

**UNIVERSIDADE LUTERANA DO BRASIL**  
**DIRETORIA ACADÊMICA**  
**ENGENHARIA MECÂNICA**



**OTIMIZAÇÃO TOPOLÓGICA DE BIELA ATRAVÉS DE UM SOFTWARE  
DESENVOLVIDO EM PYTHON**

**JORGE ALBERTO MARTINS DE ALMEIDA JUNIOR**

**Canoas, outubro de 2020**

JORGE ALBERTO MARTINS DE ALMEIDA JUNIOR

OTIMIZAÇÃO TOPOLÓGICA DE BIELA ATRAVÉS DE UM SOFTWARE  
DESENVOLVIDO EM PYTHON

Trabalho de conclusão apresentado ao  
Curso de Engenharia Mecânica da  
Universidade Luterana do Brasil para  
obtenção do grau de Engenheiro  
Mecânico.

Orientador: Profº Ms. Eng. Mec. Antônio Flavio Aires Rodrigues

Canoas, outubro de 2020

## RESUMO

O trabalho descreve o equacionamento necessário para realização de uma análise estática através do Método dos Elementos Finitos de um domínio computacional discretizado com elementos tetraédricos, apresentando como os valores são organizados dentro das matrizes utilizadas para o desenvolvimento de um Sistema de Equações que descreve as condições sob as quais o domínio está submetido. Com base no Método dos Elementos Finitos é realizada a implementação de um método de Otimização Topológica Evolucionário chamado BESO (Bidirectional Evolutionary Structure Optimization), utilizando as tensões equivalentes de Von Mises como um dos principais parâmetros realizar a distribuição de material dentro de um domínio computacional. Ambos os métodos citados anteriormente foram implementados em um software desenvolvido em linguagem Python. Por fim foram realizadas duas otimizações de um domínio correspondente à uma biela, com a finalidade de comparar o desempenho mecânico das geometrias obtidas pelo software desenvolvido com uma biela convencional. A primeira otimização foi realizada para se obter uma geometria com o mesmo volume de uma biela convencional, e a geometria obtida demonstrou melhorias na distribuição das tensões, da energia interna e também uma redução nos deslocamentos dos elementos. A segunda otimização, teve como resultados uma biela com menor volume, cerca de 16,7%, e o desempenho mecânico foi muito próximo ao desempenho da biela convencional.

O programa desenvolvido pelo autor está disponível em <https://github.com/jorgjr/toPy>.

**PALAVRAS-CHAVE:** Otimização Topológica, Python, Elementos Finitos

## **ABSTRACT**

The work describes the equation necessary to carry out a static analysis using the Finite Element Method of a computational domain discretized with tetrahedral elements, presenting how the values are organized within the matrices used for the development of a System of Equations that describes the conditions under the domain is submitted to. Based on the Finite Element Method, an Evolutionary Topological Optimization method called BESO (Bidirectional Evolutionary Structure Optimization) is implemented, using the Von Mises equivalent tensions as one of the main parameters to perform the material distribution within a computational domain. Both methods mentioned above were implemented in software developed in Python language. Finally, two optimizations of a domain corresponding to a connecting rod were performed, in order to compare the mechanical performance of the geometries obtained by the software developed with a conventional connecting rod. The first optimization was performed to obtain a geometry with the same volume as a conventional connecting rod, and the obtained geometry demonstrated improvements in the distribution of stresses, internal energy and also a reduction in the displacements of the elements. The second optimization resulted in a connecting rod with a smaller volume, about 16.7%, and the mechanical performance was very close to the performance of the conventional connecting rod.

The program developed by the author is disponible in <https://github.com/jorgjr/toPy>.

**KEYWORDS:** Topology Optimization, Python, Finite Elements.

## SUMÁRIO

1. INTRODUÇÃO	1
2. OBJETIVOS	2
2.1. OBJETIVO GERAL	2
2.2. ORGANIZAÇÃO DO TRABALHO	2
3. MÉTODO DOS ELEMENTOS FINITOS	4
3.1. INTRODUÇÃO AO MÉTODO DOS ELEMENTOS FINITOS	4
3.2. ELEMENTO TETRAÉDRICO LINEAR E A MATRIZ DE RIGIDEZ	6
3.3. CONSTRUÇÃO DAS MATRIZES GLOBAIS	9
4. OTIMIZAÇÃO TOPOLÓGICA	12
4.1. MÉTODOS EVOLUCIONÁRIOS	12
4.2. FUNÇÃO OBJETIVO DO MÉTODO BESO	12
4.3. ANÁLISE E FILTRO DE SENSIBILIDADE	13
4.4. ATUALIZAÇÃO DA GEOMETRIA E CONVERGÊNCIA	15
5. PROGRAMA	16
5.1. A LINGUAGEM PYTHON	16
5.2. NUMPY, SCIPY E GMSH	16
5.3. ORGANIZAÇÃO DO SOFTWARE	17
5.4. BLACK BOX	17
5.5. INTERFACE GRÁFICA	19
5.6. TESTES	20
6. BIELA	22
6.1. CONCEITOS BÁSICOS DE UMA BIELA	22
6.2. ANÁLISE ESTÁTICA LINEAR DE UMA BIELA	23
6.3. OTIMIZAÇÃO TOPOLÓGICA DE UMA BIELA	24
7. RESULTADOS	27
7.1. RESULTADOS DA OTIMIZAÇÃO TOPOLÓGICA	27
7.1.1 GEOMETRIA OBTIDA COM 31% DA FRAÇÃO DE VOLUME	27
7.1.2 GEOMETRIA OBTIDA COM 26% DA FRAÇÃO DE VOLUME	28

7.2.	<i>RESULTADOS DA ANÁLISE ESTÁTICA LINEAR</i>	29
7.2.1	<i>RESULTADOS DA BIELA CONVENCIONAL</i>	29
7.2.2	<i>RESULTADOS DA BIELA OTIMIZADA COM 31% DE FRAÇÃO DO VOLUME</i>	31
7.2.3	<i>RESULTADOS DA BIELA OTIMIZADA COM 26% DE FRAÇÃO DO VOLUME</i>	33
8.	CONCLUSÕES	36
9.	REFERÊNCIAS BIBLIOGRÁFICAS	37

## 1. INTRODUÇÃO

Os primeiros estudos relacionados ao melhor aproveitamento da matéria-prima em estruturas foram realizados por Maxwell, em 1872, conforme Silva E. C. N. (2001), onde ele procurou calcular as tensões principais de estruturas dado um carregamento e sugeriu conceitualmente que a estrutura ótima teria uma distribuição de material alinhado às direções das tensões principais, já que nelas não ocorrem tensões de cisalhamento, posteriormente Michell em 1904, obteve alguns avanços, porém para a época os resultados foram considerados não tiveram nenhuma aplicação prática, já que a produção das estruturas obtidas era praticamente impossível, devido às limitações da tecnologia da época. A idéia de Maxwell se provou mais tarde estar correta, para estruturas com um único carregamento para se obter uma estrutura com máxima rigidez e menor volume de material.

Depois de Michell somente na década de 80, após o surgimento dos computadores pessoais e da evolução do Método dos Elementos Finitos este problema voltou a ser estudado, principalmente após a publicação de Bendsøe e Kikuchi (1988), onde foi apresentado a otimização topológica através do método da homogeneização.

