveis de projeto (densidade), x_{min} é o vetor de densidades relativas mínimas (diferente de zero para evitar singularidades), N é o número de elementos utilizados para discretizar o domínio, resultado da multiplicação entre número de elementos horizontais pelo número de elementos verticais (=nelx*nely), p é o expoente de penalização do método SIMP, $V(x)$ e V_0

são o volume da peça e volume do domínio respectivamente, e f (*volfrac*) é a fração volumétrica que deve ser estabelecida pelo usuário do código para servir como restrição do problema de otimização. F e U são, respectivamente, vetores globais de esforços e deslocamentos e, K é a matriz de rigidez global [2].

A utilização do método SIMP original (Bendsøe 1989 [4]; Zhou and Rozvany 1991 [5]; Bendsøe and Sigmund 1999 [6]), no top99, estabelece a relação entre a densidade do elemento e o Módulo de Elasticidade do Material, dada por [1]:

$$E_e(x_e) = x_e^p E_0, \quad p \geq 1 \quad (2)$$

onde, E_0 é o módulo de elasticidade do material sólido e p é o fator de penalização, representado pela variável “**penal**” nas chamadas das funções do top99 e top88). Sendo $p = 3$ um valor normalmente utilizado [2].

Já para o código top88, o método SIMP original foi substituído pelo modificado, resultando em [1]:

$$E_e(x_e) = E_{min} + x_e^p (E_0 - E_{min}) \quad x_e \in [0,1] \quad (3)$$

onde, E_{min} é a rigidez do vazio, considerada um valor mínimo, diferente de zero, para evitar problemas numéricos [1].

Ambos os códigos, top99 e top88, assumem um domínio de *design* retangular, discretizado por elementos finitos quadrangulares. A numeração dos elementos e nós é feita de coluna a coluna, começando da esquerda para a direita. A Figura 13 ilustra uma malha de 12 elementos finitos quadrangulares para o caso da metade de viga bi apoiada. A figura também apresenta a numeração dos graus de liberdade (GDL) para o deslocamento horizontal e vertical do nó “**n**”, dado, respectivamente, por $2n - 1$ e $2n$ [11].

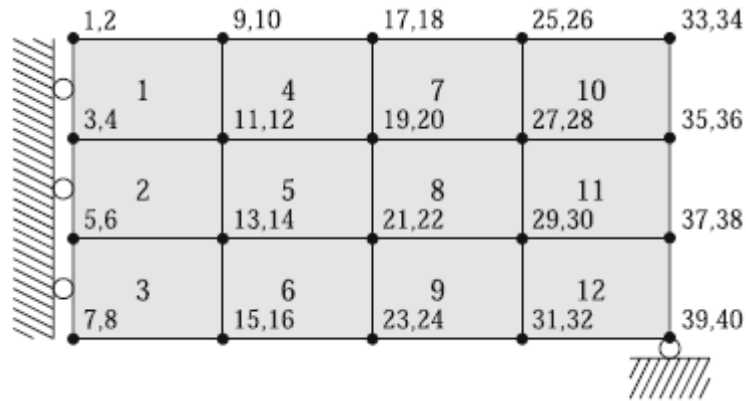


Figura 13 – Numeração de elementos e graus de liberdade dos nós. Fonte: [11]

Entendendo como funciona a numeração dos elementos, nós e graus de liberdade, é possível adaptar as condições de contorno no código para representar a peça a ser estudada.

Para definir as forças aplicadas na peça é preciso determinar as coordenadas de aplicação das forças (linha 188), o respectivo ID do nó (linha 189) e o grau de liberdade (linha 190). Dessa forma a força pode ser definida na linha 191.

A definição dos pontos de fixação segue a mesma lógica da força. Coordenadas (linha 193), IDs dos nós (linha 194) e graus de liberdade (linha 195).

```

188 - |      il = [x1Final x2Final]; j1 = [y1Final y1Final] ;           % Coordinates
189 - |      loadnid = il*(nely+1)+(nely+1-j1);                     % Node IDs
190 - |      loaddof = 2*loadnid(:);                                 % DOFs
191 - |      F = sparse(loaddof,1,-1,ndof,1);

192 - |      % USER-DEFINED SUPPORT FIXED DOFs
193 - |      [iif,jf] = meshgrid(0:68,0);                           % Coordinates
194 - |      fixednid = iif*(nely+1)+(nely+1-jf);                   % Node IDs
195 - |      fixeddofs = [2*fixednid(:); 2*fixednid(:)-1];          % DOFs

```

Figura 14 – Definição das condições de contorno (forças e pontos de fixação) da peça a ser estudada

Na Figura 14, `x1Final`, `x2Final` e `y1Final` representam as coordenadas de um conjunto de cinco pontos que determinam os arcos inferiores dos furos onde os parafusos transmitem força ao suporte do spoiler. A definição desses parâmetros em código é exibida na Figura 15.


```

167 % USER-DEFINED LOAD DOFs
168 - x1 = [];
169 - for y1 = 4:-1:0
170 -     x1 = [x1 sqrt(4^2 - y1^2)];
171 - end
172
173 - CopiaOx1 = repmat(69,1,5);    % [69 69 69 69 69]
174 - CopiaOx2 = repmat(119,1,5);   % [119 119 119 119 119]
175 - CopiaOy1 = repmat(104,1,5);   % [104 104 104 104 104]
176
177 - x1Final = CopiaOx1 + x1;
178 - x1Final = x1Final+[0,-6,0,-8,0];
179 - x1Final = round(x1Final);      % [69 66 72 65 73]
180
181 - x2Final = CopiaOx2 + x1;
182 - x2Final = x2Final+[0,-6,0,-8,0];
183 - x2Final = round(x2Final);      % [119 116 122 115 123]
184
185 - y1 = 4:-1:0;
186 - y1Final = CopiaOy1 - y1;      % [100 101 102 103 104]
187

```

Figura 15 - Definição de *x1Final*, *x2Final* e *y1Final*

Além da força e pontos de fixação, também é interessante limitar regiões da malha que desejamos garantir que sejam, ou não, sólidas na topologia final. Ou seja, determinar alguns elementos de interesse que terão densidade (variável de projeto) igual a 1 (sólido) e, elementos que terão densidade igual a 0 (vazio).

Como os furos por onde passam os parafusos responsáveis pela união entre suporte e spoiler não é composto por material da peça de estudo, é interessante garantir que uma região circular de raio 4, ao redor dos pontos de origem, seja vazia. Dessa forma, baseado na equação geral da circunferência $(x - a)^2 + (y - b)^2 = r^2$, são definidas as regiões nas linhas 299 e 301 (ver Figura 16). Todo elemento que se encontra no interior desses círculos de raio 4, recebem o valor de 0 (vazio) para a densidade (linha 309). Além disso, é interessante definir as outras linhas que delimitam o perímetro da peça. Logo, a linha 297 representa as equações das retas ($y = ax+b$) que determinam os limites da peça. Portanto, todo elemento que estiver fora do domínio do projeto, receberá valor 0 (vazio). Os valores de “a” e “b” foram obtidas com o auxílio da ferramenta de “medida” do *Solidworks*, conforme ilustrado na Figura 17.

```

295 - for ely = 1:nely
296 -     for elx = 1:nelx
297 -         if ely < -(112/64.66)*elx + (nely + 117.78) & ely > -(112/64.66)*elx + nely
298 -             passive(ely,elx) = 0;
299 -         elseif sqrt((ely-8)^2+(elx-69)^2) < 4
300 -             passive(ely,elx) = 0;
301 -         elseif sqrt((ely-8)^2+(elx-119)^2) < 4
302 -             passive(ely,elx) = 0;
303 -         else
304 -             passive(ely,elx) = 1;
305 -         end
306 -     end
307 - end
308 - passive = repmat(passive,[1,1]);
309 - x(find(passive)) = 0;

```

Figura 16 – Definição de regiões vazias para a topologia final.

