

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

ENRICO SPERANDIO BELTRÃO FRALETTI

**VALIDAÇÃO DE METODOLOGIA DE OTIMIZAÇÃO TOPOLÓGICA COM
FERRAMENTAS DE CÓDIGO LIVRE**

CURITIBA

2024

ENRICO SPERANDIO BELTRÃO FRALETTI

**VALIDAÇÃO DE METODOLOGIA DE OTIMIZAÇÃO TOPOLÓGICA COM
FERRAMENTAS DE CÓDIGO LIVRE**

Topology Optimization Method Validation Using Free Software

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia Mecânica do Curso de Engenharia Mecânica da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Eduardo Matos Germer

CURITIBA

2024



[4.0 Internacional](https://creativecommons.org/licenses/by-nc-nd/4.0/)

Esta licença permite download e compartilhamento do trabalho desde que sejam atribuídos créditos ao(s) autor(es), sem a possibilidade de alterá-lo ou utilizá-lo para fins comerciais. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

ENRICO SPERANDIO BELTRÃO FRALETTI

**VALIDAÇÃO DE METODOLOGIA DE OTIMIZAÇÃO TOPOLÓGICA COM
FERRAMENTAS DE CÓDIGO LIVRE**

Trabalho de Conclusão de Curso de Graduação
apresentado como requisito para obtenção do
título de Bacharel em Engenharia Mecânica do
Curso de Engenharia Mecânica da Universidade
Tecnológica Federal do Paraná.

Data de aprovação: 21/junho/2024

Eduardo Matos Germer
Doutor
Universidade Tecnológica Federal do Paraná

Samuel Soares Ansay
Mestre
Universidade Tecnológica Federal do Paraná

Rodrigo Schlichting
Doutor
Universidade Tecnológica Federal do Paraná

**CURITIBA
2024**

RESUMO

A otimização é um aspecto fundamental da engenharia, buscando encontrar a melhor solução, considerando os conjuntos de variáveis e restrições que regem o desempenho de um sistema. A otimização topológica é uma técnica que procura obter a melhor distribuição de material e desempenho dentro de um volume. Aplicada em engenharia, permite projetar estruturas mais leves e resistentes, ou superfícies onde haja máximo fluxo de transporte. Os programas de código livre são aqueles em que são concedidas as liberdades de leitura, execução, redistribuição e modificação de seus códigos-fonte. Quando utilizados, garantem indefinidamente aos arquivos e cálculos de um projeto validade, acessibilidade e evolução. Com o objetivo de explorar a otimização topológica utilizando somente programas de código livre, este trabalho apresenta algoritmos e pesquisas recentes, onde o estudo e a aplicação da otimização topológica em projetos de setores como civil e aeronáutico trouxe benefícios, concluindo então numa implementação em que se utilizam apenas ferramentas de código livre para realizar o projeto, a modelagem, a simulação, a otimização, e o pós-processamento de uma peça, da qual foi fabricado um par, original e aliviada, que foi submetido a ensaios de comportamento mecânico, a fim de documentar o processo e validar os programas utilizados para obter resultados satisfatórios em engenharia, oferecendo uma alternativa confiável.

Palavras-chave: otimização topológica; python; código livre; projeto auxiliado por computador; manufatura aditiva.

ABSTRACT

Optimization is a fundamental aspect of engineering, seeking to find the best solution, considering the sets of variables and restrictions that govern the performance of a system. Topology optimization is a technique that seeks to obtain the best material distribution and performance within a volume. Applied in engineering, it allows the design of lighter and more resistant structures, or surfaces where there is maximum transport flow. Free and open source software are those in which the freedoms of reading, executing, redistributing and modifying their algorithms are granted. When used, they indefinitely guarantee the validity, accessibility and evolution of a project's files and calculations. With the aim of exploring topological optimization using only free code programs, this work presents algorithms and recent research, where the study and application of topological optimization in projects in sectors such as civil and aeronautical brought benefits, concluding in an implementation in which use only free and open source tools are used to carry out the design, modelling, simulation, optimization, and post-processing of a part, of which a pair was manufactured, original and relieved, that were then subjected to mechanical behavior tests, in order to document the process and validate the programs used to obtain satisfactory engineering results, offering a reliable alternative.

Keywords: topology optimization; python; free and open source software; computer-aided design; additive manufacturing.

LISTA DE FIGURAS

Figura 1 – Captura de tela de um projeto feito no FreeCAD	14
Figura 2 – Captura de tela de um projeto feito no OpenSCAD	15
Figura 3 – Captura da janela do calculix/beso no FreeCAD	17
Figura 4 – Exemplo de OT numa asa de avião	18
Figura 5 – Captura de tela de uma visualização e pós-processamento no ParaView	19
Figura 6 – Exemplos de tipos de otimização estrutural	20
Figura 7 – Representação da delaminação em peças fabricadas por manufatura aditiva polimérica, sob solicitação em relação à direção Z de adição de camadas	21
Figura 8 – Diagrama de operação simplificado do algoritmo BESO implementado	23
Figura 9 – OT nas nervuras do bordo de ataque de uma asa	25
Figura 10 – OT na estrutura sob o bico de uma fuselagem	25
Figura 11 – OT em um dissipador de calor por convecção natural	25
Figura 12 – Inspiração	26
Figura 13 – Representação da situação em estudo	27
Figura 14 – Peça objeto do estudo: o garfo de roda	29
Figura 15 – Parametrização	29
Figura 16 – Objeto Boolean <i>Fragments</i> que compõe os domínios	29
Figura 17 – Configuração das propriedades do PETG e atribuição do material	32
Figura 18 – Definição das forças	32
Figura 19 – Definição das condições nos apoios	33
Figura 20 – Todas as condições definidas	33
Figura 21 – Geração da malha	34
Figura 22 – Simulação inicial	34
Figura 23 – Diálogo de execução de <i>macros</i>	36
Figura 24 – Diálogo de configuração e execução da OT	36
Figura 25 – Progressão da otimização	37
Figura 26 – Resultados da otimização visualizados no ParaView	37
Figura 27 – Peças fabricadas	38
Figura 28 – Deformação prevista	39
Figura 29 – Montagem da bancada de testes	39
Figura 30 – Teste da bancada	40

Figura 31 – Ensaio da peça original	41
Figura 32 – Ensaio da peça otimizada	42

LISTA DE ACRÔNIMOS E SIGLAS

Acrônimos

BEM	método dos elementos de contorno, do inglês <i>Boundary Element Method</i>
BESO	otimização estrutural evolutiva bidirecional, do inglês <i>Bidirectional Evolutionary Structural Optimization method</i>
CAD	projeto auxiliado por computador, do inglês <i>Computer-Aided Design</i>
CAM	usinagem auxiliada por computador, do inglês <i>Computer-Aided Machining</i>
ESO	otimização estrutural evolutiva, do inglês <i>Bidirectional Evolutionary Structural Optimization method</i>
FEA	análise por elementos finitos, do inglês <i>Finite Element Analysis</i>
FEM	método dos elementos finitos, do inglês <i>Finite Element Method</i>
OT	Otimização Topológica

Siglas

ABS	polímero de Acrilonitrila Butadieno Estireno
CNC	comando numérico computadorizado, do inglês <i>Computer Numerical Control</i>
LSM	método dos conjuntos de nível, do inglês <i>Level-Set Method</i>
PETG	polímero de Polietileno Tereftalato de Etileno Glicol

LISTA DE SÍMBOLOS

LETRAS LATINAS

CF	coeficiente de falha	[adimensional]
$d_{\text{mínimo}}$	diâmetro mínimo	[mm]
F_A	carga atuante total na aterrissagem	[kg]
F_c	fator de carga de inércia	[adimensional]
F_H	força componente horizontal	[N]
F_R	módulo da força resultante	[N]
F_s	fator de segurança	[adimensional]
F_V	força componente vertical	[N]
g	aceleração da gravidade	[m/s ²]
P_t	massa total da aeronave	[kg]

LETRAS GREGAS

α	número de sensibilidade	[adimensional]
ρ_e	massa específica do elemento	[kg/m ³]
σ_a	tensão admissível do material	[MPa]
σ_m	tensão normal média no elemento	[MPa]
$\tau_{\text{admissível}}$	tensão de cisalhamento admissível	[MPa]
θ	ângulo de atuação de F_R	[°]

SUMÁRIO

1	INTRODUÇÃO	10
1.1	Contexto	10
1.2	Oportunidade	11
1.3	Objetivos	11
1.4	Motivação	12
1.5	Estrutura do trabalho	12
2	REVISÃO BIBLIOGRÁFICA	13
2.1	Programas de código livre	13
2.1.1	Modeladores paramétricos tridimensionais	13
2.1.1.1	<u>FreeCAD</u>	14
2.1.1.2	<u>OpenSCAD</u>	14
2.1.2	Geradores de malha e solucionadores numéricos de elementos finitos	15
2.1.2.1	<u>Gmsh</u>	15
2.1.2.2	<u>Netgen/NGSolve</u>	16
2.1.2.3	<u>CalculiX</u>	16
2.1.2.4	<u>PETSc</u>	16
2.1.3	Otimizadores topológicos	16
2.1.3.1	<u>calculix/beso</u>	16
2.1.3.2	<u>OpenLSTO</u>	17
2.1.3.3	<u>TopOpt_in_PETSc</u>	17
2.1.4	Visualizadores e pós-processadores	18
2.1.4.1	<u>ParaView</u>	18
2.1.4.2	<u>VisIt</u>	19
2.2	Otimização topológica	19
3	FUNDAMENTAÇÃO TEÓRICA	22
3.1	Algoritmos de otimização topológica	22
3.1.1	SIMP	22
3.1.2	LSM	22
3.1.3	BESO	23
3.2	Aplicações	24
4	MATERIAIS E MÉTODOS	26
4.1	Projeto	26
4.1.1	Premissas e definição de esforços e regiões	26

4.1.2	Modelagem	28
4.1.3	Cálculos	30
4.1.4	Otimização topológica	31
4.1.5	Pós-processamento e refinamento	35
4.2	Fabricação	35
4.3	Ensaio s	38
5	RESULTADOS	41
6	CONCLUSÕES E PERSPECTIVAS	43
	REFERÊNCIAS	44
	GLOSSÁRIO	47

1 INTRODUÇÃO

A otimização consiste em encontrar a melhor solução, quaisquer sejam as variáveis envolvidas. Pode-se dizer que otimizar é o essencial da engenharia, o espírito criativo que visa projetar e desenvolver com desempenho e economia (MARTINS; NING, 2021, p. 1).

A otimização estrutural em engenharia é o conjunto de técnicas utilizadas para encontrar a melhor combinação de geometrias e materiais numa construção que possa atender às condições de desempenho desejadas, visando maximizar sua resistência, rigidez e durabilidade, minimizando seu peso e custo, em uma ampla variedade de aplicações, desde as pontes, edifícios e templos feitos na antiguidade, até as carroças, automóveis, componentes de aeronaves, satélites, turbinas eólicas, e espaçonaves do tempo atual (DUNN, 2010).

Um desafio encontrado em qualquer projeto nas indústrias aeronáutica e aeroespacial, onde o menor peso ou volume possível são variável de interesse, reside em encontrar a peça mais eficiente e que entregará um desempenho estrutural máximo, criando uma solução capaz de resistir às solicitações e exigências específicas do projeto, poupando cada grama para isto. A Otimização Topológica (OT) é uma técnica de projeto em engenharia que visa minimizar o volume de uma peça, sem comprometer sua integridade. Consiste de um algoritmo que, com o cálculo por elementos finitos, feito de maneira iterativa, busca a melhor distribuição de material, dado um contorno e condições iniciais de esforços. Essa técnica é amplamente empregada em projetos de engenharia aeroespacial, automotiva e civil (BENDSØE; SIGMUND, 2013).

Hoje, as técnicas utilizadas para otimização estrutural se baseiam fortemente em programas de computador de análise por elementos finitos, do inglês *Finite Element Analysis* (FEA), que simulam o comportamento de estruturas em condições específicas, permitindo que os engenheiros experimentem diversas configurações até encontrar a mais eficiente. Os avanços na computação de alto desempenho e nos algoritmos admitem a otimização de estruturas mais complexas, com múltiplas variáveis (COUTINHO, 2006).

1.1 Contexto

A ampla difusão da computação e da conectividade criou em todos os setores a viabilidade de utilização do cálculo numérico computadorizado, que pode realizar análises, simulações e testes virtuais que permitem a identificação de pontos críticos que precisam ser aprimorados, além de possibilitar o estudo de efeitos complexos, garantindo componentes que funcionem de maneira segura e eficiente; um elemento importante da engenharia moderna (CHAPRA; CLOUGH, 2021, seq. 1.1).

“Independente da indústria, sua companhia agora é uma empresa de *software*, e pretender o contrário implica em um sério perigo. Com *hardware* e *software* ficando mais capazes a velocidades exponenciais, dados de todo tipo estão cada vez mais acessíveis ao usuário comum — competidores, empregados e, especialmente, clientes.”
(KIRKPATRICK, 2011)

Os programas de código livre, que se baseiam em um modelo colaborativo, onde usuários e desenvolvedores formam comunidades para aprimorar e atualizar os códigos-fonte

de forma descentralizada, são uma proposta para a construção do futuro, já que garantem a longevidade virtual tanto das aplicações quanto dos arquivos de um projeto. Por seu código ser aberto e acessível a qualquer pessoa para contribuições e melhorias, a comunidade de desenvolvedores pode continuar aprimorando e atualizando o *software* ao longo do tempo, sem depender de uma empresa ou de tecnologias de *hardware* específicas, que podem deixar de existir, ou se tornarem insuficientes, como no exemplo de programas que deixam de funcionar em computadores ou sistemas operacionais mais antigos. Além disso, a transparência do código aberto torna mais fácil para outros desenvolvedores diagnosticar e corrigir problemas, resultando em um programa mais estável e seguro a longo prazo (RAYMOND, 2001).

Por essas razões, os programas de código livre frequentemente têm uma vida útil indefinida, diferente de seus equivalentes proprietários, onde a empresa por trás do programa pode deixar de existir. Possuindo o código-fonte de toda a pilha de execução de um programa, desde as instruções do processador, a maneira como os arquivos são armazenados no *hardware*, o sistema operacional, e todos os programas correlatos ao projeto, garante-se que, mesmo em caso de catástrofe mundial, seja possível reconstruir a civilização. Utilizando este tipo de programa, profissionais podem ter acesso a recursos de alta qualidade além de contribuírem em seu desenvolvimento, cobrando por seus serviços na aplicação destas ferramentas (KON *et al.*, 2011).

1.2 Oportunidade

Utilizar e desenvolver técnicas e ferramentas computacionais avançadas está ligado ao melhoramento da eficiência, segurança e economia dos sistemas que identificam as soluções mais adequadas para os desafios cotidianos encontrados na engenharia, promovendo o desenvolvimento e a inovação, tanto resolvendo o problema quanto, no caso do código livre, aprimorando as ferramentas disponíveis.