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

A utilização do Método dos Elementos Finitos é cada vez mais presente no desenvolvimento de produtos, é possível encontrar na literatura inúmeros trabalhos investigando técnicas computacionais para aumentar a velocidade de processamento de dados matriciais, bem como de se obter resultados mais precisos. Diversas técnicas computacionais estão presentes em linguagens de alto nível possibilitando a criação de rotinas complexas de cálculo sem grandes dificuldades.

Este trabalho apresenta todas as etapas necessárias para o desenvolvimento de um software capaz de realizar uma análise estática através do Método dos Elementos Finitos, e com base nos resultados obtidos, realizar uma Otimização Topológica. Por fim será apresentado uma comparação do desempenho mecânico entre um produto convencional com um produto conceitual obtido após o processo de otimização.

## **2. OBJETIVOS**

### *1.1. OBJETIVO GERAL*

Este trabalho têm como objetivo desenvolver um programa para fins educacionais na linguagem Python, capaz de realizar a Otimização Topológica através de uma discretização através de elementos tetraédricos utilizando como base para os cálculos o Método dos Elementos Finitos, e comparar os resultados obtidos através de uma análise estática de uma biela otimizada com uma biela comercial utilizando o Ansys Student 2020 R1.

### *1.2. ORGANIZAÇÃO DO TRABALHO*

Este trabalho foi dividido em nove capítulos, abordando cada assunto necessário para atingir os objetivos separadamente para uma melhor compreensão dos problemas estudados.

No primeiro capítulo é realizada a introdução sobre a evolução da otimização topológica ao longo dos últimos anos e de como o Método dos Elementos Finitos vêm sendo utilizado pela indústria.

O segundo capítulo informa quais são os objetivos principais a serem alcançados e também mostra como foi feita a organização do trabalho.

O terceiro capítulo introduz o Método dos Elementos Finitos, tendo como foco a discretização do domínio computacional a ser estudado em elementos tetraédricos.

O quarto capítulo trata exclusivamente da formulação da Otimização Topológica, utilizando o método BESO com interpolação de material com penalização baseado nas tensões equivalentes de Von Mises.

O quinto capítulo apresenta o programa desenvolvido pelo autor para realizar a análise através do Método de Elementos Finitos e a Otimização Topológica.

No sexto capítulo é desenvolvida uma análise de como as forças atuam sobre a biela, e de como será realizada a análise estática.



O sétimo capítulo demonstra os resultados de otimização obtidos no programa desenvolvido, e também análises estáticas da biela otimizada, e da biela comercial.

No oitavo capítulo são apresentadas as conclusões obtidas através dos resultados da Otimização Topológica e também sobre as análises realizadas para comparação entre as geometrias.

No nono e último capítulo é listado todas as referências bibliográficas utilizadas ao longo para o desenvolvimento deste trabalho.

### 3. MÉTODO DOS ELEMENTOS FINITOS

#### 3.1. INTRODUÇÃO AO MÉTODO DOS ELEMENTOS FINITOS

Segundo Kattan P. (2008) o Método dos Elementos Finitos é um método de cálculo numérico utilizado para analisar e resolver diversos problemas de engenharia. Este método consiste de basicamente seis etapas bem definidas que todo programa desenvolvido para realizá-lo precisa seguir.

1. Discretização: É a primeira etapa de toda análise realizada através do Método dos Elementos Finitos, nesta etapa ocorre a divisão de um domínio computacional em diversos elementos de geometria simples. O domínio computacional pode representar, por exemplo, um corpo a ser analisado em casos de análises mecânicas, uma região de mistura de fluídos diferentes utilizada para uma análise de escoamento de fluídos. Este domínio pode ser desenvolvido basicamente de três formas: em um programa do tipo CAD (Computer Aided Design); pode ser descrito através de equações paramétricas; utilizando funções de criação de formas encontradas em diversas linguagens de programações. A Figura 3 apresenta um exemplo de um domínio discretizado em elemento tetraédricos.
2. Matriz de Rigidez: Cada elemento presente no domínio deve ser representado de forma numérica para ser possível a resolução das equações em análise, para isto cada elemento possui sua Matriz de Rigidez, onde nela está presente algumas informações do elemento, como as coordenadas de suas extremidades e seu volume.
3. Matriz de Rigidez Global: Após a representação individual dos elementos, é possível representar todo o domínio em uma única matriz, a Matriz de Rigidez Global.
4. Condições de Contorno: Para finalizar a construção das equações após a obtenção da Matriz de Rigidez Global é preciso descrever o problema a ser analisado pelo programa, e isto é feito através da inclusão das Condições de Contorno. Cada ponto existente no domínio discretizado, por exemplo, no caso de uma análise de primeira ordem serão as extremidades dos elementos, ou em análises de segunda ordem além das extremidades há pontos intermediários entre as extremidades dos elementos, deverá ter um valor atribuído a ele de acordo com a sua Condição de Contorno.

5. Resolução das Equações: Com a Matriz de Rigidez Global, e as matrizes das Condições de Contorno é possível construir e resolver as equações em um único Sistema de Equações. A resolução desses sistemas através de métodos numéricos utilizando computadores dá a possibilidade do desenvolvimento de sistemas com milhares de equações.
6. Pós-processamento: Após a resolução das equações o pós-processamento deve ser executado, pois muitos resultados desejados precisam de mais desenvolvimento, como por exemplo se for realizada a apresentação do deslocamento de um corpo submetido a uma carga, deve ser verificada qual foi o deslocamento em cada ponto, e este valor deverá modificar a posição do ponto no domínio para ser possível a visualização do deslocamento, a representação da intensidades das tensões no corpo através um gradiente de cores, ou até mesmo a utilização de uma representação com flechas representando a direção, o sentido e a intensidade do fluxo de um fluido em uma determinada região.

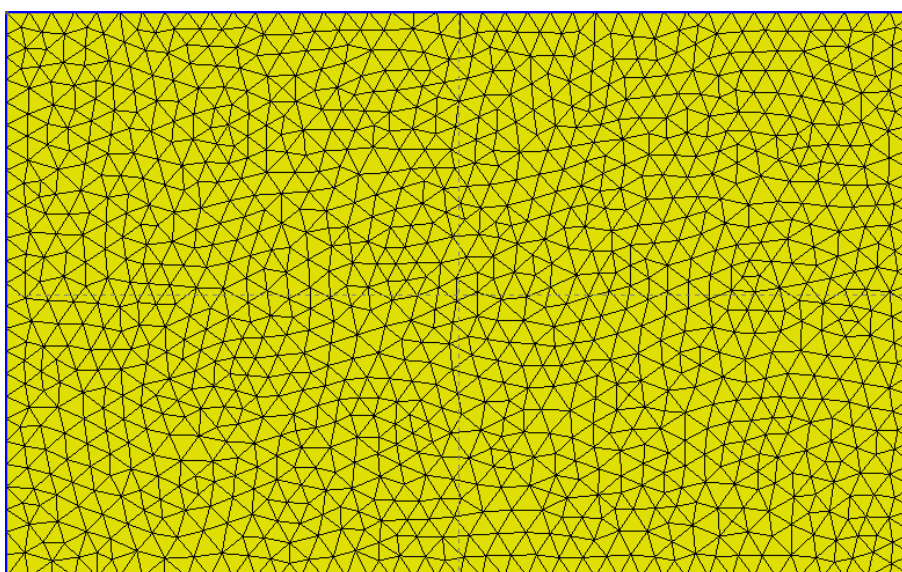


Figura 3.1.1. Exemplo de um Domínio Computacional discretizado.