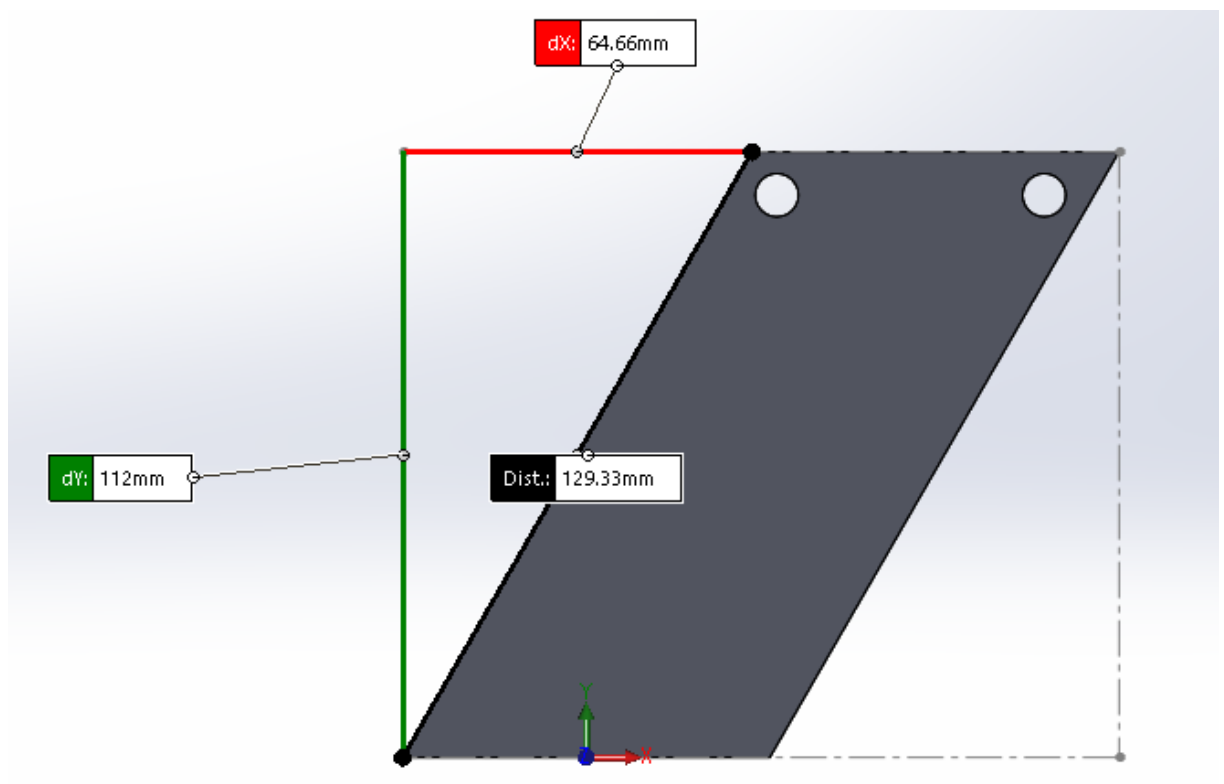


Figura 17 – Exemplo da utilização da ferramenta “medida” do *Solidworks* para definição dos parâmetros “a” e “b” da equação da reta.

Por outro lado, a região ao redor dos furos deve conter material (sólido = 1). Portanto, definimos essas regiões nas linhas 284 e 286, e atribuímos o valor de 1 para a densidade na linha 294, como mostrado na Figura 18.

```

282 - for ely = 1:nely
283 -     for elx = 1:nelx
284 -         if sqrt((ely-8)^2+(elx-69)^2) > 4 & sqrt((ely-8)^2+(elx-69)^2) < 8
285 -             active(ely,elx) = 1;
286 -         elseif sqrt((ely-8)^2+(elx-119)^2) > 4 & sqrt((ely-8)^2+(elx-119)^2) < 8
287 -             active(ely,elx) = 1;
288 -         else
289 -             active(ely,elx) = 0;
290 -         end
291 -     end
292 - end
293 - active = repmat(active,[1,1]);
294 - x(find(active)) = 1;

```

Figura 18 – Definição de regiões sólidas para a topologia final.

Outros pontos importantes, aproveitados do código original (top88), são a definição da matriz de rigidez global “K” e vetor global de deslocamentos “U” (linhas 316 e 319), a definição da função objetivo “*compliance*” (linha 322), os filtros de sensibilidade e densidade (linhas 340-345) e a sub-rotina do otimizador OC (linhas 347-367), tal como ilustrado na Figura 19.

```

311 %% START ITERATION
312 while change > 0.01
313     loop = loop + 1;
314     %% FE-ANALYSIS
315     sK = reshape(KE(:) * (Emin+xPhys(:)'.^penal*(E0-Emin)), 64*nex*nely, 1);
316     K = sparse(iK,jK,sK); K = (K+K')/2; % Matriz de rigidez global (K)
317
318     U(freedofs) = K(freedofs,freedofs)\F(freedofs); % Vetor global de deslocamentos (U)
319
320     %% OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
321     ce = reshape(sum((U(edofMat)*KE)'.*U(edofMat),2),nely,nex);
322     c = sum(sum((Emin+xPhys.^penal*(E0-Emin)).*ce)); % Função objetivo - Compliance (flexibilidade)
323     dc = -penal*(E0-Emin)*xPhys.^(penal-1).*ce;
324     dv = ones(nely,nex);
325
326     %% FILTERING/MODIFICATION OF SENSITIVITIES
327     if ft == 1 % Filtro de Sensibilidade
328         dc(:) = H*(x(:).^dc(:))./Hs./max(1e-3,x(:));
329     elseif ft == 2 % Filtro de Densidade
330         dc(:) = H*(dc(:))./Hs;
331         dv(:) = H*(dv(:))./Hs;
332     end
333
334     %% OPTIMALITY CRITERIA UPDATE OF DESIGN VARIABLES AND PHYSICAL DENSITIES
335     l1 = 0; l2 = 1e9; move = 0.2;
336     while (l2-l1)/(l1+l2) > 1e-3
337         lmid = 0.5*(l2+l1);
338         xnew = max(0,max(x-move,min(1,min(x+move,x.*sqrt(-dc./dv/lmid)))));
339
340         if bc == 7
341             xnew(find(active)) = 1;
342             xnew(find(passive)) = 0;
343         end
344
345         if ft == 1
346             xPhys = xnew;
347         elseif ft == 2
348             xPhys(:) = (H*xnew(:))./Hs;
349         end
350         if sum(xPhys(:)) > volfrac*nex*nely, l1 = lmid; else l2 = lmid; end
351     end
352     change = max(abs(xnew(:)-x(:)));
353     x = xnew;

```

Figura 19 – Definições importantes do código original (top88).

3.2 Código top3d (Tridimensional)

O código top3d [17] é também escrito em MATLAB e inspirado no top99 de Sigmund e, consequentemente, no top88. O código conta com melhorias e diferentes formas de escrita que foram usadas na adaptação do top88 para atender o problema da peça de estudo desse trabalho.