1.3 Objetivos

Este trabalho tem como objetivo geral implementar a OT utilizando somente programas de código livre, a fim de documentar o processo e validar a capacidade dos programas utilizados de obterem resultados satisfatórios, na forma da peça otimizada suportar a mesma carga de sua contraparte original, de maneira que ficam delineados os objetivos específicos, abaixo:

- a) apresentar algoritmos e pesquisas recentes na área de OT;
- b) dimensionar uma peça inicial;
- c) implementar e documentar o processo utilizado nos programas para realizar o projeto, a modelagem, a simulação, a otimização, e o pós-processamento da peça otimizada;
- d) fabricar uma peça com a forma original, e outra aliviada pela otimização;
- e) submeter o par a um ensaio de comportamento mecânico;
- f) avaliar a capacidade da solução de resistir aos esforços predeterminados;
- g) inferir a capacidade de cálculo dos programas utilizados.

1.4 Motivação

Os programas de código livre são intimamente ligados à segurança, confiabilidade, inovação e posteridade. Tem como parte de seu papel a construção de uma sociedade melhor no futuro (RAYMOND, 2001), como a maneira mais eficaz de desenvolver pensando “7 gerações à frente” (SILVA, 2019). Ao compreender melhor os tópicos abordados neste trabalho, podemos criar soluções de engenharia mais eficientes, sustentáveis, manuteníveis e longevas, o que se espera, contribuirão para a melhoria da qualidade de vida e equilíbrio no Universo.

A OT tem a capacidade de produzir peças mais leves eliminando material desnecessário, o que pode levar a, por exemplo, uma redução nos custos de matéria-prima e transporte, consumo de combustível ou energia armazenada, mais espaço livre, e maiores eficiências aerodinâmica e de transferência de calor, obtendo peças de geometrias complexas, com curvas suaves e transições precisas, eliminando qualquer ponto concentrador de tensões com suscetibilidade à fadiga, o que pode ser difícil ou impossível de se obter sem o uso dessa técnica.

1.5 Estrutura do trabalho

O Capítulo 2 trata dos estudos e do estado da arte do tema, enquanto no Capítulo 3 estão delineados os conceitos dos algoritmos de cálculo numérico computadorizado utilizados. O Capítulo 4 documenta como foram executados o projeto, fabricação, e testes. Em conclusão, o Capítulo 5 traz a avaliação dos resultados, enquanto o Capítulo 6, palavras finais.

2 REVISÃO BIBLIOGRÁFICA

2.1 Programas de código livre

Os programas de código livre surgiram como uma alternativa ao modelo proprietário de *software*. Esse movimento teve como um dos principais idealizadores Richard Stallman, que criou a Free Software Foundation (FSF) em 1986. Stallman defende a liberdade de acesso e modificação do código-fonte dos programas, promovendo uma cultura de colaboração e compartilhamento do conhecimento (SILVEIRA, 2004).

Em 1991, Linus Torvalds publicou a primeira versão do sistema operacional Linux, um dos mais populares do mundo, utilizado em servidores e equipamentos onde a missão crítica e o tempo de operação são de suma importância, podendo operar anos a fio sem qualquer tipo de reinicialização nem recursos computacionais avançados; os servidores de comando e controle de telemetria da estação espacial internacional operam desde seu lançamento em processadores dos anos 1980 com 20 MHz e pouca memória, executando uma versão personalizada do sistema Linux (ORTEGA, 1999). O mesmo ocorre em diversas aplicações embarcadas desde automóveis (SIVAKUMAR; KUMAR; DEVI, 2022, p. 107) até roteadores de *internet*, tanto domésticos quanto da infraestrutura de espinha dorsal da rede.

A cada ano, o movimento de código livre e seu modelo de desenvolvimento colaborativo vêm crescendo e angariando mais adeptos, devido suas diversas vantagens em inovação e segurança. Ao utilizá-los, os usuários têm a liberdade de estudar, modificar e distribuir o código-fonte, o que significa que muitas pessoas podem trabalhar juntas para melhorar o *software* e encontrar soluções para problemas mais rapidamente. Além disso, oferecem maior segurança pois a comunidade de desenvolvedores pode encontrar e corrigir vulnerabilidades de forma mais eficaz. Oferecem mais transparência, o que significa que os usuários podem ver como o *software* está sendo desenvolvido e ter mais controle sobre suas informações e dados pessoais.

Nas subseções 2.1.1 a 2.1.4 estão listados alguns programas e algoritmos de código livre que podem ser utilizados para execução de projetos de engenharia e OT, com menção especial às ferramentas escolhidas para o desenvolvimento deste trabalho.

Estes programas são escritos de maneira a oferecerem interface e implementação facilitada de equacionamentos e algoritmos de solução diretamente, através de linguagens de programação como Python (ROSSUM; DRAKE, 2009), entre outras. Isso permite a utilização de todas as funcionalidades destas linguagens e suas vastas gamas de bibliotecas e projetos. Um exemplo seria utilizar uma rotina de coleta de dados em tempo real como entrada de parâmetros para o projeto ou a OT, que podem ser instruções provenientes de sensores, mensagens eletrônicas, etc.

2.1.1 Modeladores paramétricos tridimensionais

Programas modeladores de projeto auxiliado por computador, do inglês *Computer-Aided Design* (CAD) 3D são utilizados para criar objetos em formato digital, a partir de desenhos complexos e precisos, com uma variedade de ferramentas, gerando visualizações em tempo

real, ajudando na elaboração eficiente de projetos. Quando possuem habilidade paramétrica, permitem a definição de aspectos da geometria como funções e variáveis.

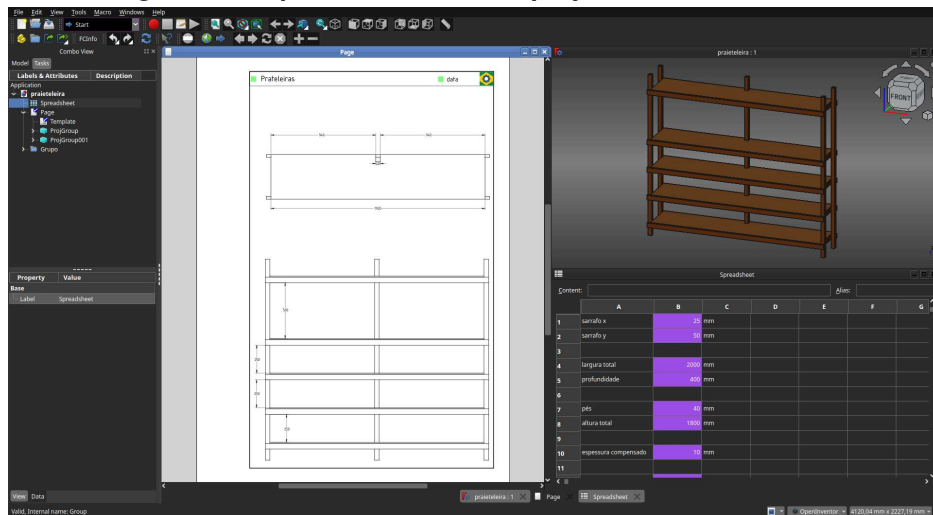
2.1.1.1 FreeCAD

FreeCAD (RIEGEL; MAYER; HAVRE, 2002) é um programa de CAD tridimensional paramétrico. Consiste de um núcleo codificado em C++, garantindo robustez e desempenho, que suporta as camadas externas, como as bancadas de trabalho, usadas para modelagem, desenho técnico tradicional, superfícies, arquitetura, impressão 3D, usinagem auxiliada por computador, do inglês *Computer-Aided Machining* (CAM) e comando numérico computadorizado, do inglês *Computer Numerical Control* (CNC), desenho de trajetórias de robôs industriais, simulação FEA, e muito mais.

Oferece também ferramentas fáceis para instalação e gerenciamento de bancadas adicionais e *macros* desenvolvidos pela comunidade de usuários, já que quase toda a comunicação entre o núcleo e a interface está codificada em Python, linguagem de programação flexível, amigável e fácil de aprender, utilizada para codificar, por exemplo, o *macro* de OT disponibilizado por Löffelmann (2017a), apresentado na subseção 2.1.3.1.

Em virtude de suas qualidades, este é o programa escolhido para o desenvolvimento deste trabalho. Na Figura 1 está um exemplo de um projeto realizado com o FreeCAD, onde a planilha controla os parâmetros, que quando alterados, são instantaneamente processados e aparecem tanto na janela de visualização tridimensional quanto na folha de desenho para produção.

Figura 1 – Captura de tela de um projeto feito no FreeCAD



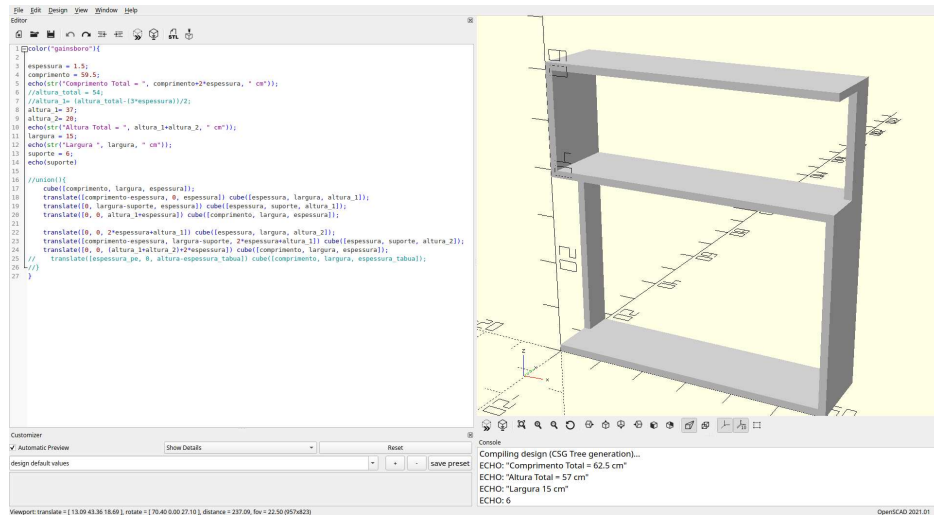
Fonte: Autoria própria (2024).

2.1.1.2 OpenSCAD

OpenSCAD (KINTEL; WOLF, 2011) é um programa para a criação de sólidos tridimensionais. Diferente de um modelador interativo, é baseado somente em *scripts* e usa sua própria

linguagem de descrição. Isso dá ao projetista controle completo sobre o modelo, que pode depender de funções para gerar características na peça, a partir de operações booleanas com sólidos primitivos, extrusão de esboços 2D, além de definir parâmetros como variáveis interdependentes, obtendo assim facilmente quaisquer configurações geométricas necessárias. O modelo pode ser exportado em formato .STL. Na Figura 2 está um exemplo de um projeto paramétrico de uma mesa de apoio, feito no OpenSCAD.

Figura 2 – Captura de tela de um projeto feito no OpenSCAD



Fonte: Autoria própria (2024).

2.1.2 Geradores de malha e solucionadores numéricos de elementos finitos

O método dos elementos finitos, do inglês *Finite Element Method* (FEM), é uma técnica utilizada na análise e resolução de modelos matemáticos, que podem ser baseados em equações diferenciais sem solução trivial. Consiste na discretização do domínio do problema em pequenos elementos finitos, onde cada um deles é descrito por uma função matemática mais simples, dividindo-o em pequenas partes mais simples de serem analisadas, representando as condições físicas em questão.

Programas geradores de malhas são usados para criar modelos discretizados de geometrias, compostos de pequenos elementos simples, por exemplo cubos ou pirâmides, que aproximam a geometria do objeto modelado original.

Solucionadores de elementos finitos são programas que permitem a aplicação de cálculos numéricos computadorizados em malhas de elementos finitos para realizar simulações. Dessa forma, é possível calcular e prever o comportamento das propriedades mecânicas, térmicas, elétricas ou químicas de um objeto complexo sob diferentes condições.

2.1.2.1 Gmsh

Gmsh (GEUZAINÉ; REMACLE, 1997) é um programa gerador de malha de elementos finitos tridimensional, com capacidades básicas de CAD e pós-processamento. Seu objetivo

é prover uma ferramenta de geração de malhas rápida, de uso fácil, com entradas paramétricas e um visualizador robusto. Seus módulos geométrico, gerador de malha, solucionador e pós-processador podem ser controlados a partir de uma interface gráfica ou do terminal de comandos, utilizando arquivos de texto escritos em sua linguagem de *script* .GEO, ou através de interfaces para as linguagens de programação C, C++, Python, Julia e Fortran. É uma das opções disponíveis para uso diretamente através de um botão no FreeCAD.

2.1.2.2 Netgen/NGSolve

Netgen (SCHÖBERL, 1997) é um programa gerador de malha de elementos finitos, oferecido como opção através de um botão no FreeCAD. Faz parte do projeto NGSolve (SCHÖBERL, 2000), que inclui um solucionador de FEA multifísica de alto desempenho, utilizado para analisar modelos de mecânica dos sólidos, dinâmica dos fluidos e eletromagnetismo. Possui uma interface flexível com linguagem Python.

2.1.2.3 CalculiX

CalculiX (DHONDT; WITTIG, 1998) é um programa de FEA que possui um solucionador implícito e explícito, além de um pré e pós-processador. O programa original foi desenvolvido para o sistema operacional Linux. Este é o programa que será utilizado no desenvolvimento deste trabalho, já que é o solucionador embutido no FreeCAD e que é usado pelo otimizador topológico escolhido (subseção 2.1.3.1).

2.1.2.4 PETSc

PETSc (BALAY *et al.*, 1991) é um solucionador de sistemas de equações diferenciais parciais, escalável e de alto desempenho, utilizado como biblioteca de FEA de um dos programas otimizadores topológicos disponibilizados pelo grupo de pesquisa TopOpt (subseção 2.1.3.3).

2.1.3 Otimizadores topológicos

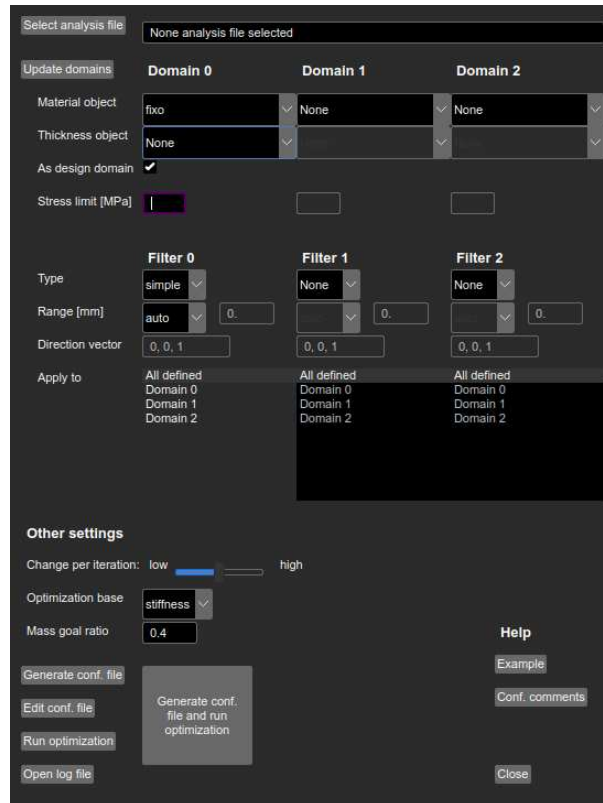
Os otimizadores topológicos são a classe de programas que tomam como entrada as malhas e os parâmetros de FEA determinados, para então realizar as operações iterativas que encontram dentro da malha a topologia mais eficiente.