O programa foi desenvolvido para resolver um modelo elástico-linear, através de um domínio discretizado em elementos tetraédricos de primeira ordem, devido ao baixo custo computacional que este tipo apresenta durante a construção da Matriz de Rigidez de cada elemento.

### 3.2. ELEMENTO TETRAÉDRICO LINEAR E A MATRIZ DE RIGIDEZ

O elemento tetraédrico linear é um elemento sólido, tridimensional, com um ponto em cada extremidade, ou seja, quatro pontos e estes pontos utilizados são chamados de nós. Este é o elemento mais simples para a discretização de um sólido e o que possui o menor custo computacional para o desenvolvimento da Matriz de Rigidez.

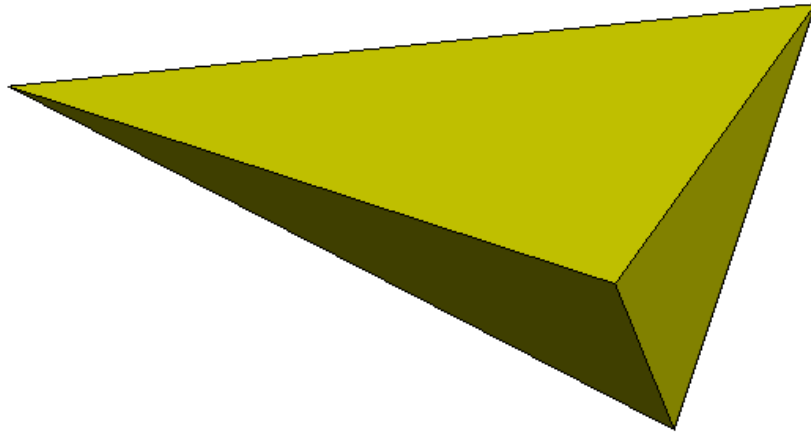


Figura 3.2.1. O elemento tetraédrico.

A construção da Matriz de Rigidez de um elemento tetraédrico é obtida por:

$$K_e = V_e B_e^T D B_e \quad (3.2.1)$$

Onde  $V_e$  é o volume de cada elemento,  $B_e$  é a Matriz de Deformação, que relaciona os deslocamentos nodais com as deformações e  $D$  é o Tensor de Elasticidade. Esta equação apresentada só pode ser considerada quando for assumido que as propriedades do material se mantêm constante em todo o domínio. O volume de um tetraédro pode ser calculado conforme:

$$V_e = \frac{1}{6} \begin{vmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \end{vmatrix} \quad (3.2.2)$$

O Tensor de Elasticidade é encontrado utilizando a seguinte equação:

$$D = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2}-\nu & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2}-\nu & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2}-\nu \end{bmatrix} \quad (3.2.2)$$

$\nu$  é o Coeficiente de Poisson do material, já a matriz B é uma matriz que depende das Funções de Formas, que são polinômios de Lagranges que transformam o sistema de coordenada global para um local parametrizando todos os elementos do domínio discretizado. A ordem desses polinômios varia de acordo com o número de nós de cada elemento, e há uma função de forma para cada nó do elemento, que no caso do elemento tetraédrico são quatro Funções de Formas.

As Funções de Forma de um tetraédro são:

$$\begin{aligned} N_1 &= \frac{1}{6V_e} (\alpha_1 + \beta_1 x + \gamma_1 y + \delta_1 z) \\ N_2 &= \frac{1}{6V_e} (\alpha_2 + \beta_2 x + \gamma_2 y + \delta_2 z) \\ N_3 &= \frac{1}{6V_e} (\alpha_3 + \beta_3 x + \gamma_3 y + \delta_3 z) \\ N_4 &= \frac{1}{6V_e} (\alpha_4 + \beta_4 x + \gamma_4 y + \delta_4 z) \end{aligned} \quad (3.2.4)$$

Os valores de  $\alpha_i$ ,  $\beta_i$ ,  $\gamma_i$ , e  $\delta_i$  por sua vez, dependem das coordenadas globais dos nós:

$$\begin{aligned} \alpha_1 &= \begin{vmatrix} x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \\ x_4 & y_4 & z_4 \end{vmatrix} & \alpha_2 &= - \begin{vmatrix} x_1 & y_1 & z_1 \\ x_3 & y_3 & z_3 \\ x_4 & y_4 & z_4 \end{vmatrix} & \alpha_3 &= \begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_4 & y_4 & z_4 \end{vmatrix} & \alpha_4 &= - \begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{vmatrix} \\ \beta_1 &= - \begin{vmatrix} 1 & y_2 & z_2 \\ 1 & y_3 & z_3 \\ 1 & y_4 & z_4 \end{vmatrix} & \beta_2 &= \begin{vmatrix} 1 & y_1 & z_1 \\ 1 & y_3 & z_3 \\ 1 & y_4 & z_4 \end{vmatrix} & \beta_3 &= - \begin{vmatrix} 1 & y_1 & z_1 \\ 1 & y_2 & z_2 \\ 1 & y_4 & z_4 \end{vmatrix} & \beta_4 &= \begin{vmatrix} 1 & y_1 & z_1 \\ 1 & y_2 & z_2 \\ 1 & y_3 & z_3 \end{vmatrix} \\ \gamma_1 &= \begin{vmatrix} 1 & x_2 & z_2 \\ 1 & x_3 & z_3 \\ 1 & x_4 & z_4 \end{vmatrix} & \gamma_2 &= - \begin{vmatrix} 1 & x_1 & z_1 \\ 1 & x_3 & z_3 \\ 1 & x_4 & z_4 \end{vmatrix} & \gamma_3 &= \begin{vmatrix} 1 & x_1 & z_1 \\ 1 & x_2 & z_2 \\ 1 & x_4 & z_4 \end{vmatrix} & \gamma_4 &= - \begin{vmatrix} 1 & x_1 & z_1 \\ 1 & x_2 & z_2 \\ 1 & x_3 & z_3 \end{vmatrix} \\ \delta_1 &= - \begin{vmatrix} 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \\ 1 & x_4 & y_4 \end{vmatrix} & \delta_2 &= \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_3 & y_3 \\ 1 & x_4 & y_4 \end{vmatrix} & \delta_3 &= - \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_4 & y_4 \end{vmatrix} & \delta_4 &= \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix} \end{aligned} \quad (3.2.5)$$

Após aplicar os valores dos coeficientes nas Funções de Forma a Matriz de Deformação pode ser desenvolvida:

$$B_e = \begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & 0 & \frac{\partial N_2}{\partial x} & 0 & 0 & \frac{\partial N_3}{\partial x} & 0 & 0 & \frac{\partial N_4}{\partial x} & 0 & 0 \\ 0 & \frac{\partial N_1}{\partial y} & 0 & 0 & \frac{\partial N_2}{\partial y} & 0 & 0 & \frac{\partial N_3}{\partial y} & 0 & 0 & \frac{\partial N_4}{\partial y} & 0 \\ 0 & 0 & \frac{\partial N_1}{\partial z} & 0 & 0 & \frac{\partial N_2}{\partial z} & 0 & 0 & \frac{\partial N_3}{\partial z} & 0 & 0 & \frac{\partial N_4}{\partial z} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & 0 & \frac{\partial N_3}{\partial y} & \frac{\partial N_3}{\partial x} & 0 & \frac{\partial N_4}{\partial y} & \frac{\partial N_4}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial z} & \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial z} & \frac{\partial N_2}{\partial y} & 0 & \frac{\partial N_3}{\partial z} & \frac{\partial N_3}{\partial y} & 0 & \frac{\partial N_4}{\partial z} & \frac{\partial N_4}{\partial y} \\ \frac{\partial N_1}{\partial z} & 0 & \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial z} & 0 & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial z} & 0 & \frac{\partial N_3}{\partial x} & \frac{\partial N_4}{\partial z} & 0 & \frac{\partial N_4}{\partial x} \end{bmatrix} \quad (3.2.6)$$