Dessa forma, as principais diferenças nas linhas de código, quando comparado com as apresentadas na seção do top88 (3.1), são referentes a inclusão da terceira dimensão, a direção Z, dando profundidade à peça. Na chamada da função do top3d percebemos a variável **nelz**, que representa a quantidade de elementos na direção Z.

top3d(nelx,nely,nelz,volfrac,penal,rmin)

A Figura 20 apresenta as Definições de pontos de aplicação da força e pontos de fixação no top3d.

```

57 - x1 = [];
58 - for y1 = 4:-1:0
59 -     x1 = [x1 sqrt(4^2 - y1^2)];
60 - end
61 -
62 - CopiaOx1 = repmat(69,1,5);           % [69 69 69 69 69]
63 - CopiaOx2 = repmat(119,1,5);          % [119 119 119 119 119]
64 - CopiaOy1 = repmat(104,1,5);          % [104 104 104 104 104]
65 -
66 - x1Final = CopiaOx1 + x1;
67 - x1Final = x1Final+[0,-6,0,-8,0];
68 - x1Final = round(x1Final);             % [69 66 72 65 73]
69 -
70 - x2Final = CopiaOx2 + x1;
71 - x2Final = x2Final+[0,-6,0,-8,0];
72 - x2Final = round(x2Final);             % [119 116 122 115 123]
73 -
74 - y1 = 4:-1:0;
75 - y1Final = CopiaOy1 - y1;              % [100 101 102 103 104]
76 -
77 - i1 = [x1Final x2Final]; j1 = [y1Final y1Final]; k1 = [repmat(nelz/2,1,5) repmat(nelz/2,1,5)]; % Coordinates
78 - loadnid = k1*(nelx+1)*(nely+1)+i1*(nely+1)+(nely+1-j1); % Node IDs
79 - loaddof = 3*loadnid(:)-1;              % DOFs
80 - F = sparse(loaddof,1,-1,ndof,1);
81 - F(loaddof,1) = -3000;                  % Força = 3000 N

83 - % USER-DEFINED SUPPORT FIXED DOFs
84 - [iif,jf,kf] = meshgrid(0:68,0,0:nelz); %0:34 % Coordinates
85 - fixednid = kf*(nelx+1)*(nely+1)+iif*(nely+1)+(nely+1-jf); % Node IDs
86 - fixeddof = [3*fixednid(:); 3*fixednid(:)-1; 3*fixednid(:)-2]; % DOFs

```

Figura 20 – Definições de pontos de aplicação da força e pontos de fixação no top3d.

3.3 Altair Inspire (software comercial)

O Altair Inspire [18] é uma ferramenta de desenho assistido por computador (DAC), mais conhecido como CAD, do *inglês*, *computer aided design*. O *software* é amplamente utilizado para otimização topológica e análise estrutural oferecendo uma grande diversidade de recursos e funcionalidades.

Portanto, durante o trabalho, foi modelada a peça de estudo e aplicado força de 100 N nos furos. O material definido para a peça é o aço (AISI 304) com módulo de elasticidade “E” igual a 195 GPa, coeficiente de Poisson “nu” 0,29 e limite de escoamento de 215 MPa. Em seguida, foi executada uma análise estática da peça nessa configuração.

Inicialmente, é executada uma análise estática do domínio inicial antes de realizar a otimização e alguns pontos interessantes são observados. Filtrando por “deslocamento”, fica nítido que a região onde é aplicada a força é onde se encontra o maior deslocamento da peça (Figura 21). Já a análise por tensão de von Mises mostra que a tensão máxima é de 47,87 MPa (Figura 22), logo, inferior ao limite de escoamento do material (215 MPa). Portanto, a aplicação de um carregamento de 100 N não provoca, a princípio, o escoamento da peça, o que é comprovado

pela análise do coeficiente de segurança, cujo valor mínimo é de 4.5 (Figura 23). Posteriormente, após o processo de otimização, será necessário executar outra análise estática para verificar se a topologia final encontrada continua sendo suficiente para absorver a força sobre o spoiler.

Além disso, vale ressaltar que a força média sobre um spoiler de um carro de corrida pode variar significativamente dependendo da velocidade do carro, do *design* e características do spoiler. A 100km/h, essa variação pode ir de algumas dezenas para algumas centenas de Newtons. Para a determinação mais assertiva da força são necessários testes em túneis de vento ou simulações computacionais mais específicas. Dessa forma, a escolha de 100N é aceitável para o estudo de caso em questão.

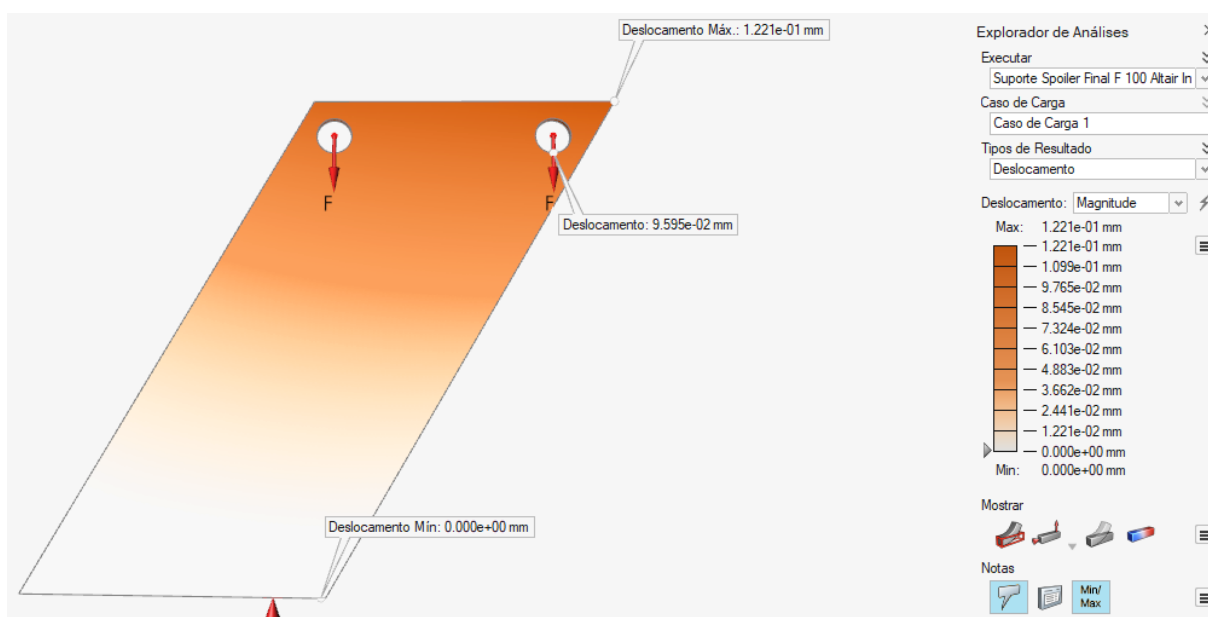


Figura 21 – Análise estática da peça com resultado de “deslocamento”