2.1.3.1 calculix/beso

calculix/beso (LÖFFELMANN, 2017a) é um *macro* para o FreeCAD (Figura 3), que realiza a OT seguindo o algoritmo de otimização estrutural evolutiva bidirecional, do inglês *Bidirectional Evolutionary Structural Optimization method* (BESO), descrito na subseção 3.1.3. Pela facilidade de uso, é a ferramenta escolhida para o desenvolvimento deste trabalho. Permite a definição de regiões ou geometrias fora do domínio de otimização, como maneira de garantir

superfícies para pinos e rolamentos ou profundidades para roscas e assentamentos de parafusos, além de conseguir lidar com a simulação simultânea de até três materiais. Está disponível um parâmetro de filtragem com restrição para fabricação por fundição, e outras restrições estão em desenvolvimento. Além da fácil definição do objetivo de redução da massa, possui opção para otimização de fluxo de calor.

Figura 3 – Captura da janela do calculix/beso no FreeCAD



Fonte: Autoria própria (2024).

2.1.3.2 OpenLSTO

OpenLSTO (KAMBAMPATI *et al.*, 2018) é um programa em C++ para OT que utiliza o algoritmo LSM, descrito na subseção 3.1.2. O solucionador foi escrito para problemas de mecânica estrutural, mas os autores alegam que o código pode ser modificado facilmente, a fim de mudar a física do programa para resolver problemas de OT envolvendo transferência de calor, frequência natural, ou interação fluido-estrutural.

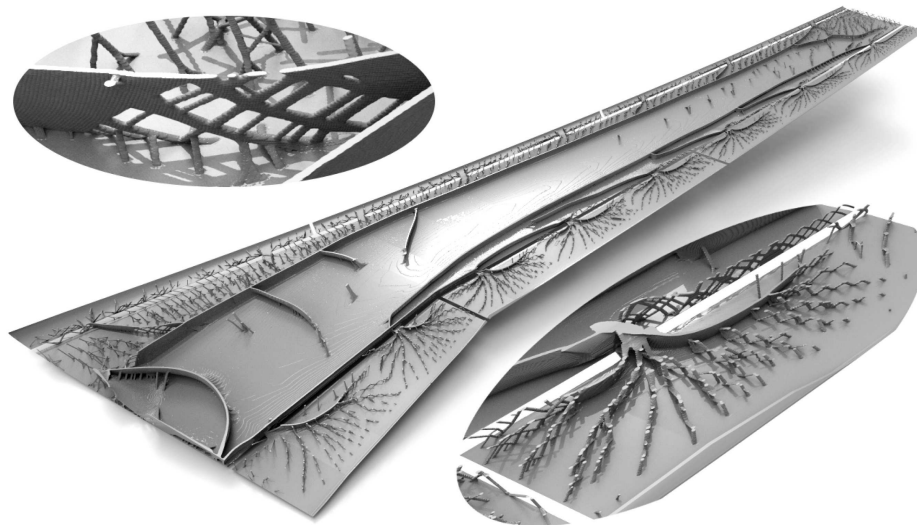
2.1.3.3 TopOpt_in_PETSc

TopOpt é um grupo formado pelos departamentos de Engenharia Mecânica e de Computação da Universidade Técnica da Dinamarca, que desenvolve pesquisas na área da OT, com o objetivo de promover avanço teórico e aplicação prática do método.

O grupo publicou uma implementação de OT utilizando a biblioteca PETSc (subseção 2.1.2.4). Um exemplo da capacidade do algoritmo está na Figura 4, onde a ferramenta foi

utilizada no projeto da estrutura interna de uma asa de avião em escala real. A asa otimizada possui detalhe estrutural sem precedente, com medidas variando entre dezenas de metros até o milímetro, e, curiosamente, mostra notável semelhança com as estruturas ósseas naturais de, por exemplo, bicos de pássaros. Estimaram que a estrutura otimizada representa uma redução de peso de 2–5% comparado a estruturas modernas comuns, o que se traduz em uma economia de consumo de combustível da nota de 40–200 toneladas por ano por avião. O processo pode ser aplicado de maneira geral, além do projeto mecânico, também a sistemas de fluxo, antenas, nano-óptica e micro-sistemas. (AAGE; ANDREASSEN; LAZAROV, 2014)

Figura 4 – Exemplo de OT numa asa de avião



Fonte: Aage et al. (2017).

2.1.4 Visualizadores e pós-processadores

Durante a progressão das iterações dos algoritmos de OT, são gerados vários arquivos, contendo malhas e diversas informações sobre as propriedades de seus elementos. A fim de interpretar estes dados, devem ser extraídos e manipulados, com um tipo de programa que permita a disponibilização gráfica de, por exemplo, tensões nos elementos, assim como definir planos de corte para analisar regiões interiores, além da capacidade de aplicar filtros, como de alisamento dos polígonos convexos nas superfícies.

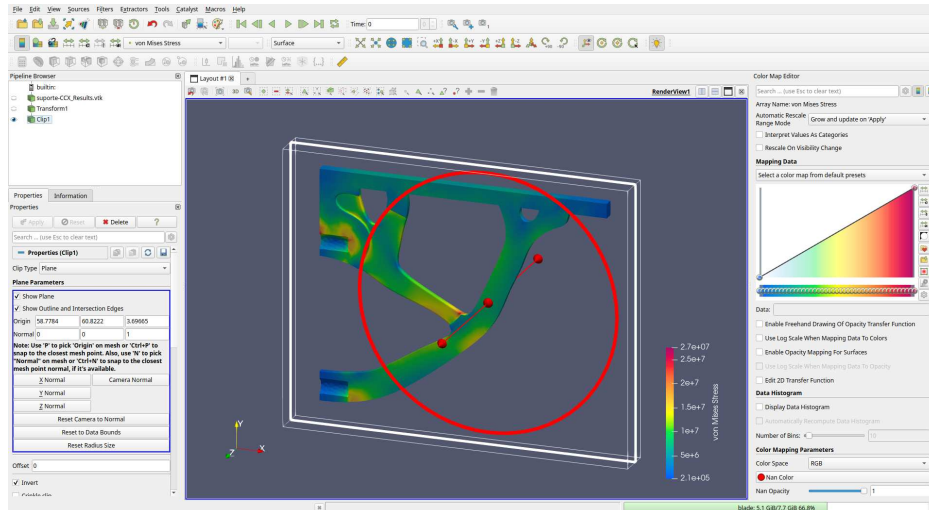
2.1.4.1 ParaView

ParaView (AHRENS; GEVECI; LAW, 2005) é um programa utilizado para visualização e análise de dados científicos em 2D e 3D. Suporta uma ampla gama de formatos de entrada de dados e oferece recursos avançados de análise estatística, pré e pós-processamento. Entre suas principais aplicações estão a visualização de dados geofísicos, simulações termofluidodinâmicas, análise de estruturas mecânicas e modelos estelares.

Com recursos e ferramentas de visualização de alta qualidade e em tempo real, é uma ferramenta poderosa para a análise de dados científicos em diversas áreas. Foi o programa

escolhido para a realização deste trabalho, e na Figura 5 está a visualização de uma simulação de esforços, com um plano de corte para apreciação dos detalhes nos furos e contornos internos de um suporte.

Figura 5 – Captura de tela de uma visualização e pós-processamento no ParaView



Fonte: Autoria própria (2024).

2.1.4.2 VisIt

VisIt (CHILDs *et al.*, 2012) é um programa de código aberto para conjuntos de dados simulados e experimentais em grande escala. Consiste de uma infraestrutura para pós-processamento geral e paralelizado de conjuntos de dados extremamente massivos. Os casos de uso alvo incluem exploração de dados, análise comparativa, depuração visual, análise quantitativa e gráficos de apresentação.

O VisIt reúne interface gráfica de usuário, linguagem de programação Python para um interpretador de linha de comando e muitos algoritmos de visualização, com capacidade de paralelização de processamento de grandes conjuntos de dados, na interface do usuário, na implementação de rotinas personalizadas de análise, e no tratamento de modelos de dados despadronizados, como malhas de refinamento adaptativas e zonas de materiais mistos.

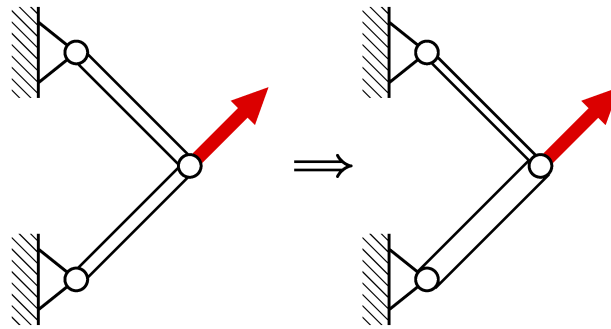
O projeto VisIt originou-se no Laboratório Nacional *Lawrence Livermore* como parte de um programa da Agência Nacional de Segurança Nuclear do Departamento de Energia dos Estados Unidos da América, mas tornou-se um projeto distribuído, sendo desenvolvido por vários grupos.

2.2 Otimização topológica

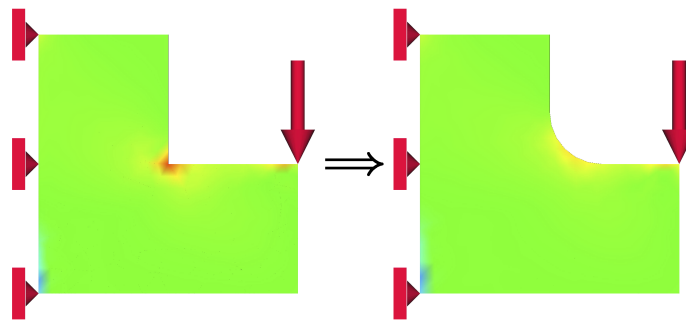
Segundo Bendsøe e Sigmund (2013), a otimização estrutural busca alcançar o melhor desempenho para uma estrutura que satisfaça várias restrições, e uma das variáveis é por exemplo determinar a quantidade de material. O projeto estrutural ideal torna-se cada vez mais importante devido aos recursos limitados, impacto ambiental e competição tecnológica, que

demandam equipamentos leves, de baixo custo e alto desempenho. Nas últimas três décadas, a disponibilidade de computadores de alta velocidade e a rápida melhoria em algoritmos usados para otimização de projetos transformaram o tópico de otimização estrutural, de interesse principalmente acadêmico, para o estágio atual, onde um número crescente de projetistas se beneficiam das técnicas de otimização. Tem havido cada vez mais atividades de pesquisa e desenvolvimento direcionadas para difundir este tipo de ferramenta para uso de forma fácil, confiável, eficiente e barata. As otimizações estruturais em engenharia são habitualmente classificadas em: *Dimensional*, que procura espessuras, comprimentos, diâmetros, áreas de seção transversal, etc. (Figura 6a); de *Forma*, que procura o contorno externo, a fim de, por exemplo, eliminar concentradores de tensão (Figura 6b); e *Topológica*, que procura a melhor distribuição de material dentro de um contorno (Figura 6c).

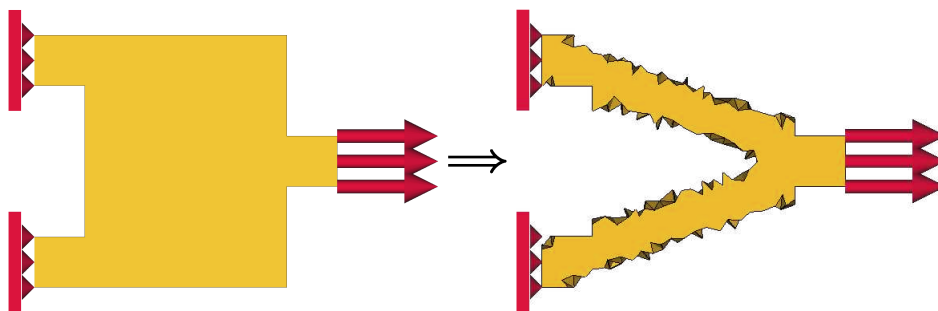
Figura 6 – Exemplos de tipos de otimização estrutural
(a) Exemplo de otimização dimensional em um par de treliças



(b) Exemplo de otimização de forma em um ponto concentrador de tensões



(c) Exemplo de OT em uma peça sob tração



Fonte: Autoria própria (2024).

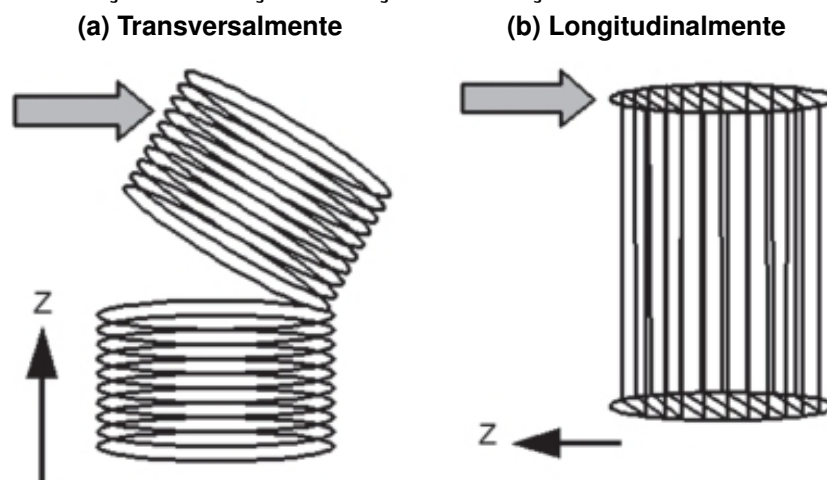
O problema básico resolvido pela OT consiste na busca pela distribuição ótima de material em uma estrutura linearmente elástica (BENDSØE; SIGMUND, 2013). A seção 3.1

apresenta uma breve revisão de alguns algoritmos mais populares para a resolução de casos de OT, com menção especial ao utilizado no programa com o qual será realizado o projeto proposto neste trabalho. Um excelente histórico da trajetória dos métodos de OT se encontra no trabalho de Coutinho (2006).