O resultado obtido através da Equação (3.2.1) é:

$$K_e = \begin{bmatrix} K_{aa} & K_{ab} & K_{ac} & K_{ad} & K_{ae} & K_{af} & K_{ag} & K_{ah} & K_{ai} & K_{aj} & K_{ak} & K_{al} \\ K_{ba} & K_{bb} & K_{bc} & K_{bd} & K_{be} & K_{bf} & K_{bg} & K_{bh} & K_{bi} & K_{bj} & K_{bk} & K_{bl} \\ K_{ca} & K_{cb} & K_{cc} & K_{cd} & K_{ce} & K_{cf} & K_{cg} & K_{ch} & K_{ci} & K_{cj} & K_{ck} & K_{cl} \\ K_{da} & K_{db} & K_{dc} & K_{dd} & K_{de} & K_{df} & K_{dg} & K_{dh} & K_{di} & K_{dj} & K_{dk} & K_{dl} \\ K_{ea} & K_{eb} & K_{ec} & K_{ed} & K_{ee} & K_{ef} & K_{eg} & K_{eh} & K_{ei} & K_{ej} & K_{ek} & K_{el} \\ K_{fa} & K_{fb} & K_{fc} & K_{fd} & K_{fe} & K_{ff} & K_{fg} & K_{fh} & K_{fi} & K_{fj} & K_{fk} & K_{fl} \\ K_{ga} & K_{gb} & K_{gc} & K_{gd} & K_{ge} & K_{gf} & K_{gg} & K_{gh} & K_{gi} & K_{gj} & K_{gk} & K_{gl} \\ K_{ha} & K_{hb} & K_{hc} & K_{hd} & K_{he} & K_{hf} & K_{hg} & K_{hh} & K_{hi} & K_{hj} & K_{hk} & K_{hl} \\ K_{ia} & K_{ib} & K_{ic} & K_{id} & K_{ie} & K_{if} & K_{ig} & K_{ih} & K_{ii} & K_{ij} & K_{ik} & K_{il} \\ K_{ja} & K_{jb} & K_{jc} & K_{jd} & K_{je} & K_{jf} & K_{jg} & K_{jh} & K_{ji} & K_{jj} & K_{jk} & K_{jl} \\ K_{ka} & K_{kb} & K_{kc} & K_{kd} & K_{ke} & K_{kf} & K_{kg} & K_{kh} & K_{ki} & K_{kj} & K_{kk} & K_{kl} \\ K_{la} & K_{lb} & K_{lc} & K_{ld} & K_{le} & K_{lf} & K_{lg} & K_{lh} & K_{li} & K_{lj} & K_{lk} & K_{ll} \end{bmatrix} \quad (3.2.7)$$

Sendo  $a, b, c, d, e, f, g, h, i, j, k$  e  $l$  valores que representam os índices dos nós que fazem parte do elemento em análise da Matriz de Rigidez, e posteriormente da Matriz de Rigidez Global, conforme:

$$\begin{aligned} a &= 3 * n1 - 2; b = 3 * n1 - 1; c = 3 * n1; \\ d &= 3 * n2 - 2; e = 3 * n2 - 1; f = 3 * n2; \\ g &= 3 * n3 - 2; h = 3 * n3 - 1; i = 3 * n3; \\ j &= 3 * n4 - 2; k = 3 * n4 - 1; l = 3 * n4; \end{aligned} \quad (3.2.8)$$

Onde os valores  $n1, n2, n3$ , e  $n4$  correspondem aos índices dos nós do elemento  $e$  que está sendo analisado, a utilização desses índices é muito importante para a

montagem da Matriz de Rigidez Global. A Figura 3.2.2 mostra um exemplo de como os índices devem ser organizados no elemento, o primeiro, o segundo e o terceiro nó do tetraédro devem sempre seguir um sentido crescente e anti-horário em uma face quando visto do quarto nó.

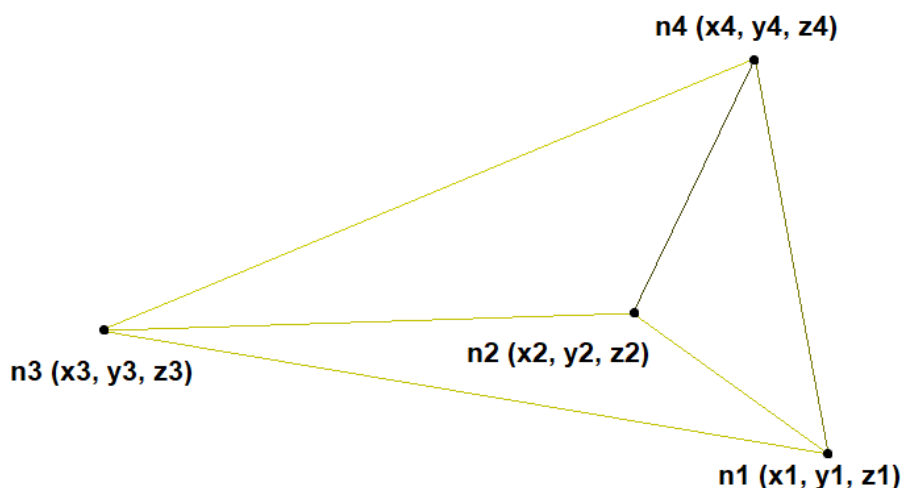


Figura 3.2.2. Exemplo dos índices de um elemento tetraédrico.

Formulações da Matriz de Rigidez de outros tipos de elementos podem ser encontradas em Kattan P. (2008).

### 3.3. CONSTRUÇÃO DAS MATRIZES GLOBAIS

A equação a do modelo elástico-linear é dada pela Lei de Hooke, que é resolvida através de um Sistema de Equações Lineares:

$$\{F\} = [K]\{U\} \quad (3.3.1)$$

Associando a Lei de Hooke com os termos genéricos de um Sistema de Equações temos que  $K$  é a Matriz de Coeficientes, neste caso a Matriz de Rigidez Global,  $F$  é a Matriz de Termos Independentes e  $U$  é a Matriz das Variáveis, sendo que  $F$  contém os valores referente às forças aplicadas sobre cada nó do domínio e  $U$  contém os valores dos deslocamento dos nós deste domínio.

$F$  e  $U$  são matrizes do tipo coluna com  $\lambda$  valores, sendo  $\lambda = n * gl$ , onde  $n$  é o número de nós do domínio e  $gl$  o número de Graus de Liberdade, que representa

quantas direções cada nó pode se deslocar, no caso de um sistema tridimensional há um total de três Graus de Liberdade no sistema de coordenadas cartesiano, sendo elas a direção em  $x$ , em  $y$ , e  $z$ .