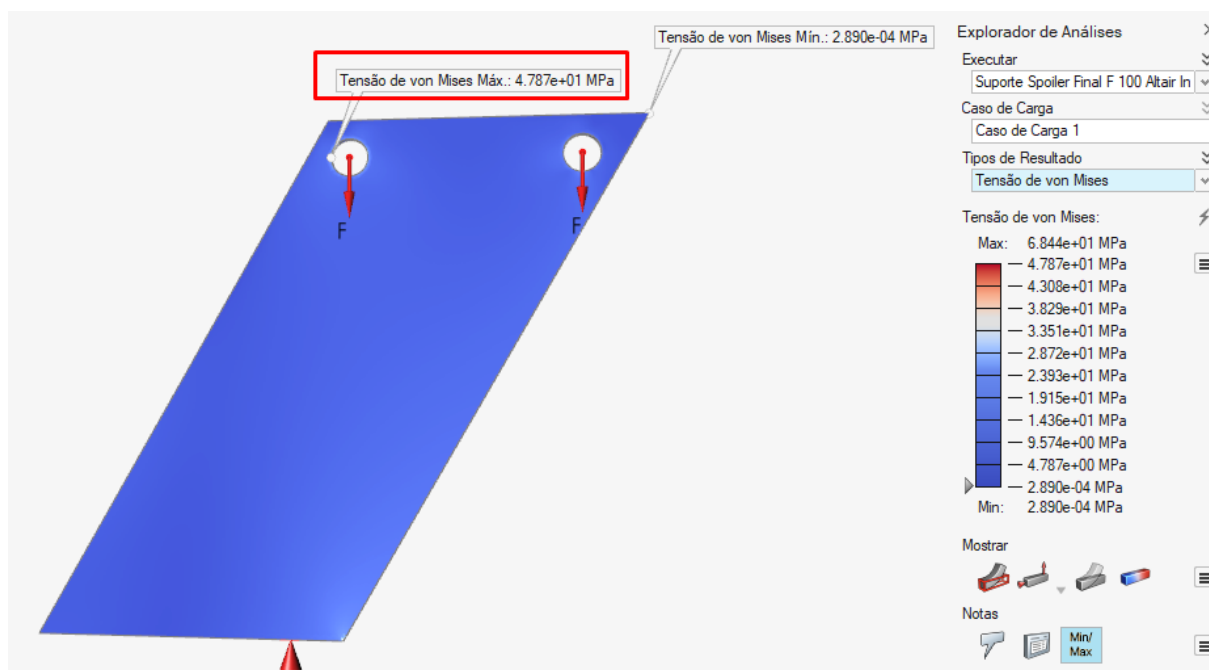


Figura 22 – Análise estática da peça com resultado de “Tensão de von Mises”

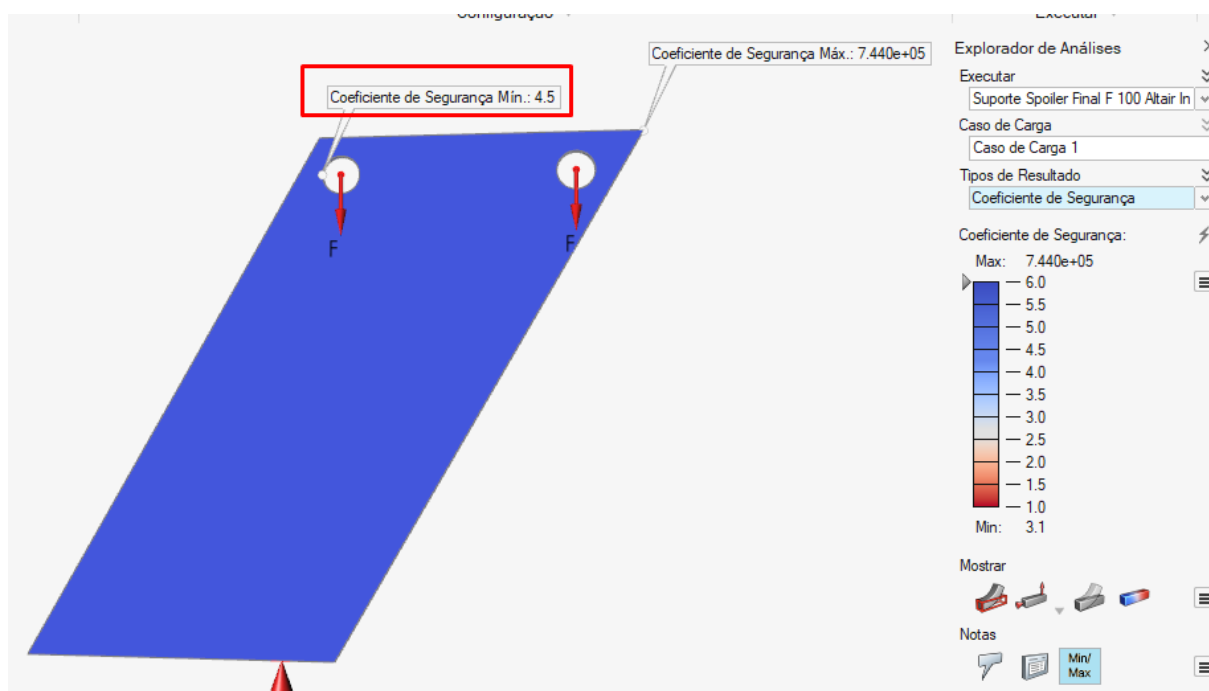


Figura 23 – Análise estática da peça com resultado de “Coeficiente de Segurança”

4. Resultados

4.1 Altair Inspire

Antes de executar a otimização topológica no Altair Inspire, é configurada uma malha com elementos de 1 milímetro de tamanho médio, assim como nos códigos Matlab. Com o tamanho dos elementos configurados, inicia-se a otimização topológica maximizando a rigidez, com restrição de 50% do volume e com 6 milímetros de raio mínimo de espessura (Figura 24). Esse último parâmetro, por sua vez, é o responsável por definir a menor espessura possível para o elemento finito no processo de OT, sendo recomendado valores de pelo menos 3 vezes o tamanho médio do EF.

Nove minutos e 13 segundos após o início, a otimização é finalizada, chegando na topologia da Figura 25. Um processo pós otimização é iniciado usando a ferramenta *Polynurbs* inclusa no *software*, revestindo um sólido ao redor da topologia. O resultado final desse processo é apresentado na Figura 26.

Executar Otimização

Nome: Suporte Spoiler Final F 100 Altair Inspire

Tipo: Topologia

Objetivo: Maximizar Rigidez

Massas-Alvo: % do Volume Total do Espaço de Projeto

5 10 15 20 25 30 35 40 45 50%

Restrições de Frequência

Nenhum

Maximizar frequências

Mínimo: 20 Hz Aplicar aos modos 10 mais baixos

Usar apoios do caso de carga: Sem Apoios

Restrições de Espessura

Mínimo: 6 mm

Máximo: 6.2 mm

Velocidade/Precisão

Contatos

O tipo de contato agora é definido usando a ferramenta Contatos.

Gravidade

Casos de Carga

Restaurar Exportar Executar Fechar

Figura 24 - Configurações para OT no Altair Inspire



🌐 Suporte Spoiler Rmin 6 F 100 Rigidez Máx. Massa 50% _ Modelo ✔ 06/06/2023 13:56:51 35.22 MB 00:09:13

Figura 25 – Otimização Topológica Altair Inspire; Maximização da rigidez; 50% volume; Rmin = 6; Tempo Decorrido: 00:09:13



Figura 26 – Revestimento sólido ao redor da topologia da Figura 25 obtida no processo de OT.