O trabalho publicado por Vatanabe *et al.* (2016) é um compêndio sobre as restrições que devem ser aplicadas aos algoritmos de OT para a obtenção de geometrias possíveis de serem fabricadas, principalmente, a necessidade de evitar vazios internos ou faces oblíquas, de maneira que impossibilite tanto a confecção de um molde de fundição, quanto a usinagem por ferramentas de corte comum ou especial, como eletroerosão ou fresadoras de CNC de cinco eixos. Verifica-se que a OT bidimensional é particularmente viável para o corte a laser e a jato de água. Ao encontrar a melhor distribuição de material, é possível então recortar as formas ótimas, a partir de chapas metálicas maciças. O mesmo pode ser dito da aplicação em peças que serão fabricadas por fundição, onde geometrias complexas podem ser obtidas em material metálico resistente (KÅMARK, 2018; ROSNITSCHKEK *et al.*, 2023).

A melhor maneira de fabricar uma geometria obtida através da OT é utilizando a manufatura aditiva. Como o processo depende da deposição de material em camadas, a delaminação é o limite mais evidente, como ilustrado na Figura 7, quando são utilizados os equipamentos mais comuns nos dias de hoje, que utilizam a deposição polimérica. A manufatura aditiva metálica é ainda proibitiva, por virtude da novidade da tecnologia, mas num futuro próximo, sem dúvidas será a técnica de fabricação mais utilizada em conjunto com a OT.

Figura 7 – Representação da delaminação em peças fabricadas por manufatura aditiva polimérica, sob solicitação em relação à direção Z de adição de camadas



Fonte: Adaptado de Volpato (2021).

3 FUNDAMENTAÇÃO TEÓRICA

3.1 Algoritmos de otimização topológica

De maneira geral, os algoritmos consistem na utilização de funções que determinam a quantidade de energia de deformação em cada elemento da malha, retirando os elementos de menor densidade de tensão, iterativamente. A diferença entre os métodos apresentados está em como eles encontram estes elementos, e se eles utilizam sempre a mesma malha, geram novas malhas a cada iteração, ou utilizam os contornos dos elementos para a resolução.

3.1.1 SIMP

O método de material sólido isotrópico com penalização, do inglês *Solid Isotropic Material with Penalization* (SIMP), foi proposto inicialmente por Bendsøe e Kikuchi (1988) e Rozvany, Zhou e Birker (1992). Consiste em discretizar o domínio de otimização em uma malha isotrópica de elementos finitos, onde todos os elementos possuem a mesma geometria e tamanho, o que chamou-se de microestruturas artificiais.

Para encontrar a melhor distribuição de material num dado domínio sob solicitações de esforços, cada elemento deve possuir densidade ρ_e igual a 0 (vazio) ou 1 (preenchido de material), sendo assim uma função discreta de distribuição de densidade. Então, é introduzido o conceito de atribuir a ρ_e uma função contínua de densidade relativa, onde $\rho_{\text{mínimo}} < \rho_e < 1$.

Desta maneira, pode-se definir um módulo de elasticidade relativo E ; ou seja, a propriedade mecânica que mede a rigidez de um material sólido surge como função das densidades relativas ρ_e . É introduzido um parâmetro de penalização de ρ, η , que diminui a contribuição de elementos com densidades intermediárias, fazendo-os tender mais fortemente para 0 ou para 1; além de E_0 , que representa o módulo de elasticidade do material sob densidade plena.

Assim, o algoritmo busca solucionar a FEA de maneira iterativa, com objetivo de encontrar as densidades dos elementos que minimizem a deformação da estrutura, que é função de E . O conjunto de densidades resposta conterà a melhor distribuição de material para a malha.

3.1.2 LSM

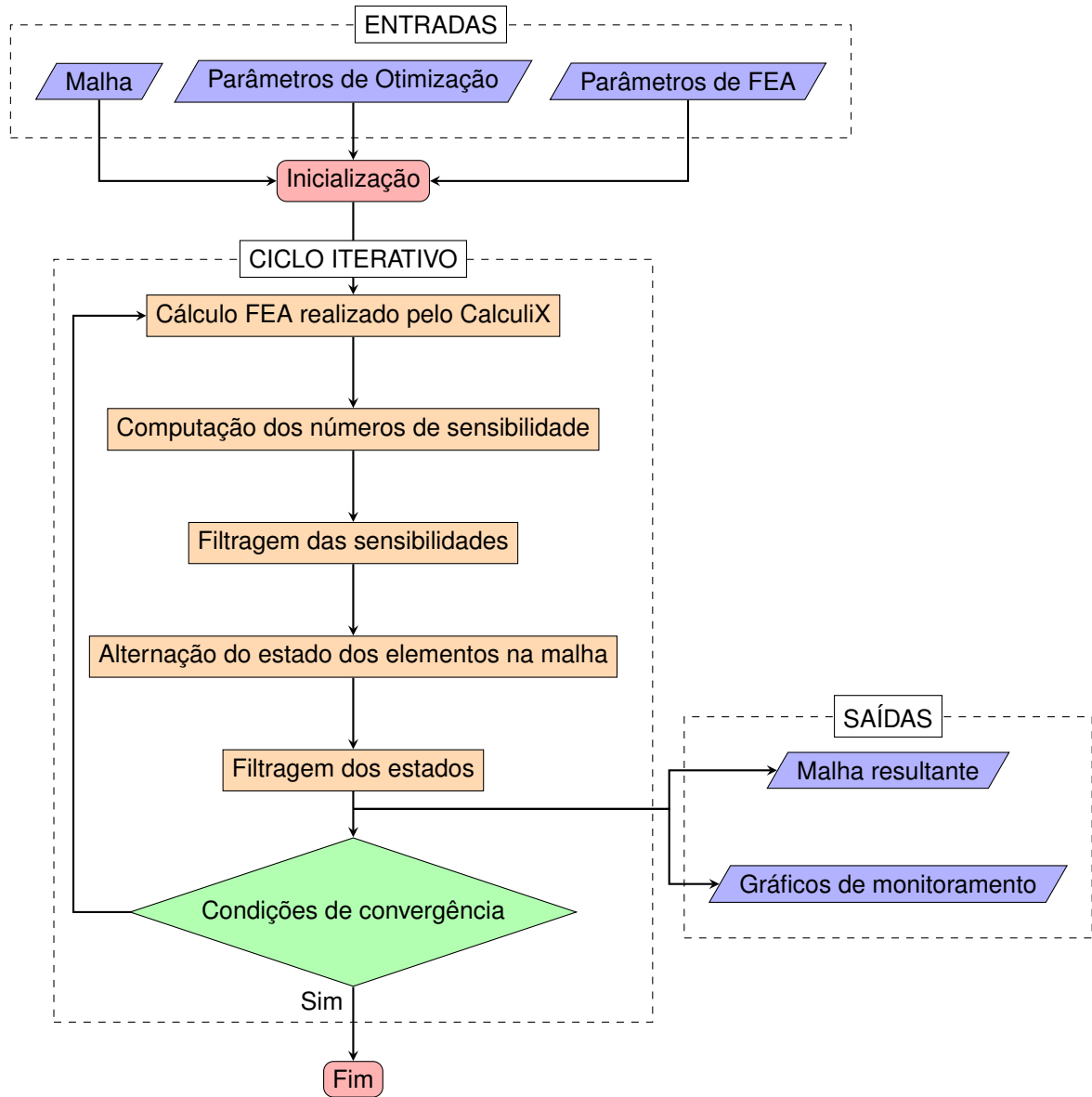
Diferente do método descrito anteriormente, o método dos conjuntos de nível, do inglês *Level-Set Method* (LSM), é uma técnica de OT que está acoplada ao método dos elementos de contorno, do inglês *Boundary Element Method* (BEM); utiliza a parametrização de uma superfície que descreve os esforços de uma geometria através de uma função escalar de menor dimensão, onde o gradiente entre curvas de nível encontra a variação na topologia dos elementos que sofrem esforços dentro da malha, a fim de filtrar aqueles que menos contribuem para a estabilidade elástica estrutural do conjunto, encontrando assim a melhor distribuição de material (WANG; WANG; GUO, 2003).

3.1.3 BESO

O método de BESO, é um algoritmo de OT acoplado ao FEM, que, de maneira iterativa, remove os elementos menos eficientes da malha. Estes elementos podem ser recolocados, caso a solução se torne instável algumas iterações depois, e esta é a principal diferença entre otimização estrutural evolutiva, do inglês *Bidirectional Evolutionary Structural Optimization method* (ESO) e BESO.

Este é o algoritmo utilizado no otimizador topológico escolhido para este trabalho. A implementação de Löffelmann (2017b) é uma variação do BESO publicado por Huang e Xie (2010), cujo esquema de funcionamento pode ser apreciado na Figura 8.

Figura 8 – Diagrama de operação simplificado do algoritmo BESO implementado



Fonte: Adaptado de Löffelmann (2017b).

Realizado o primeiro cálculo de FEA para simular a resposta mecânica da estrutura, são determinados os números de sensibilidade α de cada elemento da malha, computados segundo a Equação 1:

$$\alpha = \frac{CF}{\rho_e}, CF = \frac{\sigma_m}{\sigma_a} \quad (1)$$

Onde ρ_e é a massa específica do elemento e CF seu *coeficiente de falha*, equivalente à tensão normal média no elemento σ_m sobre a tensão normal admissível do material σ_a .

De maneira iterativa, o algoritmo altera o estado dos elementos (“ligado” ou “desligado”, “presente” ou “ausente”) com menor número de sensibilidade, a fim de reduzir áreas de baixa e aumentar áreas de alta densidade de tensão, diminuindo assim a massa da malha, removendo gradualmente os elementos que menos contribuem para a estabilidade global. Esta característica de alteração de estado é o que define um grande ganho de eficiência do BESO em relação a métodos que recorrem à geração de uma nova malha a cada iteração, o que aumenta o custo computacional.

A FEA é recalculada para a nova configuração da malha. Se o desempenho da estrutura tiver melhorado, a nova topologia é mantida, e novos elementos são retirados; caso contrário, a iteração anterior é desfeita em parte, e outros elementos são selecionados para serem retirados.

O processo iterativo cessa quando o objetivo redução de massa for atingido, ou se $CF \geq 1$ por 5 iterações seguidas.

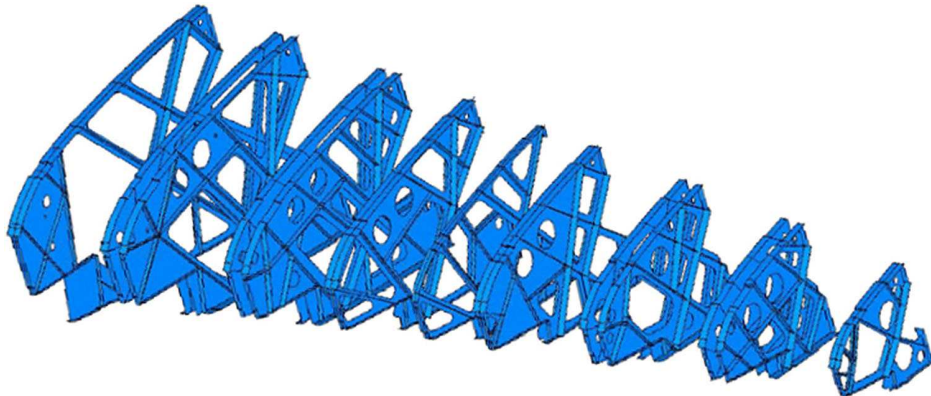
Diversos filtros foram implementados a fim de prevenir o efeito “tabuleiro de xadrez”, ou a presença de vazios no interior da malha. Outras restrições estão previstas para revisões futuras do programa.

3.2 Aplicações

A OT é particularmente interessante quando aplicada em peças onde a eficiência estiver na redução de peso. No caso de aparelhos voadores, cada grama tem valor real na autonomia. Alguns exemplos de aplicações da OT na aeronáutica estão nas nervuras do bordo de ataque de uma asa, ou na estrutura sob a fuselagem do bico, ilustrados nas figuras 9 e 10.

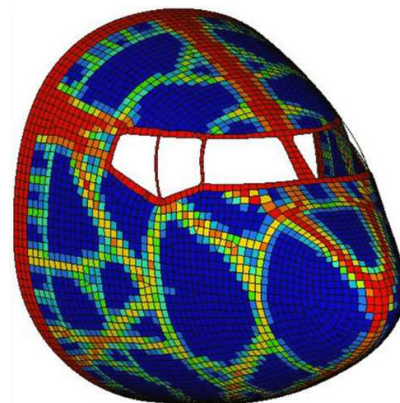
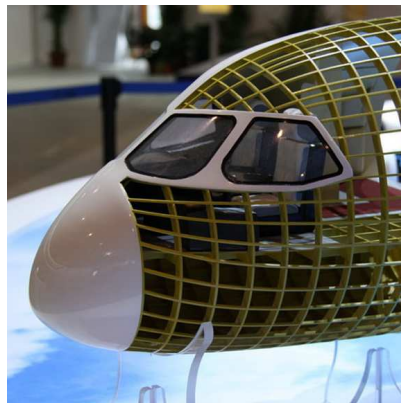
Pesquisas recentes estão voltadas à utilização do método de OT na solução de sistemas termofluidodinâmicos, como ilustrado na Figura 11. O estudo realizado determinou que um dissipador de calor por convecção natural projetado com OT obteve o mesmo desempenho que um dissipador comercial com o dobro do tamanho e peso.

Figura 9 – OT nas nervuras do bordo de ataque de uma asa



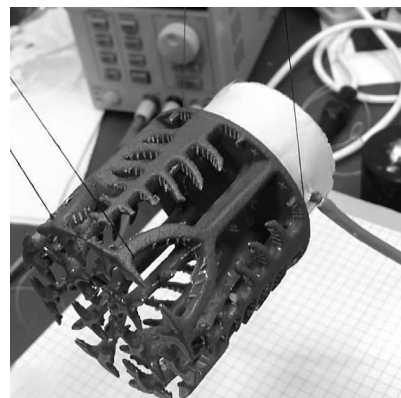
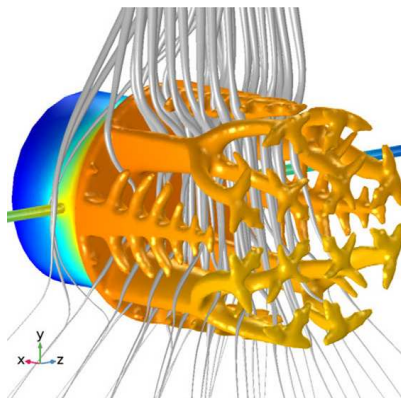
Fonte: Zhu, Zhang e Xia (2015).

Figura 10 – OT na estrutura sob o bico de uma fuselagem
 (a) Modelo convencional (b) Estrutura obtida por OT



Fonte: Adaptado de Zhu, Zhang e Xia (2015).

Figura 11 – OT em um dissipador de calor por convecção natural
 (a) Simulação (b) Teste em bancada



Fonte: Adaptado de Lazarov *et al.* (2018).