O Sistema de Equações se torna melhor de observar ao demonstrar como as variáveis ficam organizadas dentro das matrizes:

$$\begin{Bmatrix} F_{1x} \\ F_{1y} \\ F_{1z} \\ F_{2x} \\ F_{2y} \\ F_{2z} \\ F_{3x} \\ \vdots \\ F_{nz} \end{Bmatrix} = \begin{bmatrix} K_{1,1} & K_{1,2} & K_{1,3} & K_{1,4} & K_{1,5} & K_{1,6} & K_{1,7} & \cdots & K_{1,\lambda} \\ K_{2,1} & K_{2,2} & K_{2,3} & K_{2,4} & K_{2,5} & K_{2,6} & K_{2,7} & \cdots & K_{2,\lambda} \\ K_{3,1} & K_{3,2} & K_{3,3} & K_{3,4} & K_{3,5} & K_{3,6} & K_{3,7} & \cdots & K_{3,\lambda} \\ K_{4,1} & K_{4,2} & K_{4,3} & K_{4,4} & K_{4,5} & K_{4,6} & K_{4,7} & \cdots & K_{4,\lambda} \\ K_{5,1} & K_{5,2} & K_{5,3} & K_{5,4} & K_{5,5} & K_{5,6} & K_{5,7} & \cdots & K_{5,\lambda} \\ K_{6,1} & K_{6,2} & K_{6,3} & K_{6,4} & K_{6,5} & K_{6,6} & K_{6,7} & \cdots & K_{6,\lambda} \\ K_{7,1} & K_{7,2} & K_{7,3} & K_{7,4} & K_{7,5} & K_{7,6} & K_{7,7} & \cdots & K_{7,\lambda} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ K_{\lambda,1} & K_{\lambda,2} & K_{\lambda,3} & K_{\lambda,4} & K_{\lambda,5} & K_{\lambda,6} & K_{\lambda,7} & \cdots & K_{\lambda,\lambda} \end{bmatrix} \begin{Bmatrix} U_{1x} \\ U_{1y} \\ U_{1z} \\ U_{2x} \\ U_{2y} \\ U_{2z} \\ U_{3x} \\ \vdots \\ U_{nz} \end{Bmatrix} \quad (3.3.2)$$

Observando as matrizes da força e do deslocamento pode ser entendido o motivo de haver  $\lambda$  valores de variáveis, onde cada nó está associado à três Graus de Liberdade, e para o sistema ser possível de realizar, a Matriz de Rigidez Global deve ser uma matriz quadrada  $\lambda \times \lambda$ , caso contrário o sistema não terá solução.

A Matriz de Rigidez Global é composta por todos os valores de todas as Matrizes de Rigidez, e quando houver repetição de índices durante a construção da Matriz de Rigidez Global os valores destes índices serão somados.

Já os valores das matrizes de forças e deslocamento dependem das condições de contorno que forem atribuídas ao sistemas. Inicialmente se supõe que todas as forças no sistema são iguais a zero, e que todos os deslocamentos são incógnitas, em seguida são atribuídos os valores das condições aos índices correspondentes.

Todos os nós que pertencem às regiões que são fixadas, terão um deslocamento igual a zero no respectivo índice referente ao nó e ao Grau de Liberdade restringido, e ao mesmo tempo na matriz de força esses mesmo índices se tornarão incógnitas.

Ao possuir incógnitas na Matriz de Termos Independentes o sistema não possui solução, mas ao mesmo tempo sabe-se que o valor correspondente ao mesmo índice na Matriz das Variáveis é igual a zero, logo pode ser realizada uma redução do Sistema



de Equações, e após esta redução será possível resolver este Sistema de Equações reduzido.

Com a resolução deste Sistema de Equações reduzido será obtido todos os valores na Matriz das Variáveis, que no caso é a matriz dos deslocamento, e com todos estes valores é possível encontrar as incógnitas na matriz das forças, e essas incógnitas que foram atribuídas devido às restrições dos Graus de Liberdade nos nós, serão as Forças de Reação presentes nas fixações.

Com os valores dos deslocamentos de todos os nós, e todos os valores das Forças de Reação, é possível então determinar algumas condições, como por exemplo a posição de cada nó após a aplicação da força, ou as tensões em cada elemento.

## 4. OTIMIZAÇÃO TOPOLÓGICA

### 4.1. MÉTODOS EVOLUCIONÁRIOS

Durante os últimos anos diversas abordagens em relação à Otimização Topológica foram desenvolvidas, principalmente após a implementação do Método dos Elementos Finitos em computadores pessoais entre os anos 80 e 90.

O primeiro método evolucionário proposto foi o ESO (Evolutionary Structural Optimization) por X. Huang, Y. M. Xie (2010). A ideia inicial deste método foi de fazer a remoção dos elementos menos eficiente em relação às tensões de toda a estrutura, sendo a quantidade de elementos removidos por iteração limitada à uma quantidade percentual predeterminada.

Já o método BESO (Bidirectional Evolutionary Structure Optimization) além de remover também pode adicionar elementos, dependendo do número da sensibilidade que este elemento apresentar. Neste trabalho foi utilizada uma versão soft-kill do método BESO utilizando uma interpolação de material com penalização baseada nas tensões de Von Mises como parâmetro para a remoção ou inclusão do elemento. Na versão soft-kill o elemento nunca é totalmente removido do domínio, mas sim atribuído um valor de densidade relativa muito baixo, indicando que ele é um elemento vazio.

### 4.2. FUNÇÃO OBJETIVO DO MÉTODO BESO

O problema de otimização para esta variação do método BESO pode ser formulado como:

$$\begin{aligned}
 &\text{Minimizar} && C = \sum_{e=1}^N \frac{1}{2} (x_e)^p u_e^T k_e u_e \\
 &\text{Sujeito a} && V^* - \sum_{e=1}^N V_e x_e = 0 \\
 &&& x_e = x_{\min} \text{ ou } 1
 \end{aligned} \tag{4.2.1}$$

Onde  $C$  é a função objetivo,  $u$  é o vetor de deslocamento,  $k_e$  é a Matriz de Rigidez do elemento,  $N$  é o número de elementos dentro do domínio computacional discretizado,

$V^*$  é o volume determinado que a peça tenha após o processo de otimização,  $V_e$  é o volume de cada elemento, e  $x_e$  é o valor de densidade relativa do elemento (também chamado de vetor da variável de projeto), sendo  $x_{min}$  igual a 0.001 para representar elementos vazios e assim evitar problemas durante a análise de elementos finitos, ou 1 para elementos sólidos, já o valor de  $p$  é valor da potência de penalidade.