Uma nova análise estática é executada na topologia da peça final pós processado para ter certeza de que continua sendo resistente suficiente para absorver a *downforce*. Sendo assim, observa-se que o maior deslocamento ($1,181 \times 10^{-1} \text{ mm}$) ocorre na região de aplicação das forças (Figura 27). Além disso, percebe-se pela análise da tensão de von Mises (Figura 28) que o valor máximo ainda é inferior ao limite de escoamento do material (215 MPa), e que o coeficiente de segurança mínimo é maior que 1 (Figura 29). Portanto, a peça pós otimização e processamento não escoam com a aplicação do carregamento. Porém, vale ressaltar que os maiores valores de tensões estão concentrados em uma região reduzida. Configurando, assim, uma topologia não tão boa, mas válida por não apresentar risco a peça com carregamento de 100N.

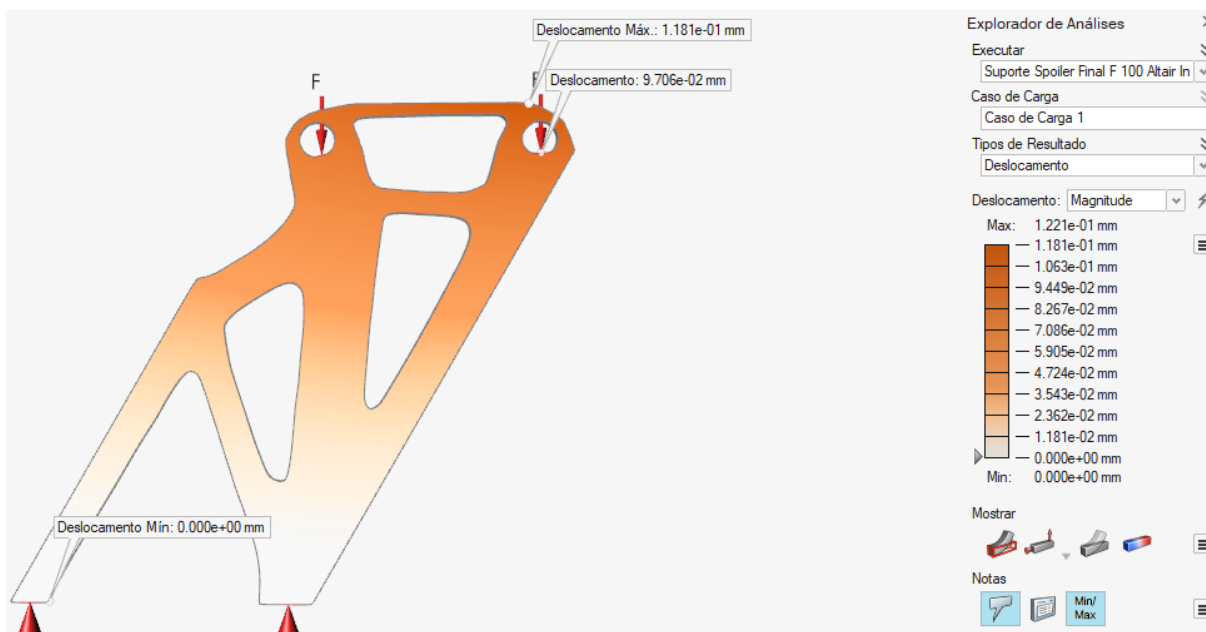


Figura 27 - Análise de deslocamento da peça otimizada e pós processada

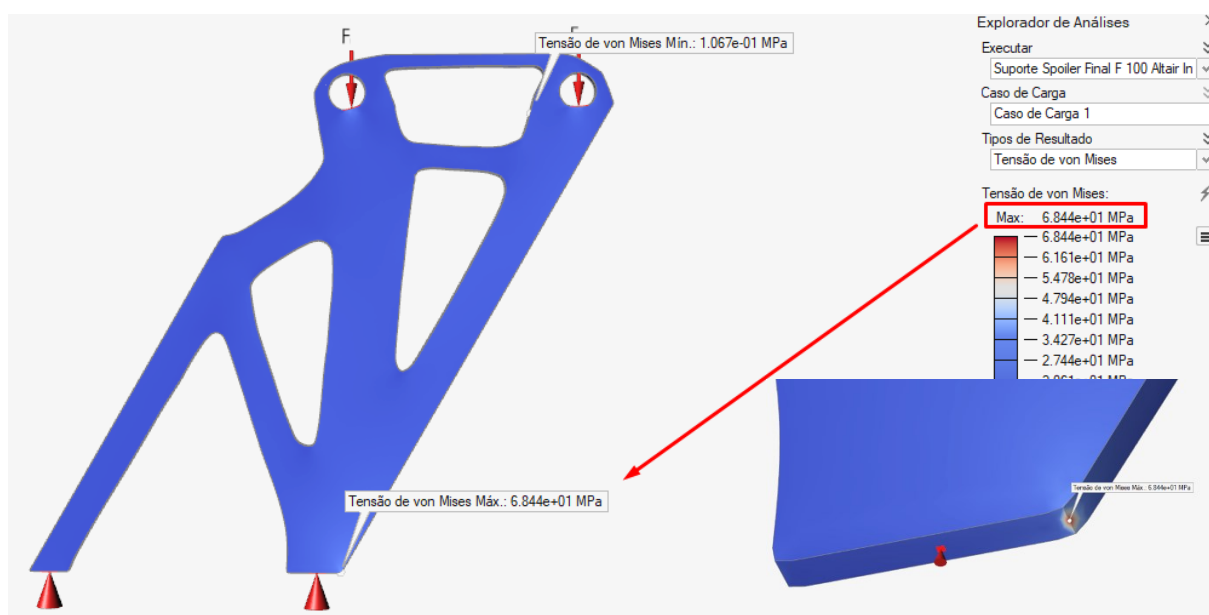


Figura 28 - Análise da Tensão von Mises da peça otimizada e pós processada

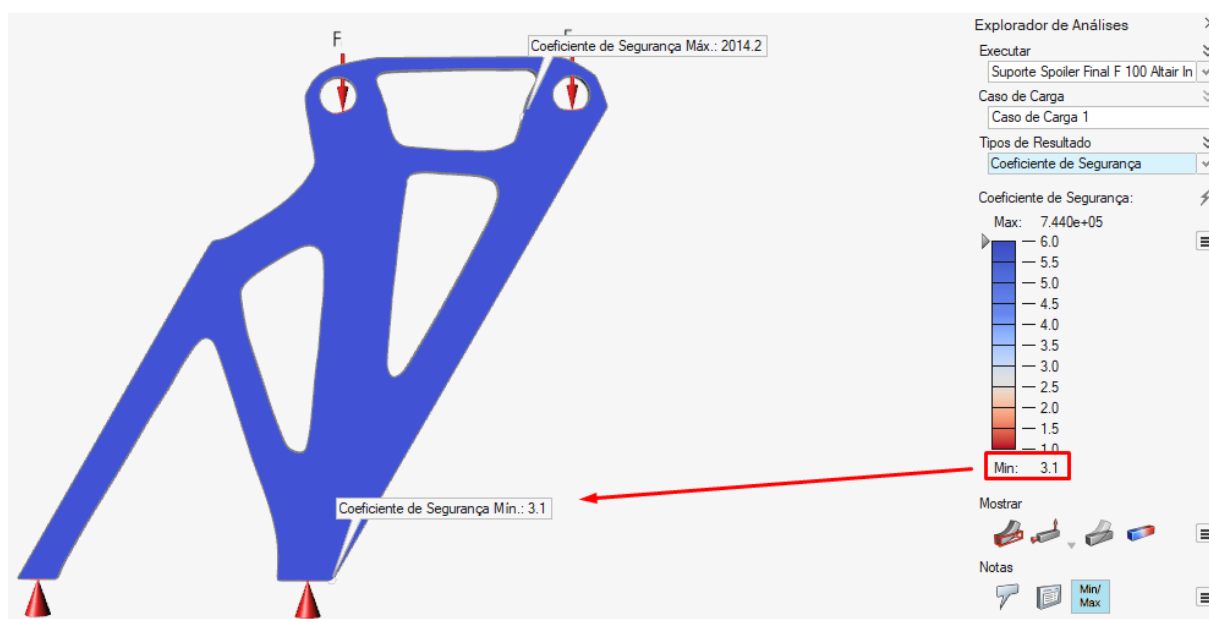


Figura 29 - Análise do Coeficiente de Segurança da peça otimizada e pós processada