4 MATERIAIS E MÉTODOS

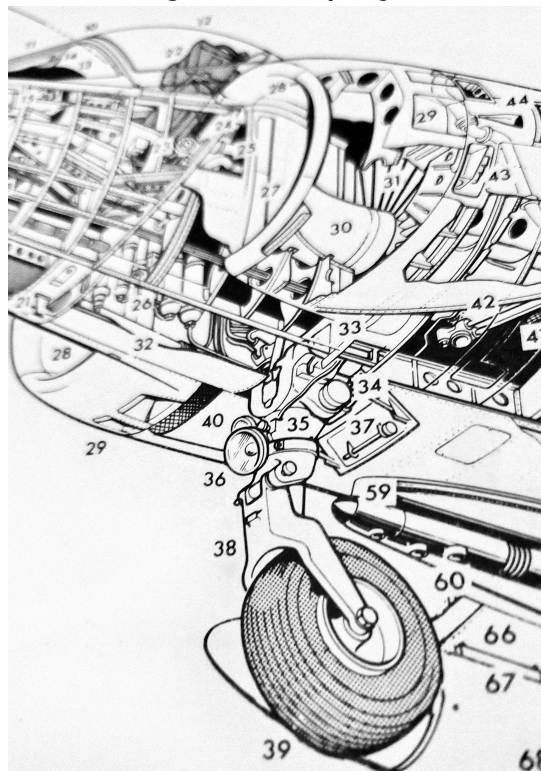
Este capítulo apresenta os materiais utilizados e os procedimentos realizados em cada etapa do estudo, que consistiu do projeto, da fabricação e dos ensaios.

4.1 Projeto

Na execução do projeto, foi empregado um computador portátil com processador de 4,3 GHz, 16 GB de memória RAM, e 2 GB de espaço em disco, sendo 1 GB para os arquivos do projeto e 1 GB para os programas FreeCAD v0.21.2, e ParaView v5.12.1.

O projeto para otimização topológica começa com a escolha da peça e a definição das proposições iniciais a respeito de seu conjunto, função e cargo na operação do sistema. Tomando inspiração na Figura 12, foi escolhida como peça objeto do estudo o garfo de roda do conjunto do trem de pouso frontal de um aeromodelo, por ser uma das peças maciças que atendem a uma solicitação importante e onde pode haver otimização de massa.

Figura 12 – Inspiração



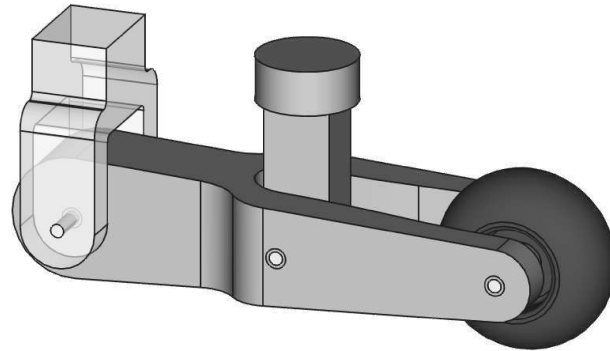
Fonte: Espilotro (1991).

4.1.1 Premissas e definição de esforços e regiões

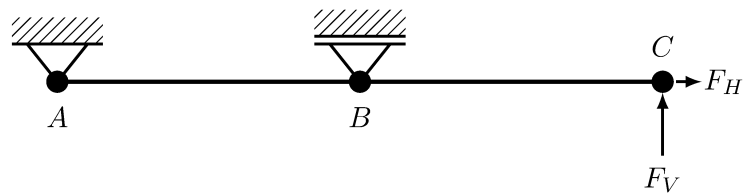
Consultando a revisão bibliográfica, é proposto que a configuração do trem de pouso seja triciclo, com duas rodas principais posteriores ao centro de gravidade da aeronave e uma auxiliar em frente (ABELING, 2016).

A situação de pouso avaliada é a em três rodas, no momento em regime permanente onde o amortecedor está em fim de curso, quando fica o garfo paralelo ao solo, como representado na Figura 13, onde o apoio A é o pino do eixo de rotação, o apoio B o pino do eixo do amortecedor, e C o pino do eixo da roda, onde a força é aplicada.

Figura 13 – Representação da situação em estudo
(a) Montagem



(b) Diagrama de corpo livre



Fonte: Autoria própria (2024).

Considerando as Equações 2 a 4, a estimativa dos esforços vertical F_V e horizontal F_H na roda do trem de pouso dianteiro começa com a determinação da força atuante F_A :

$$F_A = gP_t F_c F_s = 550 \text{ N} \quad (2)$$

$$F_V = 0,15F_A = 82,5 \text{ N} \quad (3)$$

$$F_H = 0,40F_V = 33 \text{ N} \quad (4)$$

onde: F_A = carga atuante total na aterrissagem

g = aceleração da gravidade, $9,81 \text{ m/s}^2$

P_t = massa total da aeronave, definida em 14 kg para este projeto

F_c = fator de carga de inércia, adimensional, recomendado como 2,67

F_s = fator de segurança, adimensional, recomendado como 1,5

F_V = componente vertical na roda dianteira

F_H = componente horizontal na roda dianteira

Conhecidas as componentes ortogonais da força, infere-se:

$$F_R = \left| \sqrt{F_V^2 + F_H^2} \right| = 89 \text{ N} \quad (5)$$

$$\theta = \arctan \frac{F_V}{F_H} = 68^\circ \quad (6)$$

onde: F_R = módulo da força resultante na roda

θ = ângulo de atuação de F_R

Sabendo que no pino da roda atua $F_R = 89 \text{ N} \angle 68^\circ$, determina-se o diâmetro deste componente, com a Equação 7 (HIBBELER, 2022):

$$d_{\text{mínimo}}[\text{mm}] = \sqrt{\frac{F_R}{\tau_{\text{admissível}} \cdot 4 \cdot \pi}} \cdot 1000 \approx 2 \text{ mm} \quad (7)$$

onde: $d_{\text{mínimo}}$ = diâmetro mínimo do pino

$\tau_{\text{admissível}}$ = tensão de cisalhamento admissível para o material, aço 1020 com 330 MPa

Pela facilidade de utilizar como pinos pregos de bitola 15x21, fica definido o diâmetro do pino das rodas como 3 mm, que todos os pinos possuem o mesmo diâmetro, e que a espessura das cascas cilíndricas da camada que os envolve, que devem ser mantidas inalteradas pelo processo de otimização, como maneira de garantir que fiquem intactas as superfícies de assentamento, fica parametrizada na planilha da modelagem como função da metade do diâmetro dos pinos.

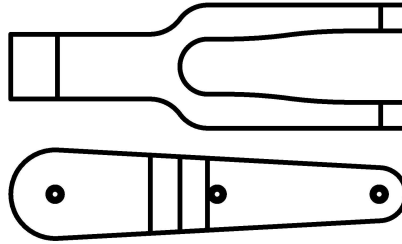
4.1.2 Modelagem

A modelagem da solução consiste do desenho tridimensional paramétrico e de um cálculo de FEA inicial, para garantir que a peça resiste ao esforço predeterminado, antes de seguir para a otimização. No FreeCAD, após iniciar um novo arquivo e salvá-lo, a ilustração da inspiração (Figura 12) foi importada com a opção `File→Import` e utilizada subposta no modelador, para auxiliar no desenho da forma da peça da Figura 14a, como ilustrado na captura de tela da Figura 14b.

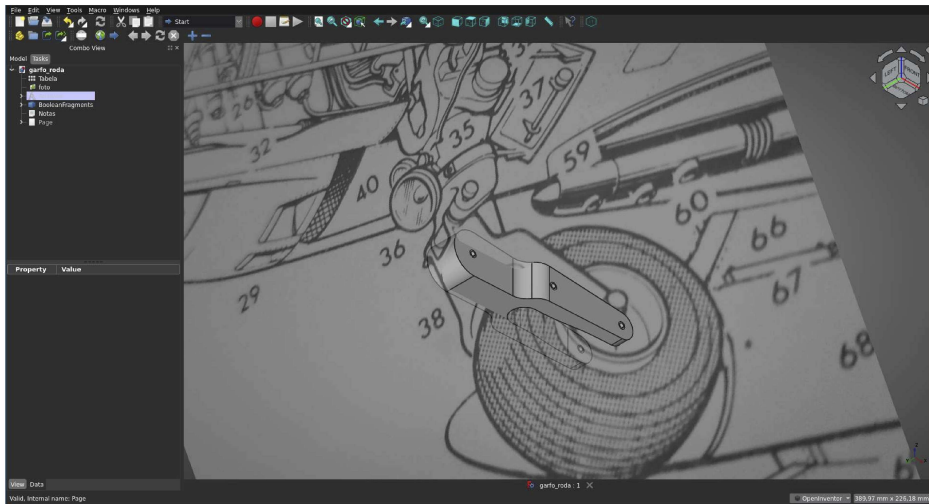
Foi criada uma planilha na bancada `Spreadsheet`, em `Spreadsheet→Create spreadsheet`, então inseridos os valores iniciais das dimensões da roda, dos pinos, das dimensões gerais, e da espessura das cascas das regiões inalteradas, de maneira que em conjunto com a bancada `Part Design` estas medidas estejam referenciadas nos desenhos, como ilustrado da captura de tela da Figura 15.

As cascas que definem as regiões inalteradas são modeladas separadamente do corpo principal da peça, que tem um vazio correspondente, então agregadas num objeto composto, a partir da bancada de trabalho `Part`, selecionando todos os elementos e então a opção `Part→Split→Boolean fragments`, como ilustrado na Figura 16.

Figura 14 – Peça objeto do estudo: o garfo de roda
(a) Modelo

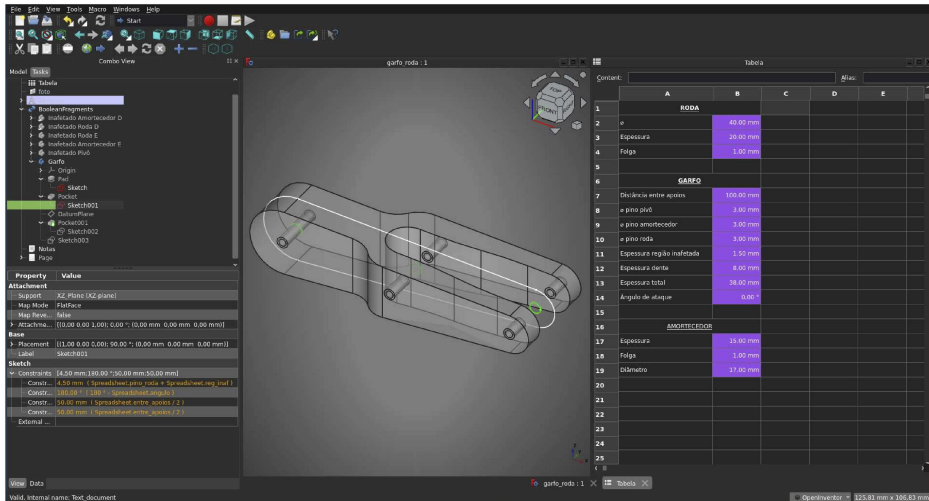


(b) Gabarito



Fonte: Autoria própria (2024).

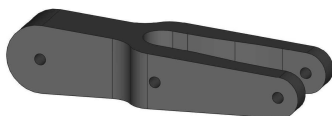
Figura 15 – Parametrização



Fonte: Autoria própria (2024).

Figura 16 – Objeto Boolean Fragments que compõe os domínios

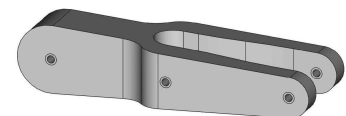
(a) Peça



(b) Regiões inalteradas



(c) Boolean Fragments



Fonte: Autoria própria (2024).

4.1.3 Cálculos

Partindo para a prova inicial de que a peça gerada com os parâmetros atuais consegue resistir aos esforços, é criada a análise, na bancada FEM, na opção `Model→Analysis container`. Começando pela definição do material, foi escolhido para o processo de manufatura aditiva o filamento de PETG, que possui boa maleabilidade, refletida em seu alto coeficiente de Poisson, além de retrair pouco durante o esfriamento, evitando deformações nas regiões mais finas da peça aliviada, como acontece com o ABS, por exemplo.

Na versão atual do programa, dentre os materiais predefinidos na bancada FEM, falta uma definição de material (*material card*) para o PETG, logo, foi criado um arquivo de caracterização .FCMat, com dados retirados da pesquisa de Silva (2018): densidade de 1850 kg/m^3 , módulo de Poisson de 0,32, módulo de elasticidade (Young) de 1560 MPa (Listagem 1), e importado na opção `Model→Materials→Materials for solid⇒FEM material→Editing material→use FreeCAD material editor⇒Open→PETG.FCMat→OK`.

Listagem 1 – Definição do material PETG na forma do arquivo PETG.FCMat

```

1 ; PETG
2
3 [General]
4 Name = PETG
5 Description = PolyEthylene Terephthalate Glycol-modified
6 Father = Thermoplast
7
8 [Mechanical]
9 Density = 1850 kg/m^3
10 PoissonRatio = 0.32
11 YoungsModulus = 1560 MPa

```

Fonte: Autoria própria (2024).

Os cálculos de elementos finitos realizados a seguir consideram o material como isotrópico e homogêneo. Foi assumido que desde que seja respeitada a direção longitudinal entre a força resultante e a direção de deposição do material durante a fabricação, utilizando o valor do módulo de elasticidade retirado da bibliografia, obtido em método experimental com corpos de prova feitos do mesmo material, em processo de fabricação e condição de carga semelhante, os interstícios e vazios característicos do processo de impressão 3D e sua influência podem ser desprezados.

Então, é atribuído o material à peça, sem as regiões inalteradas, em `Geometry reference selector for a Solid, Face, Edge→Selection mode→Solid→Add`, selecionar o corpo correspondente, e OK. O elemento foi renomeado de `MaterialSolid001` para PETG. Em seguida, foi repetido o procedimento, mas selecionando os corpos das regiões inalteradas (Figura 17), e renomeando de `MaterialSolid002` para Inalterado. Esta separação na atribuição do material no contenedor da análise é como o algoritmo de otimização implementado diferencia entre as regiões que serão removidas e quais permanecem inalteradas.

O passo seguinte é definir F_V , em `Model→Mechanical Constraints→Constraint force⇒Add`, selecionados os dois cilindros que apoiam o pino C , inserido

o valor da força, então a opção `Direction`→OK, selecionada uma aresta da peça que aponte verticalmente, novamente a opção `Direction` e então `Reverse direction` (Figura 18a). O procedimento é repetido para F_H , mas sem a necessidade de inserir a direção, pois o valor padrão já é apropriado (Figura 18b).

São então definidos os apoios dos pinos A e B , que podem rotacionar ao redor de seus eixos de centro, na direção y , e impedidos de transladar em qualquer direção. Inserir primeiro a fixação do eixo y para o cilindro A , em `Model`→`Mechanical Constraints`→`Constraint displacement`⇒`Add`, selecionar a face interna do cilindro, marcar a opção `Displacement y`→`Fixed` e OK, então, repetir o processo para os dois cilindros do apoio B (Figura 19a). Em seguida, definir para cada um dos três cilindros suas respectivas condições de rotação, em `Model`→`Geometrical Constraints`→`Constraint transform`⇒`Cylindrical transform`, selecionar a face interna do cilindro, `Add`→OK (Figura 19b). Ficam definidas assim todas as condições de contorno iniciais para o cálculo de elementos finitos, como ilustrado na Figura 20.