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

$$E_{(x_e)} = E_0 x_e \quad (4.2.2)$$

Onde  $E_0$  é o Módulo de Young. O volume pretendido ao fim de cada iteração é dado por:

$$V_k = V_{k-1}(1 - ER) \quad (4.2.3)$$

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

#### 4.3. ANÁLISE E FILTRO DE SENSIBILIDADE

Inicialmente todos os elementos são considerado sólidos, e a alteração dos elementos de sólido para vazio, ou vice-versa, é determinado após o cálculo do valor de sensibilidade do elemento,  $\alpha_e$ , que para o caso do método BESO baseado nas tensões equivalentes de Von Mises é dado por:

$$\alpha_e = (x_e)^{p-1} \sigma_e^{vm} \quad (4.3.1)$$

Onde  $\sigma_e^{vm}$  é a tensão equivalente de Von Mises do elemento:

$$\sigma_e^{vm} = \frac{\sqrt{(\sigma_{e1} - \sigma_{e2})^2 + (\sigma_{e2} - \sigma_{e3})^2 + (\sigma_{e3} - \sigma_{e1})^2}}{2} \quad (4.3.2)$$

Sendo  $\sigma_{e1}$ ,  $\sigma_{e2}$  e  $\sigma_{e3}$  as tensões principais do elemento. Se somente for calculada a sensibilidade irá ocorrer o chamado problema do tabuleiro, onde há uma variação entre elementos sólidos e vazios em determinadas regiões da geometria, conforme demonstrado na Figura 4.3.1

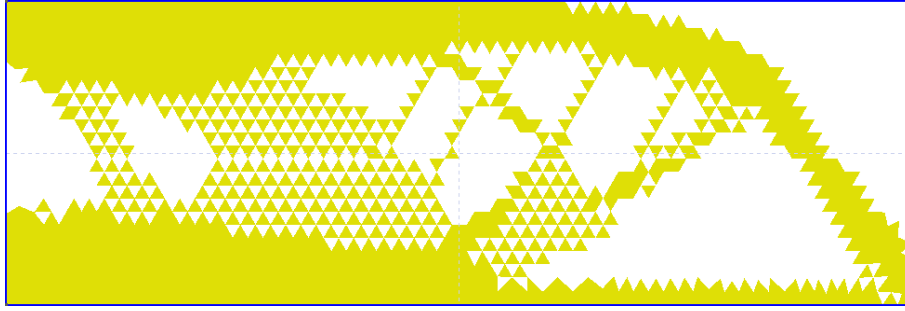


Figura 4.3.1. Problema do tabuleiro.

Este problema pode ser contornado utilizando um filtro que realiza uma média ponderada baseada na quantidade de elementos conectados a um mesmo nó, e na quantidade de vizinhos dentro de um raio mínimo. No início do filtro calcula-se a sensibilidade dos nós,  $\bar{\alpha}$ :

$$\bar{\alpha}_n = \sum_{e=1}^M w_e \alpha_e \quad (4.3.3)$$

Onde  $M$  é a quantidade de elementos conectados com o nó  $n$ , e  $w_e$  é o fator peso do elemento  $e$  dado por:

$$w_e = \frac{1}{M-1} \left( 1 - \frac{r_{en}}{\sum_{e=1}^M r_{en}} \right) \quad (4.3.4)$$

Sendo  $r_{en}$  a distância entre o nó  $n$  e o centróide do elemento  $e$ . Em seguida o valor de sensibilidade do elemento é atualizado conforme:

$$\alpha_e = \frac{\sum_{n=1}^K w(r_{en}) \bar{\alpha}_n}{\sum_{n=1}^K w(r_{en})} \quad (4.3.5)$$

Onde  $K$  é o número total de nós dentro de uma esfera limitada por um raio mínimo,  $r_{min}$ , e  $w(r_{en})$  é um fator peso obtido por:

$$w(r_{en}) = r_{min} - r_{en} \quad (4.3.6)$$

#### 4.4. ATUALIZAÇÃO DA GEOMETRIA E CONVERGÊNCIA

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

$$x_e = \begin{cases} 1, & \text{se } \alpha_e > \alpha_{th} \\ x_{min}, & \text{se } \alpha_e \leq \alpha_{th} \end{cases} \quad (4.4.1)$$

Onde o  $\alpha_{th}$  é o valor da raiz que pode ser obtido através de um algoritmo de bissecção, tendo como limites máximo e mínimos, o maior e o menor valor de sensibilidade, respectivamente. Em seguida é verificado se há convergência:

$$\frac{|\sum_{i=1}^N (C_{k-i+1} - C_{k-N-i+1})|}{\sum_{i=1}^N C_{k-i+1}} \leq \tau \quad (4.4.2)$$

Onde  $C$  é o valor da função,  $K$  é o número da iteração atual,  $K$  é um valor definido para que o critério de convergência ocorra somente após a estabilização do método, geralmente este valor é igual a dez, e  $\tau$  é o valor de tolerância da convergência.

## 5. PROGRAMA

### 5.1. *A LINGUAGEM PYTHON*

O Python é um intepretador, com uma linguagem de programação de alto nível, pois diversas etapas necessárias para a interpretação do programa pelo processador não são necessárias, estando essas etapas já incluídas através da sintaxe utilizada pelo usuário. Este tipo de linguagem é amplamente utilizada por engenheiros e cientistas já que não é necessário possuir um conhecimento da arquitetura do processador que está sendo utilizado.

Além de possuir uma linguagem muito atraente para os desenvolvedores de software, o Python permite a instalação de módulos e pacotes de terceiros. Esses pacotes e módulos são conjuntos de funções que podem ser importados para o algoritmo que está sendo desenvolvido, afim de reduzir a necessidade da criação de funções e rotinas que já foram programadas para solucionar um determinado problema.

### 5.2. *NUMPY, SCIPY E GMSH*

Para o desenvolvimento do programa utilizado para realizar a Otimização Topológica foram utilizados três pacotes adicionais, NumPy, SciPy e Gmsh.

O pacote NumPy traz diversas funções utilizadas para a criação e manipulação de arranjos, matrizes e objetos, como a indexação, ordenamento, entre outras funções pertinentes. Já o SciPy é responsável para a resolução de problemas científicos e matemáticos, possuindo inúmeras funções dos mais diversos campos de estudo, como por exemplo, a resolução de Sistemas de Equações e a estruturação de dados de particionamento de espaço do tipo árvore k-d.

O pacote Gmsh por sua vez é voltado exclusivamente para a criação de malhas, possuindo diversos algoritmos para realizar a discretização do domínio como o Delaunay, Frontal-Deulaunay e o HXT, e outras funções essenciais para um software capaz de realizar uma análise através do Método dos Elementos Finitos como a importação de geometrias, a manipulação e criação de geometrias, exportação da malha gerada em diversos formatos de arquivos. Além das funções, o pacote Gmsh

possuí uma interface gráfica que pode ser modificada de acordo com a necessidade do programa.

### 5.3. ORGANIZAÇÃO DO SOFTWARE

No capítulo 3 foi apresentado as seis etapas necessárias para a execução do Método dos Elementos Finitos e elas podem ser resumidas para o usuário do programa em um modelo de apenas três etapas:

1. Pré-processamento: Na primeira etapa o usuário deve alimentar o software com todos os dados necessários para a realização da análise, esses dados de entrada são os inputs, que no caso do Método dos Elementos Finitos são o domínio computacional discretizado e as Condições de Contorno. É comum a realização da discretização do domínio no mesmo programa em que são incluídas as Condições de Contorno;
2. Processamento: Após a inclusão de todos os inputs a análise será realizada através de uma Black Box. A Black Box um algoritmo desenvolvido que a partir dos inputs realiza todas as rotinas e cálculos necessários para a análise informada pelo usuário.
3. Pós-processamento: Com a conclusão da análise a Black Box prepara os resultados obtido para apresentar ao usuário de forma gráfica ou numérica.

Os três pacotes utilizados conseguem realizar todas as etapas informadas, com o Gmsh é possível discretizar um domínio e incluir as Condições de Contorno, bem como apresentar os dados obtidos. Com o NumPy e o SciPy é possível realizar todas as rotinas e cálculos necessários para a obtenção dos resultados desejados.