4.2 Top3d

Levando em consideração a modelagem da peça em 3D no Altair Inspire, o código Matlab que melhor serve de comparação ao resultado do software CAD é

o top3d, que também simula a peça tridimensional. Sabendo disso, determina-se os parâmetros de entrada da chamada do código.

```
top3d(nelx,nely,nelz,volfrac,penal,rmin)
```

Respeitando a dimensão da peça modelada no Altair Inspire, os parâmetros *nelx*, *nely* e *nelz* são definidos como 133, 112, e 2 respectivamente. Para a otimização no top3d, um raio mínimo de espessura três vezes maior que o tamanho médio do elemento finito ($rmin = 3$) é um dos parâmetros que garante uma topologia semelhante a obtida com o *software* Altair Inspire.

A fração volumétrica (volfrac), por sua vez, não pode ser de 50% (0,50). Isso é explicado pelo domínio do projeto estar incluso e não representar 100% da malha de elementos finitos. O cálculo da fração volumétrica (volfrac) é feito por:

$$f = \frac{\text{Volume Peça otimizada Altair Inspire}}{nelx * nely * nelz} = \frac{8295}{133 * 112 * 2} = 0,278 \approx 0,28 \quad (4)$$

Dessa forma, com todos os parâmetros definidos, é iniciado o processo de OT ao chamar a função abaixo. O processo dura 1085.250841 segundos, sendo o equivalente a 00:18:05. A topologia final encontrada é semelhante com a obtida usando o Altair Inspire.

```
top3d(133,112,2,0.28,3,2)
```

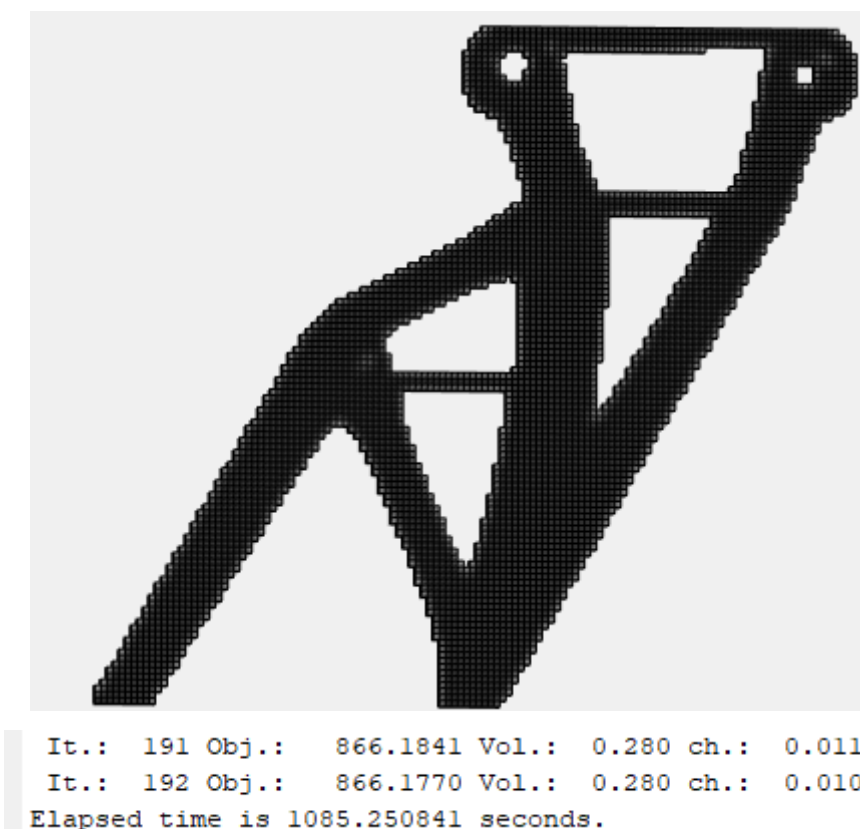


Figura 30 – Otimização Topológica top3d

A seguir, foi modelada a topologia da Figura 30 no Altair Inspire, obtida ao executar o processo de OT no top3d. Com a topologia modelada, é feita uma análise estática para que se possa comparar com a topologia obtida com a otimização do Altair Inspire.

Na Figura 31, observa-se que o descolamento máximo ($1,137 \times 10^{-1} \text{ mm}$) encontra-se na extremidade superior da peça, onde os carregamentos são aplicados. Já a Figura 32 apresenta a análise de tensão de von Mises, com maiores valores também concentrados em uma região reduzida. Apesar disso, nota-se que a topologia da peça obtida com o top3d também não escoava com a aplicação do carregamento que provoca tensão máxima de 58,28 MPa, bem inferior ao limite de escoamento do material de 215 MPa. O que é reforçado pelo coeficiente de segurança de 3.7 (Figura 33).