Foi gerada a malha de elementos finitos pela interface direta do FreeCAD com o Gmsh, selecionando o objeto `Boolean Fragments` e a opção `Mesh`→`FEM mesh from shape by Gmsh`, utilizando os parâmetros como na figura Figura 21, então `Apply` e OK.

Finalmente, é executada uma simulação numérica, garantindo que a forma inicial pode suportar os parâmetros, adicionando o solucionador, em `Solve`→`Solver CalculiX Standard`⇒`Write .inp file`⇒`Run CalculiX` como ilustrado da captura de tela da Figura 22a, que tomou 3,3 s para ser computada.

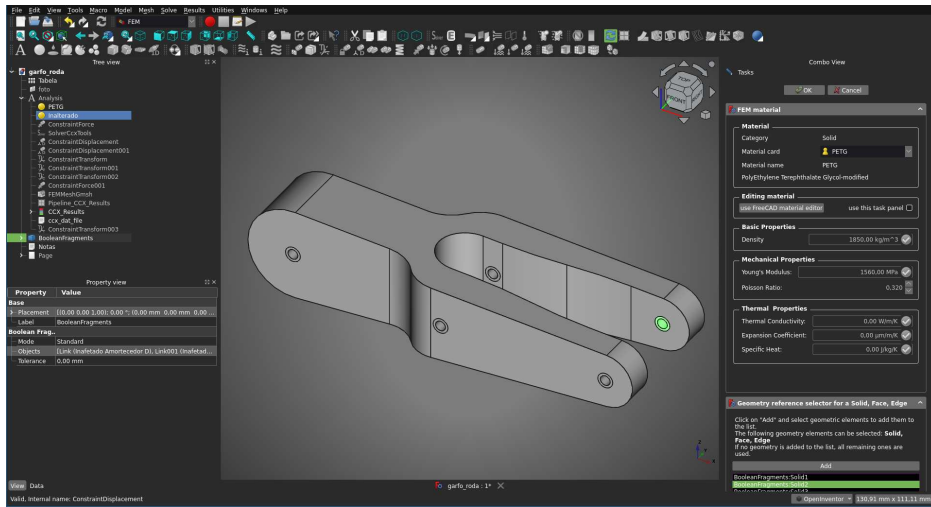
Para visualizar os resultados, selecionar com duplo clique o item `CCX_Results` e então `Result type`→`von Mises Stress`, onde pode ser observado, como na Figura 22b, a tensão máxima na ordem de 6,3 MPa, menos de $\frac{1}{4}$ da tensão máxima de escoamento, discutida abaixo, e também a deformação esperada. Confirmado que a geometria da peça está pronta para a otimização, é exportada para fabricação da peça inicial, selecionando o objeto `Boolean Fragments` e então `File`→`Export`, com a opção `Files of type`: definida para arquivos `.STL`.

4.1.4 Otimização topológica

A otimização foi computada utilizando o `macro calculix/beso`, citado na subseção 2.1.3.1. Para acessar a função, adquirir os arquivos em Löffelman (2017a) (opção `Code`→`Local`→`Download ZIP`), extrair numa pasta, e executar no FreeCAD com `Macro`→`Macros ...`⇒`User macros location`:→`...`, selecionar a pasta onde foram extraídos os arquivos, então selecionar `beso_fc_gui.py`→`Execute` (Figura 23).

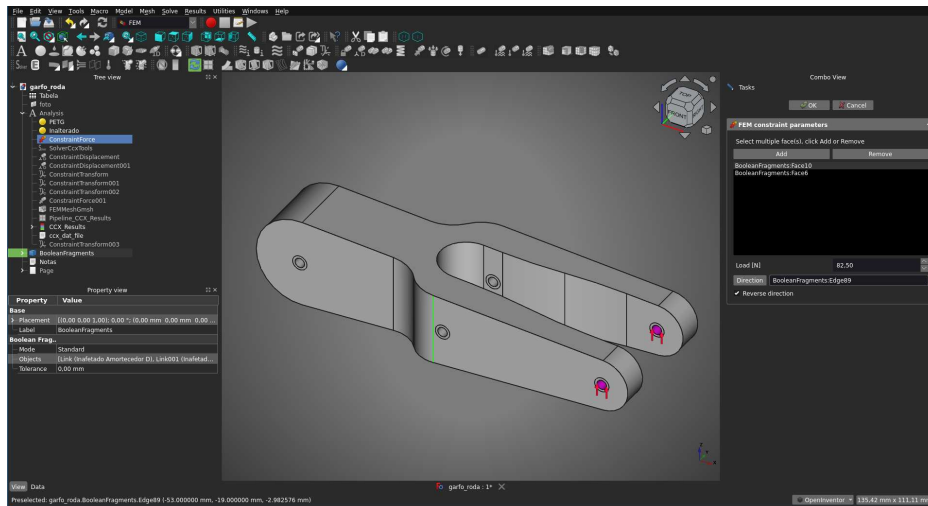
Agora, na janela de execução da otimização (Figura 24), selecionar o arquivo `FEMMeshGmsh.inp` criado na execução da análise inicial, em `Select analysis file`. O programa coloca automaticamente o `Material object`→`Domain 0`⇒`PETG` como `As design domain`. Basta definir a região inalterada, em `Material object`→`Domain 1`⇒`Inalterado` e desmarcar `As design domain`→`Domain 1`. Inserir para ambos os domínios a tensão de limite de escoamento do material, que foi definida conforme uma média conser-

Figura 17 – Configuração das propriedades do PETG e atribuição do material

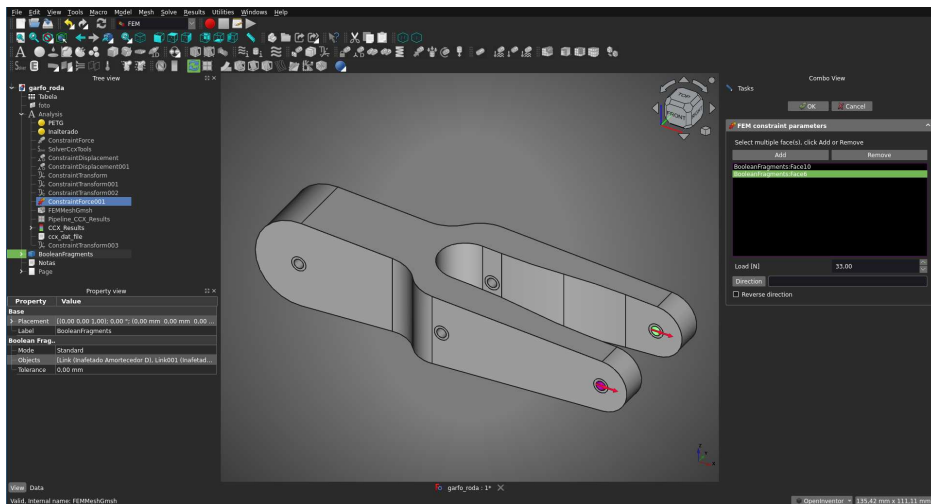


Fonte: Autoria própria (2024).

Figura 18 – Definição das forças
(a) Força vertical

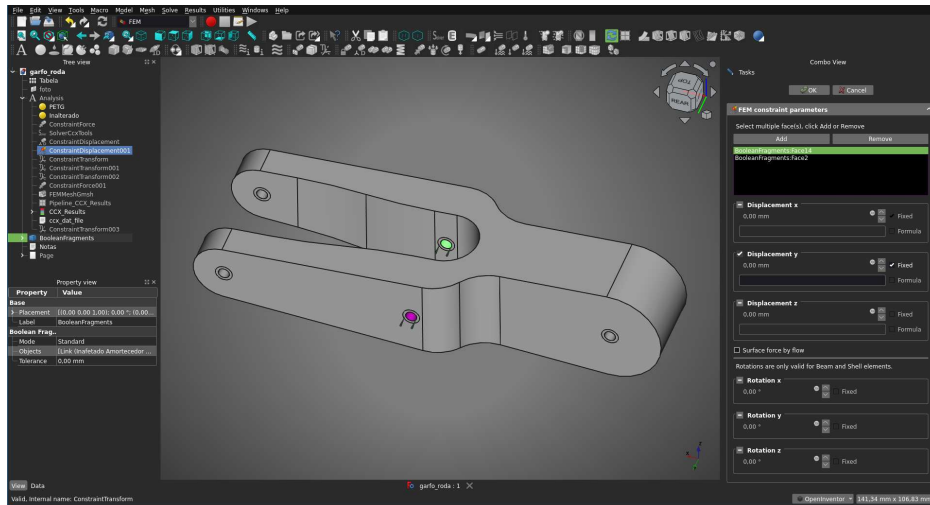


(b) Força horizontal

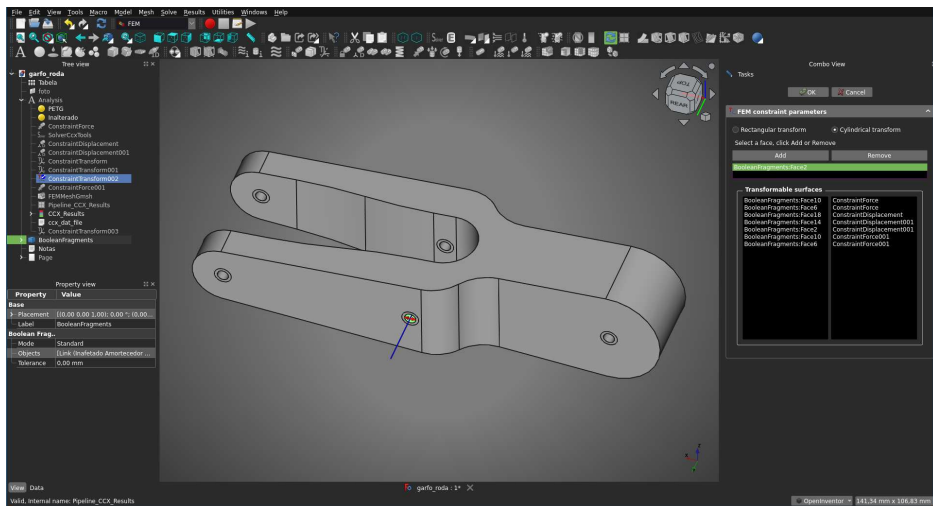


Fonte: Autoria própria (2024).

Figura 19 – Definição das condições nos apoios
(a) Fixação do eixo y no apoio B

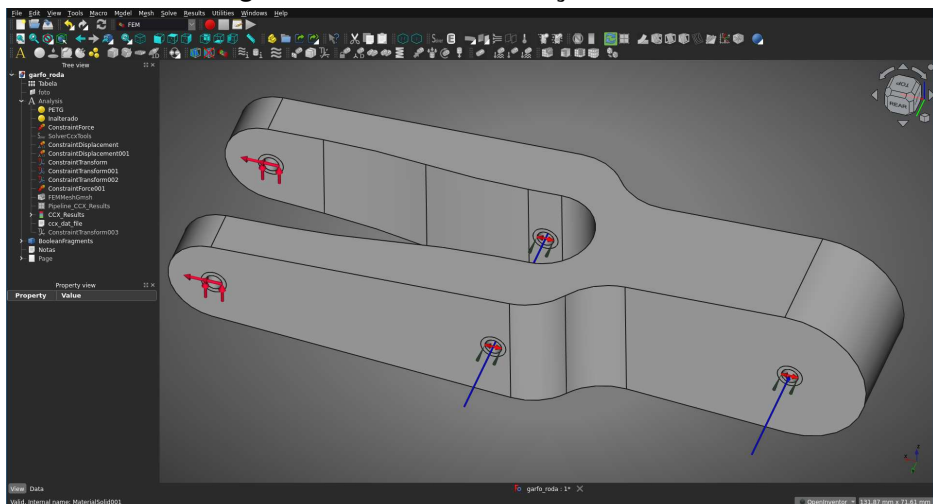


(b) Translação nos cilindros do apoio B



Fonte: Autoria própria (2024).

Figura 20 – Todas as condições definidas



Fonte: Autoria própria (2024).

vadora, utilizando valores retirados de pesquisa bibliográfica, provenientes da pesquisa de Silva (2018), de 26 MPa. O parâmetro `Mass goal ratio` foi de 0,3, ou seja, o otimizador buscará uma solução onde a peça final tem 30% da massa original. A otimização topológica foi computada, em `Generate conf. file and run optimization`.

Os resultados da otimização ficam registrados no arquivo `FEMMEshGmsh.log`, de onde foram extraídos os dados ilustrados na Figura 25, onde podem ser observados durante a progressão das iterações o decaimento da massa total e o aumento da densidade média de tensão nos elementos. O processo durou 18 min 29 s e foi executado em 65 passos. Foi obtida redução de massa de 65%, partindo de 92 g e chegando em 32 g, e aumento do coeficiente de falha em 44%, com a utilização teórica da capacidade de carga da peça subindo de 29% para 42%.

4.1.5 Pós-processamento e refinamento

A partir deste ponto, é utilizado o ParaView. Ao iniciar o programa, abrir os arquivos da solução em `File→Open⇒file...vtk`, na pasta `SolverCcxTools` onde foi executada a otimização, e então em `Properties→Apply`. Os filtros são aplicados a partir da opção `Filters→Alphabetical`. O primeiro filtro é o `Transform`, com `Rotate` igual a -90 no eixo x , já que no ParaView o eixo y aponta para cima por padrão, enquanto no FreeCAD é o z . Feito isso, aplica-se o filtro `Threshold`, com as propriedades `Scalars` como `element_states`, e `Threshold Method` em `Above Upper Threshold`, que é 1, para deixar apenas os elementos com estado “ligado” fazendo parte da análise.

Então, avançar para a última iteração em `Time:`, adicionar o filtro `Clip`, com `Clip Type` em `Scalar`, onde `Scalars` são `element_states_averaged_at_nodes` e `Value` no valor médio (ícone das flechas em círculo). Adicionar os filtros `Extract Surface`, que permite as próximas operações, e `Triangulate`, que simplifica a malha, para então passar ao último filtro, `Smooth`, com `Number of Convergence` igual a 31, que foi um valor encontrado empiricamente.

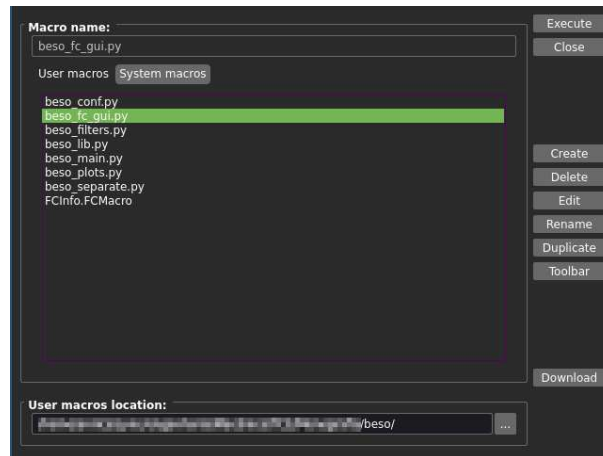
O resultado final pode ser observado com as tensões de von Mises ao selecionar `Properties→Coloring→FI=stress_von_Mises/26.0` (Figura 26a). A progressão do alívio de peso nas iterações intermediárias 0, 17, 34 e 51 está ilustrada na Figura 26b. A superfície final é exportada em `File→Save Data...`, onde `Files of type:` é selecionado o formato `.STL` e o arquivo salvo para envio à fabricação da peça aliviada.