### 5.4. BLACK BOX

A construção da Black Box do programa foi dividida em duas partes, a primeira foi desenvolver um algoritmo capaz de realizar a análise através do Método dos Elementos Finitos, e a segunda foi desenvolver o algoritmo de Otimização Topológica.

A primeira parte foi desenvolvida com base no trabalho de Kattan P. (2008), onde é apresentado de forma objetiva diversas funções necessárias para conseguir realizar

diversas análises através do Método dos Elementos Finitos em diversos tipos de elementos e estruturas.

Para a segunda parte dois trabalhos foram essenciais para a construção do algoritmo. O primeiro trabalho utilizado como base foi o de Sigmund O. (2001), onde é apresentado um código desenvolvido para Matlab capaz de realizar a Otimização Topológica através do método SIMP (Solid Isotropic Material Penalization). Com base nesse trabalho, X. Huang, Y. M. Xie (2010) modificaram este código para aplicar uma variação do método BESO com interpolação de material com penalização, e com base nestes dois códigos a parte da Black Box referente à otimização foi construída.

Para redução do custo computacional durante o ciclo de otimização, foi observado que durante a análise do filtro de sensibilidade pode ser identificado que apenas a sensibilidade do elemento é variável durante o processo de otimização, logo é possível construir matrizes para pré-alocar esses valores constantes, e assim o ciclo da otimização realiza menos operações, reduzindo o tempo de execução em cada iteração.

A conversão dos códigos desenvolvidas em Matlab por Kattan P. (2008), para o Python pode ser realizada sem grandes problemas com o auxílio dos pacotes Numpy e SciPy. Uma dificuldade encontrada foi no custo computacional, principalmente na construção da Matriz de Rigidez Global, que é muito pouco eficiente devido aos inúmeros ciclos de cálculos para cada elemento. A solução utilizada para resolver este problema foi da substituição dessa rotina pelo uso de uma Matriz Esparsa, como demonstrado por Shure, L. (2007), onde é apresentado um ganho muito significativo no tempo de construção da Matriz de Rigidez Global. Para construir uma Matriz Esparsa é necessário três matrizes do tipo linha, que são conhecidas como arranjo. Um arranjo contém os índices das colunas da matriz, um outro arranjo contém os índices das linhas da matriz, e o último arranjo contém os valores referentes aos índices.

Um outro problema está na automatização das rotinas, para que o usuário apenas realize os inputs, sem que ocorra a interação dele com a Black Box. Para isto foi desenvolvida uma lógica para a inclusão das Condições de Contorno utilizando as funções de Grupos Físicos presente no pacote Gmsh, onde é possível atribuir um nome à um determinado grupo de pontos, arestas, faces ou volumes, e através desses nomes foi construído um algoritmo que identifica os nós, os valores o tipo de Condição de Contorno.



Apesar do foco deste trabalho é a realização de otimização de um sólido tridimensional com uma malha composta por elementos tetraédricos, foi implementado no programa a capacidade de resolver problemas bidimensionais com elementos triangulares, afim de comparar os resultados obtidos com os resultados presentes na literatura.

### 5.5. INTERFACE GRÁFICA

A interface gráfica do pacote Gmsh modificada é apresentada na Figura 5.4.1.

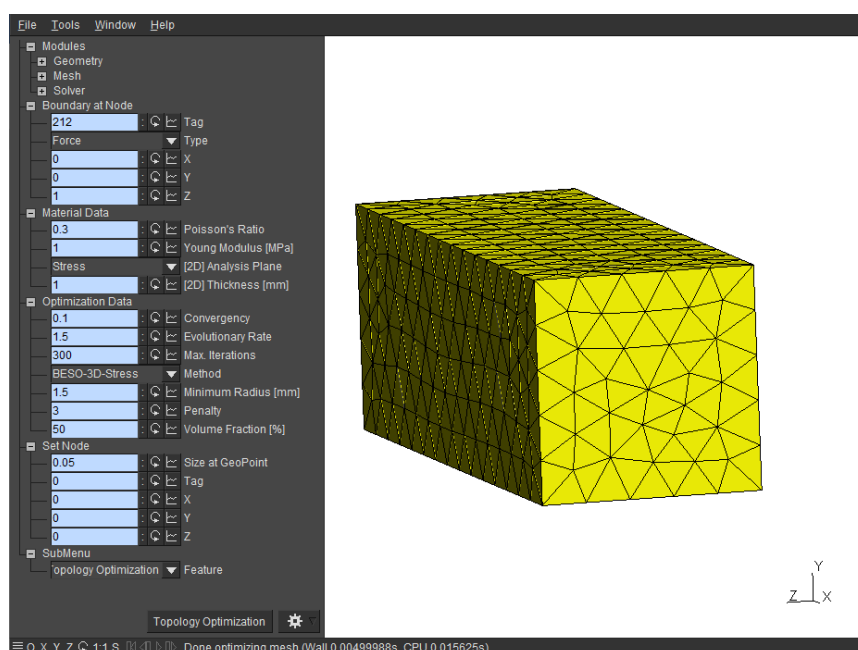


Figura 5.5.1. Interface gráfica modificada.

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

A inclusão das condições de contorno deve ser realizada antes da geração da malha, utilizando as funções de grupos físicos do pacote Gmsh, pois com estas funções é possível obter todos os nós pertencentes à uma determinada região, e com isto as condições podem ser facilmente atribuídas, conforme a Figura 5.5.2. Para aplicar as condições de contorno deve-se:

1. Escolher o tipo de grupo físico (ponto, curva, face ou corpo);

2. Selecionar os elementos físicos da geometria que pertencem ao mesmo tipo de grupo físico;
3. Agora deve ser decidido o tipo da condição de contorno:
  - a. Para uma condição de deslocamento, o nome do grupo físico deve ser *displacement\_x\_a\_y\_b\_z\_c*, onde *a* é o valor correspondente à componente no eixo x, *b* é o valor correspondente à componente no eixo y, e *c* é o valor correspondente à componente no eixo z, no caso 2D o valor de *c* não possui nenhum efeito nos cálculos;
  - b. Para uma condição de força, o nome do grupo físico deve ser *force\_x\_a\_y\_b\_z\_c*, onde *a*, *b* e *c* são os valores correspondentes às componentes nos eixos x, y, e z, respectivamente;
  - c. Para informar uma condição de que os elementos em contato com os nós do grupo físico não podem ser removidos, o nome do grupo deve ser *fixel\_xx*, onde *xx* pode possuir qualquer valor alfanumérico para organizar os grupos de elementos fixo.

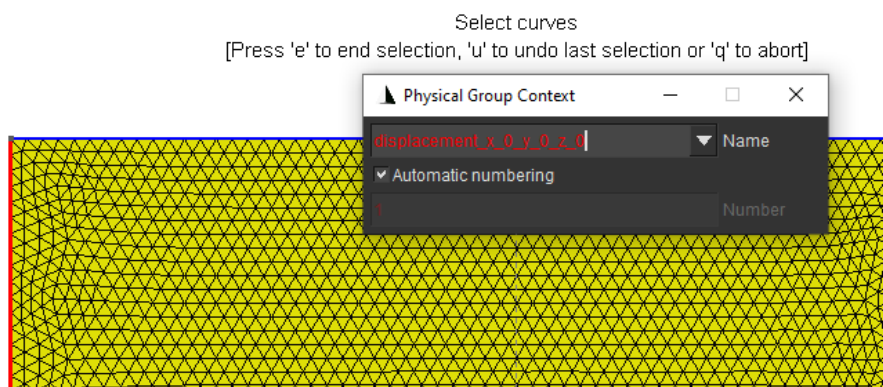


Figura 5.5.2. Aplicação das Condições de Contorno.