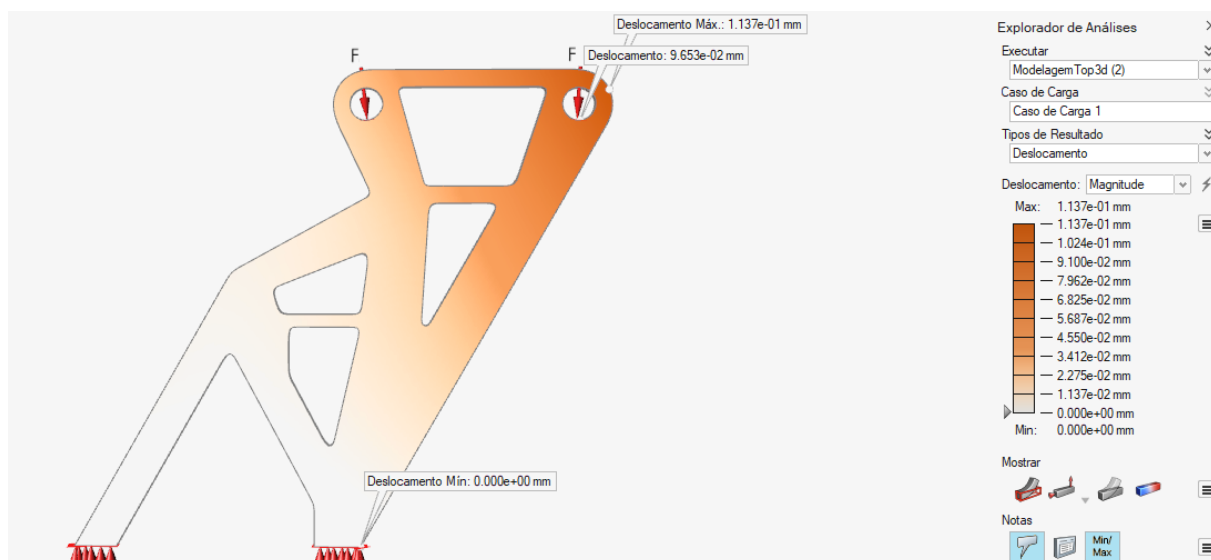


Figura 31- Análise estática por “deslocamento” da topologia obtida com top3d

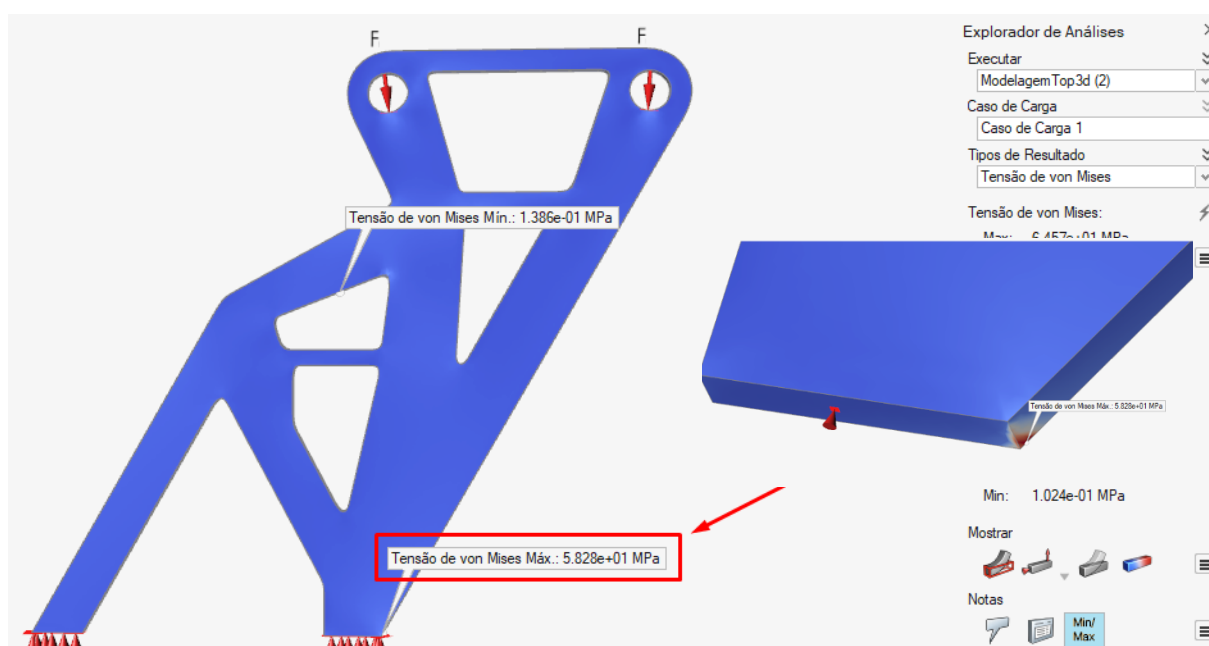


Figura 32 - Análise estática por “tensão de von Mises” da topologia obtida com top3d

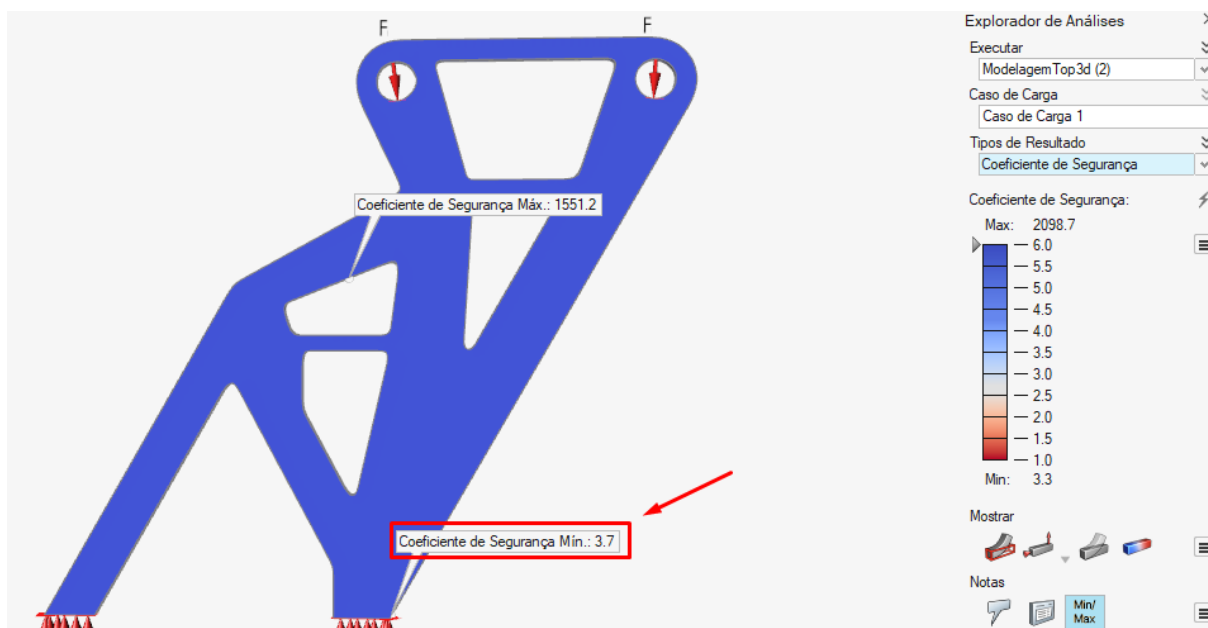


Figura 33 - Análise estática por “coeficiente de segurança” da topologia obtida com top3d

A análise da topologia obtida com top3d torna-se possível a comparação com os resultados da análise da peça obtida com o Altair Inspire. Para facilitar a visualização, os dados das análises foram organizados na tabela abaixo.

	<i>Altair Inspire</i>	<i>Código MATLAB</i>
<i>Deslocamento no ponto de aplicação de carga</i>	$9,706 \times 10^{-2} \text{ mm}$	$9,653 \times 10^{-2} \text{ mm}$
<i>Tensão de von Mises máxima</i>	68,44 MPa	58,28 MPa
<i>Coeficiente de Segurança mínimo</i>	3,1	3,7
<i>Massa da estrutura</i>	0,066177 kg	0,069182 kg

Como esperado, após otimização, a análise por deslocamento no ponto de aplicação de carga apresenta valores superiores ao encontrado no domínio inicial, $9,595 \times 10^{-2} \text{ mm}$. Isso é explicado pela retirada do material durante o processo de OT. Contudo, por se tratar de uma otimização topológica com objetivo de aumentar a rigidez, o deslocamento máximo obtido pós processo de otimização deve ser inferior ao da peça atualmente comercializada. Além disso, a tabela também mostra a proximidade dos resultados obtidos entre Altair Inspire e o top3d, o que nos aproxima do objetivo desse trabalho: mostrar que os códigos abertos podem ser úteis para a otimização de peças industriais.

4.3 Top88

Apesar do código educacional top88 (bidimensional) não ser a melhor opção de comparação com a otimização executada no Altair Inspire (tridimensional), como a peça foi modelada com uma pequena espessura de 2 milímetros de profundidade, a topologia final usando o top88 não ficaria muito distante das demais. Logo, usando os mesmos parâmetros usados no top3d, é obtida a topologia da Figura 34 abaixo.