4.2 Fabricação

As peças foram fabricadas sob encomenda com empresa especializada, pelo processo de manufatura aditiva por extrusão polimérica. Foram produzidas duas peças, uma original e uma aliviada (Figura 27), com filamento PETG, numa máquina Bambu Labs P1P, com os seguintes parâmetros:

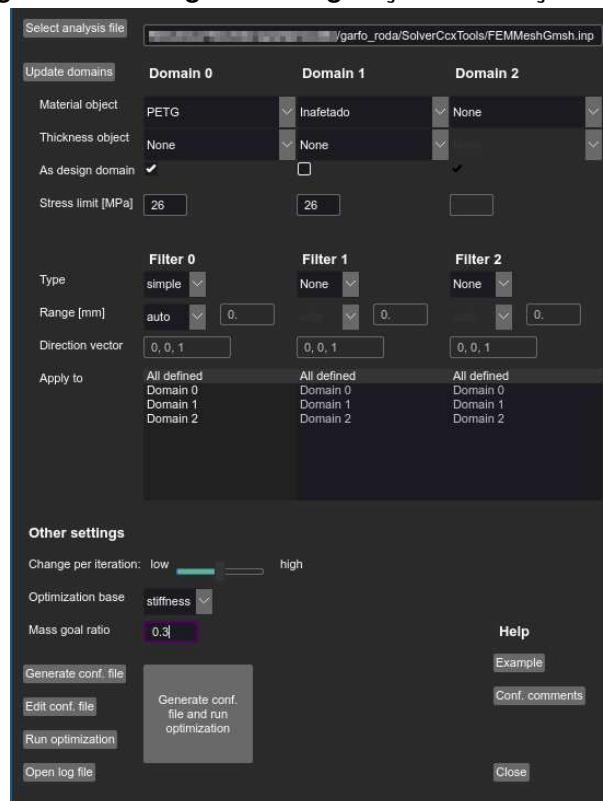
- a) preenchimento de 70% com perfil gyroid;
- b) camada de 0,2 mm;

Figura 23 – Diálogo de execução de macros



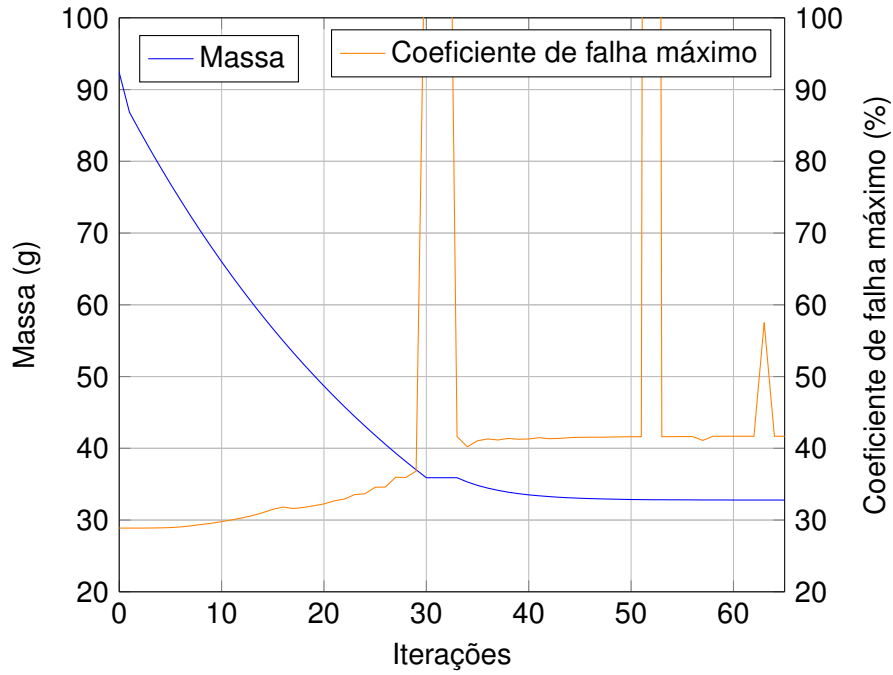
Fonte: Autoria própria (2024).

Figura 24 – Diálogo de configuração e execução da OT



Fonte: Autoria própria (2024).

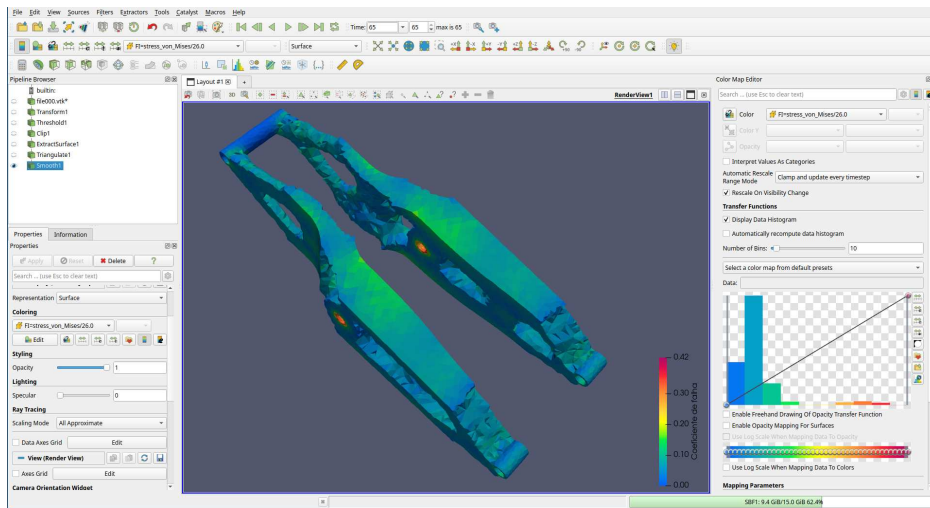
Figura 25 – Progressão da otimização



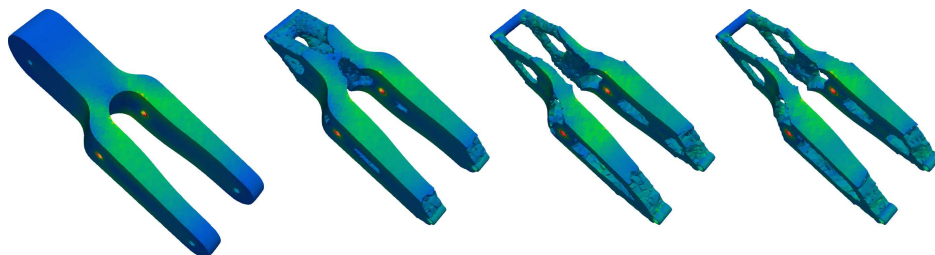
Fonte: Autoria própria (2024).

Figura 26 – Resultados da otimização visualizados no ParaView

(a) Geometria resultante após a aplicação dos filtros



(b) Progressão das iterações 0, 17, 34 e 51

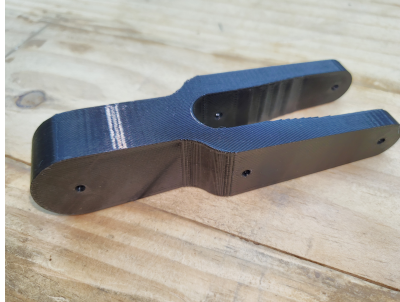


Fonte: Autoria própria (2024).

- c) 3 cascas;
- d) temperatura do bico 220°;
- e) velocidades: camada base 60 mm/s, topo 70 mm/s e o restante a 200 mm/s

Figura 27 – Peças fabricadas

(a) Original



(b) Otimizada



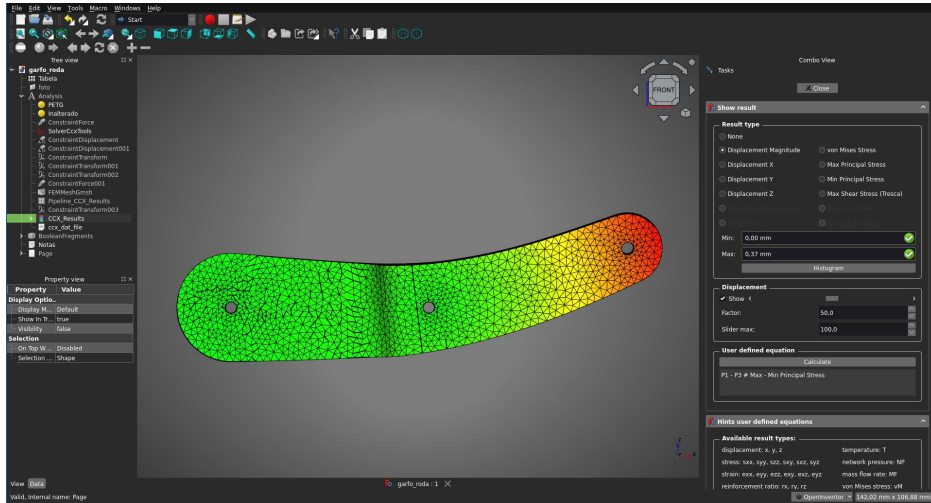
Fonte: Autoria própria (2024).

4.3 Ensaios

O ensaio foi realizado com uma bancada de testes fabricada sob medida (Figura 29). O gabarito consiste em um caibro com dois mancais aparafusados, por onde são passados dois pinos de 3 mm de diâmetro, na forma de pregos, que afixam a peça nos pontos *A* e *B*. Há uma marcação de 68° do eixo central da peça, de maneira que puxando a balança pendurada no pino em *C* neste sentido, a força aplicada corresponde à situação em estudo. A balança possui capacidade de 12 kgf ou aproximadamente 118 N.

A deformação prevista na peça original é de 0,37 mm (Figura 28), e a peça aliviada pode ser considerada validada se suportar 9,07 kgf, o equivalente a F_R . Após da peça foi colocada uma régua graduada, e acima de tudo foi montado um celular com câmera gravando em alta velocidade, de maneira a capturar a quantidade de deformação e a força aplicada. A bancada passou por um teste de calibração e garantia da estabilidade (Figura 30), de onde saiu ileso e partiu para o ensaio das peças.

Figura 28 – Deformação prevista



Fonte: A autoria própria (2024).

Figura 29 – Montagem da bancada de testes

(a) Suporte, peças, pinos, balança

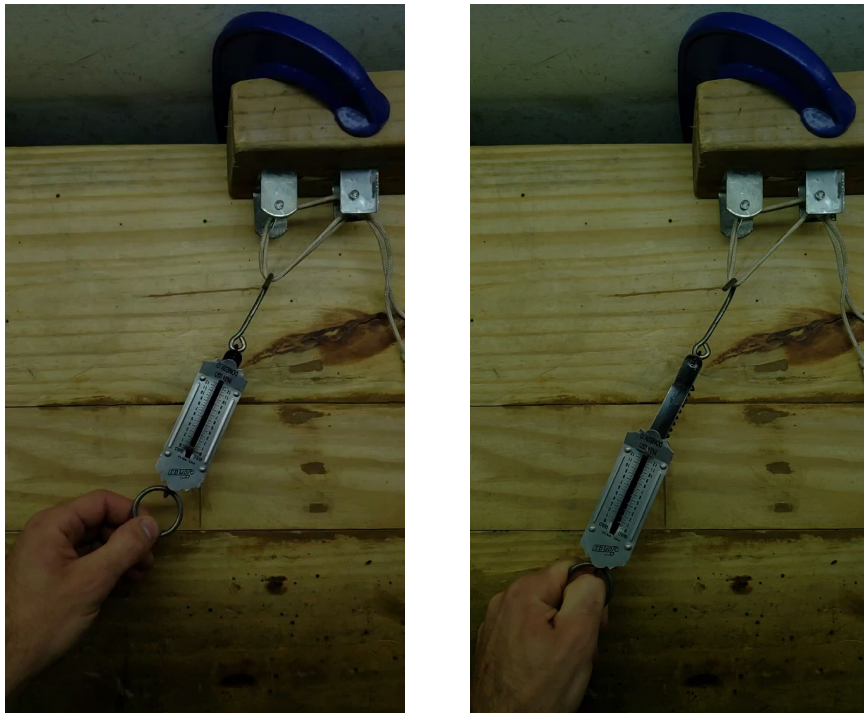


(b) Régua e transferidor



Fonte: A autoria própria (2024).

Figura 30 – Teste da bancada
(a) Sem carga **(b) Carga máxima**



Fonte: Autoria própria (2024).

5 RESULTADOS

O peso das peças fabricadas foi de 46 g e 14 g, uma diferença percentual de 70%, condizente com o determinado como objetivo e obtido durante a otimização.

A fim de avaliar o motivo da disparidade entre os pesos medidos e calculados, num vaso graduado contendo um volume conhecido de água, a peça aliviada foi submergida, determinando assim que ela possui um volume de aproximadamente 25 mL, o que representaria uma densidade de 560 kg/m^3 , muito abaixo da especificação. Considerando então que a peça possui vazios internos, já que foi produzida com preenchimento de 70%, foi encontrada uma nova especificação de densidade para o PETG (MATWEB, 2024), de 1270 kg/m^3 , e verificado que houve erro na definição da densidade durante os cálculos. Recalculando a massa da peça original no FreeCAD, com o novo valor de densidade, foi obtido 64 g, do qual 70% é 45 g, muito próximo do peso da peça real.

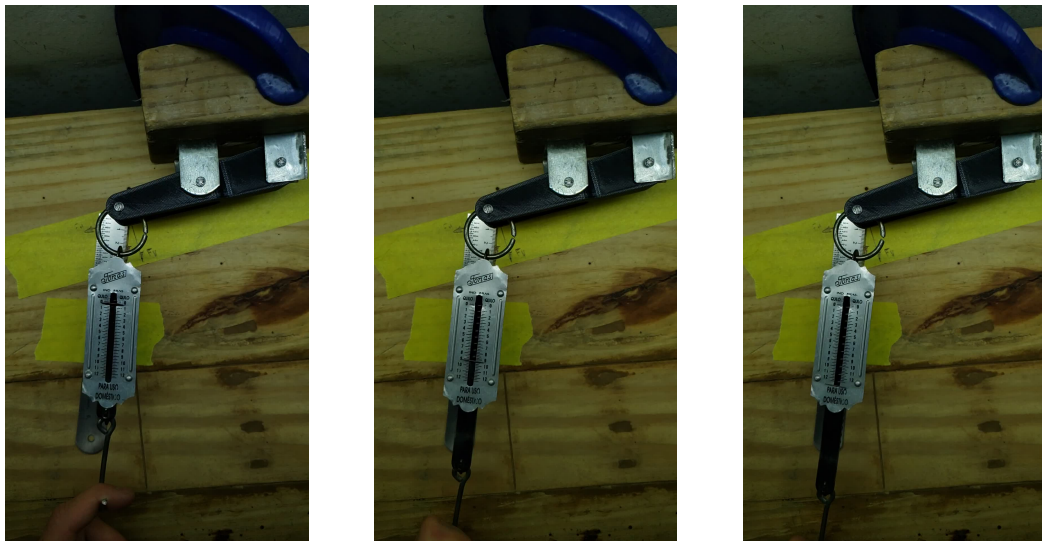
Durante os ensaios de tração, na peça original, a deflexão foi de aproximadamente 1 mm a 9 kgf e 2 mm no fim de curso (Figura 31), e na peça otimizada, a deflexão foi de aproximadamente 2 mm a 9 kgf e 3 mm no fim de curso (Figura 32).