Figura 34 – Otimização Topológica top88

5. Conclusões

Este trabalho contou com o estudo da otimização topológica em um suporte de *spoiler* com objetivos além da simples otimização da peça. Um dos principais objetivos era comprovar a viabilidade da utilização de códigos abertos como possíveis alternativas para ferramentas de otimização em meio industrial/comercial. Dessa forma, os resultados nos mostra que sim, algoritmos com código aberto podem ser úteis no processo de otimização de peças encontradas no mercado.

A seguir, os códigos *open-source* e *software* comerciais serão comparados por meio de avaliações em diferentes categorias, expondo as vantagens e desvantagens de cada um.

No quesito capacidade computacional, o Altair Inspire exige bastante por ser uma ferramenta poderosa, oferecendo diversas funcionalidades e recursos como, por exemplo, análise estática, interface gráfica, *Polynurbs* e muitas outras. Além de ser preparado para lidar com problemas complexos e de grandes volumes. Por sua vez, os códigos abertos não possuem as ferramentas que os *software* comerciais oferecem, além de não serem capaz de lidar com peças complexas e com grandes dimensões. Exemplificando, apesar das tentativas, não foi possível executar um processo de OT da peça objeto de estudo no top3d usando 10 elementos na direção Z ($nelz = 10$), pois não havia capacidade computacional para tanto.

Uma desvantagem dos códigos abertos comparados aos *software* comerciais é a necessidade de modelar regiões maiores das que são modeladas nos *software* comerciais. Na maioria dos casos, essas regiões incluem áreas que serão vazias. Portanto, o processo de otimização nos códigos *open-source* utilizam de elementos de malha desnecessários, que podem influenciar no tempo em que a tarefa é executada. Nesse trabalho, essa influência é evidenciada ao comparar as velocidades de processamento da OT executada no Altair Inspire e no top3d. O processo no *software Inspire* durou a metade do tempo corrido no top3d (Inspire: 00:09:13; top3d: 00:18:05).

Apesar das vantagens dos *software* comerciais apresentadas acima, os códigos abertos foram capazes de executar o processo de OT do suporte de *spoiler*. Além disso, é possível analisar outras categorias de comparação que concedem uma grande vantagem aos códigos abertos: curva de aprendizado

reduzida por possuir menos recursos e funcionalidades, a flexibilidade e personalização possibilitada por ser genuinamente um código *open-source* e, por fim, o custo nulo para se utilizar esses algoritmos.

Logo, comprova-se a viabilidade dos códigos abertos como alternativa aos *software* comerciais. Contudo, não é possível apontar um sendo superior ao outro. Cada um possui suas próprias características, vantagens e desvantagens. A escolha entre um *software* comercial ou um código aberto para um projeto depende dos objetivos, necessidades e recursos disponíveis.

Por fim, vale ressaltar que os algoritmos top3d e top88 utilizados nesse trabalho são códigos educacionais, geralmente usados para fins acadêmicos. Porém existem pacotes específicos para OT escritos em códigos aberto, como por exemplo a biblioteca TopOpt [7, 8] desenvolvida em Julia e o pacote TopOpt escrito em MATLAB [19].

6. Referências Bibliográfica

- 1 – Wang, C., Zhao, Z., Zhou, M. et al. A comprehensive review of educational articles on structural and multidisciplinary optimization. *Struct Multidisc Optim* 64, 2827–2880 (2021). <https://doi.org/10.1007/s00158-021-03050-7>
- 2 - Sigmund, O. A 99 line topology optimization code written in Matlab. *Struct Multidisc Optim* 21, 120–127 (2001). <https://doi.org/10.1007/s001580050176>
- 3 - Bendsøe MP, Sigmund O (2013) *Topology optimization: theory, methods, and applications*. Springer, New York
- 4 – Bendsøe MP (1989) Optimal shape design as a material distribution problem. *Struct Optim* 1(4):193–202
- 5 – Zhou M, Rozvany G (1991) The COC algorithm, Part II: topological, geometrical and generalized shape optimization. *Comput Methods Appl Mech Eng* 89(1–3):309–336
- 6 – Bendsøe MP, Sigmund O (1999) Material interpolation schemes in topology optimization. *Arch Appl Mech* 69(9):635–654
- 7 – Mohamed Tarek Mohamed. “TopOpt.jl: An efficient and high-performance topology optimization package in the Julia programming language”. In: 13th World Congress of Structural and Multidisciplinary Optimisation, WCSMO 13. 2019.
- 8 – Huang, Yijiang and Tarek, Mohamed. TopOpt.jl: Truss and Continuum Topology Optimization, Interactive Visualization, Automatic Differentiation and

More. Proceedings of the 14th World Congress of Structural and Multidisciplinary Optimization

9 – Svanberg, K. (1987). The Method of Moving Asymptotes – A new method for structural optimization. International Journal for Numerical Methods in Engineering 24, 359-373. doi:10.1002/nme.1620240207

10 – Aage, N., Johansen, V.E. (2013). A 165 LINE TOPOLOGY OPTIMIZATION CODE

11 – Andreassen, E., Clausen, A., Schevenels, M. et al. Efficient topology optimization in MATLAB using 88 lines of code. Struct Multidisc Optim 43, 1–16 (2011). <https://doi.org/10.1007/s00158-010-0594-7>

12 – Nie, Zhenguo & Lin, Tong & Jiang, Haoliang & Kara, Levent. (2021). TopologyGAN: Topology Optimization Using Generative Adversarial Networks Based on Physical Fields Over the Initial Domain. Journal of Mechanical Design. 143. 1-13. 10.1115/1.4049533.

13 - Haung, X.; Xie. Y. M. “Evolutionary topology optimization of continuum structures: methods and applications”, Wiley, 2010.

14 - Souza, R. M. (2003). O Método dos Elementos Finitos Aplicado ao Problema de Condução de Calor. Universidade Federal do Pará.

15 - Beckers, M. (1999). *Topology optimization using a dual method with discrete variables. Structural Optimization*. 17: 14–24. doi:10.1007/BF01197709

16 – Manso, D. C (2016). Otimização topológica para problemas de condução de calor usando elementos finitos poligonais. Pontifícia Universidade Católica do Rio de Janeiro – PUC-rio

17 - K. Liu and A. Tovar, “An efficient 3D topology optimization code written in Matlab”, Struct Multidisc Optim, 50(6): 1175-1196, 2014, doi:10.1007/s00158-014-1107-x

18 - Altair Inspire, <https://www.altair.com/inspire-applications>

19 - Smith, J. (2019). TopOpt: A MATLAB package for topology optimization. [Software]. Disponível em: <https://github.com/topopt/TopOpt-in-Matlab>. Acesso em: 1º de junho de 2023.

20 – MAIA, Sheyla Maurício. Otimização topológica aplicada ao problema de maximização da frequência natural. 2021. 51 f. Monografia. Universidade Federal de Sergipe. Disponível em:

https://ri.ufs.br/bitstream/riufs/15878/2/Sheyla_Mauricio_Maia.pdf. Acesso em: 21 de junho de 2023

21 – PICEELLI, Renato. Otimização Topológica. 2020. Disponível em: https://edisciplinas.usp.br/pluginfile.php/5419091/mod_resource/content/1/aula05_slides.pdf. Acesso em: 21 de junho de 2023.

22 - Bendsoe, Martin P., and Ole Sigmund. Topology Optimization: Theory, Methods, and Applications. 2^a ed. New York: Springer, 2013.