Figura 31 – Ensaio da peça original

(a) Estado inicial

(b) F_R aplicada

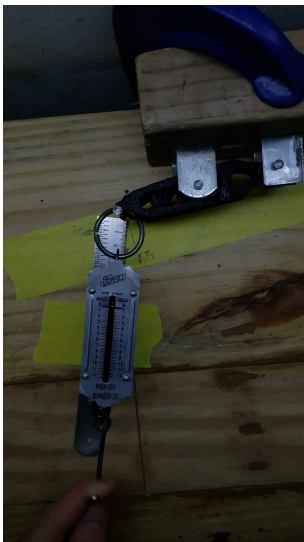
(c) Fim de curso



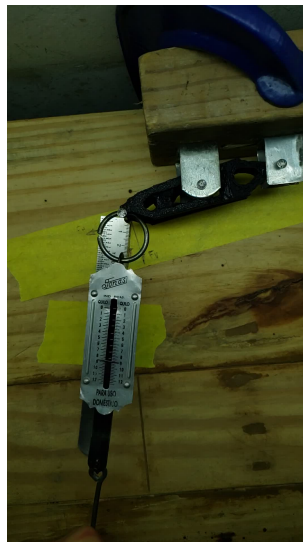
Fonte: Autoria própria (2024).

Figura 32 – Ensaio da peça otimizada

(a) Estado inicial



(b) F_R aplicada



(c) Fim de curso



Fonte: Autoria própria (2024).

6 CONCLUSÕES E PERSPECTIVAS

O estudo termina com sucesso, documentando o processo aplicado para implementar a metodologia de OT utilizando apenas programas de código livre, e validando a capacidade destes de oferecerem todo o necessário para a execução do projeto.

Ficou evidente que a peça otimizada é excessivamente frágil em qualquer carregamento diferente daquele previsto em projeto, e que na utilização da OT é importante definir quaisquer outras condições de forças à qual a peça ou sistema pode vir a ser submetido.

Ficam delineadas abaixo sugestões para trabalhos futuros:

- a) implementação da OT considerando outras funções de objetivo e sensibilidade, ajustando outros parâmetros de contribuição ou penalização sobre os elementos da malha, a fim de por exemplo buscar uma superfície ótima sob efeitos da convecção, de maneira a acoplar as soluções estrutural e fluidodinâmica, a fim de obter peças cada vez mais capazes;
- b) realização de um ensaio cinemático de prova de queda ou equivalente, em vez de um carregamento estático;
- c) melhor descrever o material anisotrópico obtido pela manufatura aditiva, definindo os módulos de *Poisson* distintos nas direções x , y e z . O CalculiX já possui essa funcionalidade.
- d) fabricação, utilizando as peças obtidas por manufatura aditiva como moldes, das peças fundidas em alumínio, a fim de garantir peças de material homogêneo e isotrópico, em que podem ser realizados ensaios de comportamento mecânico com maior precisão, isolando a influência do algoritmo de otimização das limitações do método de fabricação por impressão 3D polimérica.

REFERÊNCIAS

- AAGE, N.; ANDREASSEN, E.; LAZAROV, B. S. Topology optimization using PETSc: An easy-to-use, fully parallel, open source topology optimization framework. **Structural and Multidisciplinary Optimization**, Springer Science and Business Media LLC, v. 51, n. 3, p. 565–572, ago. 2014.
- AAGE, N. *et al.* Giga-voxel computational morphogenesis for structural design. **Nature**, Springer Science and Business Media LLC, v. 550, n. 7674, p. 84–86, out. 2017.
- ABELING, A. **Dimensionamento de um conjunto trem de pouso para aerodesign**. 2016. Dissertação (Mestrado em Engenharia) — FAHOR, Horizontina, 2016. Disponível em: https://fahor.com.br/images/Documentos/Biblioteca/TFCs/Eng_Mecanica/2016/Ariel_Abeling.pdf. Acesso em: 26/05/2024.
- AHRENS, J.; GEVECI, B.; LAW, C. Visualization handbook. *In*: _____. Burlington, MA, USA: Elsevier Inc., 2005. cap. ParaView: An End-User Tool for Large Data Visualization, p. 717–731. ISBN 9780123875822. Disponível em: <https://www.sciencedirect.com/book/9780123875822/visualization-handbook>.
- BALAY, S. *et al.* **PETSc**. 1991. Disponível em: <https://petsc.org/>. Acesso em: 24/06/2023.
- BENDSØE, M. P.; KIKUCHI, N. Generating optimal topologies in structural design using a homogenization method. **Computer Methods in Applied Mechanics and Engineering**, Elsevier BV, v. 71, n. Denmark, p. 197–224, nov. 1988. ISSN 0045-7825.
- BENDSØE, M. P.; SIGMUND, O. **Topology Optimization: Theory, Methods, and Applications**. 2. ed. Berlin: Springer-Verlag, 2013. ISBN 9783662050866.
- CHAPRA, S. C.; CLOUGH, D. E. **ISE Applied Numerical Methods with Python for Engineers and Scientists**. New York: McGraw-Hill Education, 2021. ISBN 9781265017965.
- CHILDS, H. *et al.* **VisIt: An End-User Tool for Visualizing and Analyzing Very Large Data**. [S.l.], 2012.
- COUTINHO, K. D. **Método de otimização topológica em estruturas tridimensionais**. 2006. Dissertação (Mestrado em Tecnologia de Materiais; Projetos Mecânicos; Termociências) — Universidade Federal do Rio Grande Do Norte, 2006. Disponível em: <https://repositorio.ufrn.br/handle/123456789/15665>. Acesso em: 24/06/2023.
- DHONDT, G.; WITTIG, K. **CalculiX: A Free Software Three-Dimensional Structural Finite Element Program**. 1998. Disponível em: <http://www.calculix.de/>. Acesso em: 24/06/2023.
- DUNN, C. **Lost Technologies of Ancient Egypt: Advanced Engineering in the Temples of the Pharaohs**. Inner Traditions/Bear, 2010. ISBN 9781591439684. Disponível em: <https://books.google.com.br/books?id=g1ooDwAAQBAJ>.
- ESPILOTRO, S. R. F. (Ed.). **O Melhor de Top Gun: Os Mais Famosos Aviões de Guerra**. São Paulo: Globo, 1991. v. 1.
- GEUZAINÉ, C.; REMACLE, J.-F. **Gmsh: A three-dimensional finite element mesh generator with built-in pre- and post-processing facilities**. 1997. Disponível em: <http://gmsh.info/>. Acesso em: 24/06/2023.
- HIBBELER, R. C. **Resistência dos Materiais**. 7. ed. São Paulo: Pearson Education do Brasil, 2022. 637 p. ISBN 9788576053736.

- HUANG, X.; XIE, M. **Evolutionary Topology Optimization of Continuum Structures: Methods and Applications**: Methods and applications. Hoboken, Nova Jersey, EUA: Wiley & Sons, Limited, John, 2010. 240 p. ISBN 9780470689486.
- KÅMARK, N. **Development Process of Topology Optimized Casted Components**. 2018. Dissertação (Applied Mechanics) — Chalmers University of Technology, 2018. Disponível em: <https://hdl.handle.net/20.500.12380/254903>. Acesso em: 24/06/2023.
- KAMBAMPATI, S. *et al.* OpenLSTO: Open-source software for level set topology optimization. *In: 2018 Multidisciplinary Analysis and Optimization Conference*. American Institute of Aeronautics and Astronautics, 2018. Disponível em: <https://github.com/M2DOLab/OpenLSTO>. Acesso em: 24/06/2023.
- KINTEL, M.; WOLF, C. **OpenSCAD: The Programmers Solid 3D CAD Modeller**. 2011. Disponível em: <https://openscad.org/>. Acesso em: 24/06/2023.
- KIRKPATRICK, D. Now every company is a software company. **Forbes**, nov. 2011. Disponível em: <https://www.forbes.com/sites/teconomy/2011/11/30/now-every-company-is-a-software-company>. Acesso em: 23/04/2023.
- KON, F. *et al.* Software livre e propriedade intelectual: Aspectos jurídicos, licenças e modelos de negócio. **XXX Jornada de Atualização em Informática (JAI'2011), Sociedade Brasileira de Computação**, p. 59–107, jul. 2011. Disponível em: <https://ccsl.ime.usp.br/files/publications/files/2011/slpi.pdf>. Acesso em: 24/06/2023.
- LAZAROV, B. S. *et al.* Experimental validation of additively manufactured optimized shapes for passive cooling. **Applied Energy**, Elsevier BV, Denmark, v. 226, p. 330–339, set. 2018.
- LÖFFELMANN, F. **calculix/beso**. 2017. Disponível em: <https://github.com/calculix/beso>. Acesso em: 24/06/2023.
- LÖFFELMANN, F. Failure index based topology optimization for multiple properties. *In: FUIS, V. (Ed.). Proceedings of the 23rd International Conference on Engineering Mechanics*. Brno, Czech Republic: University of Technology, Faculty of Mechanical Engineering, Institute of Solid Mechanics, Mechatronics and Biomechanics, 2017. v. 23, p. 590–593. ISSN 1805-8248. Disponível em: <https://www.engmech.cz/improc/2017/0590.pdf>. Acesso em: 24/06/2023.
- MARTINS, J. R. R. A.; NING, A. **Engineering Design Optimization**. 1. ed. Cambridge, England: Cambridge University Press, 2021. ISBN 9781108833417. Disponível em: <http://flowlab.groups.et.byu.net/mdobook.pdf>. Acesso em: 24/06/2023.
- MATWEB. **Overview of materials for PETG Copolyester**. 2024. Disponível em: <https://www.matweb.com/search/datasheet.aspx?matguid=4de1c85bb946406a86c52b688e3810d0>. Acesso em: 26/06/2024.
- ORTEGA, G. Linux for the international space station program. **Linux Journal**, Belltown Media Houston, TX, v. 1999, n. 59es, p. 8–es, 1999. Disponível em: <https://dl.acm.org/doi/fullHtml/10.5555/327697.327705>. Acesso em: 24/06/2023.
- RAYMOND, E. S. **The Cathedral & the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary**. O'Reilly Media, 2001. ISBN 9780596553968. Disponível em: <https://lists.gnu.org/archive/html/groff/2021-11/pdfRt8tRop5yy.pdf>. Acesso em: 24/06/2023.
- RIEGEL, J.; MAYER, W.; HAVRE, Y. van. **FreeCAD: Your own 3D parametric modeler**. 2002. Disponível em: <https://www.freecad.org/>. Acesso em: 24/06/2023.
- ROSNITSCHKEK, T. *et al.* Manufacturing constraints in topology optimization for the direct manufacturing of extrusion-based additively manufactured parts. **Designs**, MDPI AG, v. 7, n. 1, p. 8, jan. 2023.

- ROSSUM, G. V.; DRAKE, F. L. **Python 3 Reference Manual**. Scotts Valley, CA: CreateSpace, 2009. 244 p. ISBN 9781441412690.
- ROZVANY, G. I. N.; ZHOU, M.; BIRKER, T. Generalized shape optimization without homogenization. **Structural Optimization**, Springer Science and Business Media LLC, v. 4, n. 3-4, p. 250–252, set. 1992.
- SCHÖBERL, J. NETGEN an advancing front 2d/3d-mesh generator based on abstract rules. **Computing and Visualization in Science**, Springer Science and Business Media LLC, v. 1, n. 1, p. 41–52, jul. 1997.
- SCHÖBERL, J. **NGSolve**. 2000. Disponível em: <https://ngsolve.org/>. Acesso em: 24/06/2023.
- SILVA, M. **Em consideração aos filhos dos nossos tataranetos**. Gazeta do Povo, 2019. Disponível em: <https://www.gazetadopovo.com.br/vozes/marleth-silva/em-consideracao-aos-filhos-dos-nossos-tataranetos/>. Acesso em: 29/04/2023.
- SILVA, M. V. B. d. **Caracterização mecânica de materiais utilizados em prototipagem rápida por deposição de material fundido para aplicação aeroespacial**. 2018. Dissertação (Bacharelado em Engenharia Aeroespacial) — Universidade de Brasília, 2018. Disponível em: <https://bdm.unb.br/handle/10483/22616>.
- SILVEIRA, S. A. da. **Software Livre: A Luta Pela Liberdade do Conhecimento**. Editora Fundação Perseu Abramo, 2004. (Brasil urgente). ISBN 9788576430032. Disponível em: https://fpabramo.org.br/publicacoes/wp-content/uploads/sites/5/2017/05/Software_livre.pdf.
- SIVAKUMAR, P.; KUMAR, B. V.; DEVI, R. S. S. **Software Engineering for Automotive Systems: Principles and Applications**: Principles and applications. 1. ed. Boca Raton: CRC Press, 2022. ISBN 9781003269908.
- VATANABE, S. L. *et al.* Topology optimization with manufacturing constraints: A unified projection-based approach. **Advances in Engineering Software**, v. 100, p. 97–112, 2016. ISSN 0965-9978. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0965997816301703>.
- VOLPATO, N. **Manufatura aditiva: tecnologias e aplicações da impressão 3D**. 1. ed. São Paulo: Blucher, 2021. ISBN 9788521211518.
- WANG, M. Y.; WANG, X.; GUO, D. A level set method for structural topology optimization. **Computer Methods in Applied Mechanics and Engineering**, Elsevier BV, v. 192, n. 1-2, p. 227–246, jan. 2003.
- ZHU, J.-H.; ZHANG, W.-H.; XIA, L. Topology optimization in aircraft and aerospace structures design. **Archives of Computational Methods in Engineering**, Springer Science and Business Media LLC, v. 23, n. 4, p. 595–622, abr. 2015.

GLOSSÁRIO

.**GEO** formato de arquivo utilizado pelo Gmsh. 16

macro sequência de comandos ou conjunto de instruções armazenadas. 4, 14, 16, 31, 36

script rotina de instruções que serão interpretadas por um programa. 14, 16

.**STL** formato de arquivo utilizado para descrever geometrias tridimensionais, do inglês *Standard Triangle Language* ou *Standard Tessellation Language*. 15