Caso seja necessário realizar a inclusão de uma condição de contorno em um nó específico, pode ser utilizada a função *inspect*, para descobrir qual o índice do nó, e em

## 5.6. TESTES

Diversos testes foram efetuados para verificar o funcionamento do programa, vale ressaltar que devido à utilização de elementos tetraédricos para a discretização do domínio computacional, as geometrias obtidas em diversos casos não foram simétricas, mas possuem características que estão presentes nas geometrias apresentadas na literatura que foi utilizada como base para o desenvolvimento do programa.

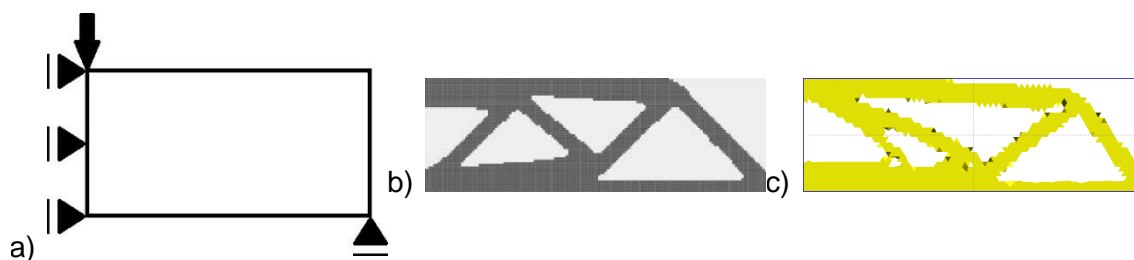


Figura 5.6.1. a) Condições de Contorno para uma estrutura do tipo MBB; b) Geometria obtida por X. Huang, Y. M. Xie (2010); c) Geometria obtida pelo programa.

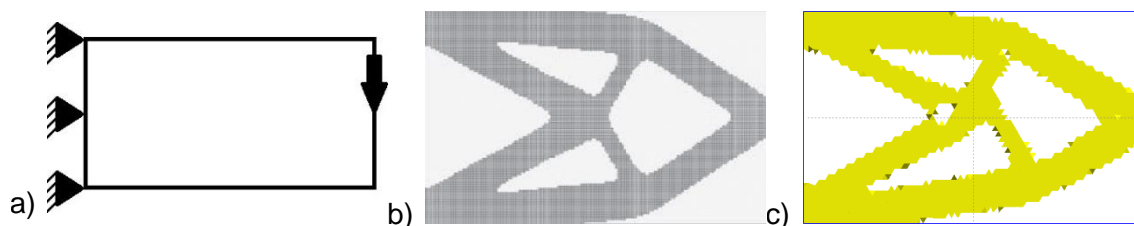


Figura 5.6.2. a) Condições de Contorno para uma estrutura do tipo viga engastada; b) Geometria obtida por X. Huang, Y. M. Xie (2010); c) Geometria obtida pelo programa.

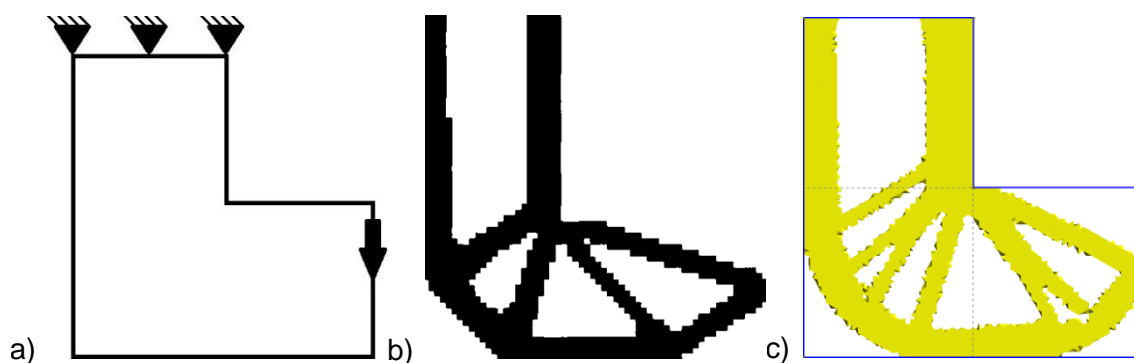


Figura 5.6.3. a) Condições de Contorno para uma estrutura do tipo L; b) Geometria obtida por Bendsøe M. P., Sigmund O. (2003); c) Geometria obtida pelo programa.

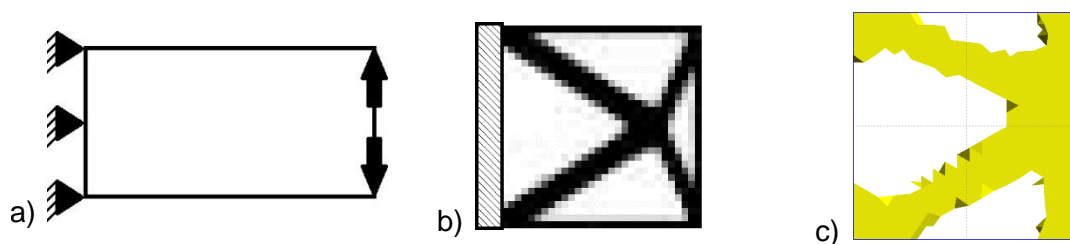


Figura 5.6.4. a) Condições de Contorno para uma estrutura do tipo viga engastada com forças opostas; b) Geometria obtida por Sigmund O. (2001); c) Geometria obtida pelo programa.

## 6. BIELA

### 6.1. CONCEITOS BÁSICOS DE UMA BIELA

Os motores de combustão interna transmitem torque em seus eixos através da conversão da energia gerada através da queima de combustível e gases que ocorre dentro da câmara de combustão, em trabalho mecânico.

A biela é o componente mecânico responsável pela transformação do movimento de translação do pistão causado pela combustão na câmara de combustão do motor, para o movimento de rotação no eixo do virabrequim.

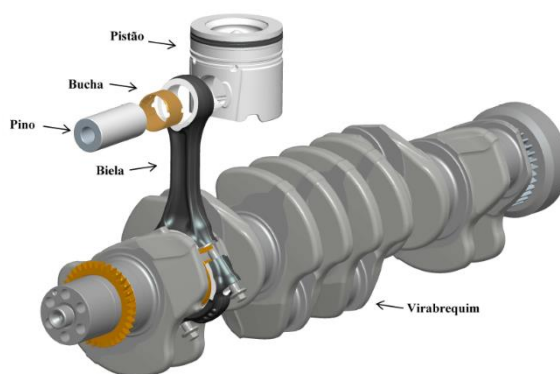


Figura 6.1.1. Conjunto pistão-biela-virabrequim do motor.

Fonte: Lima e Silva (2013).

A biela pode ser dividida em três regiões: olhal menor, olhal maior e alma. A biela convencional que será utilizada como base para as análises é apresentada na Figura 6.1.2 e suas dimensões na Figura 6.1.3.

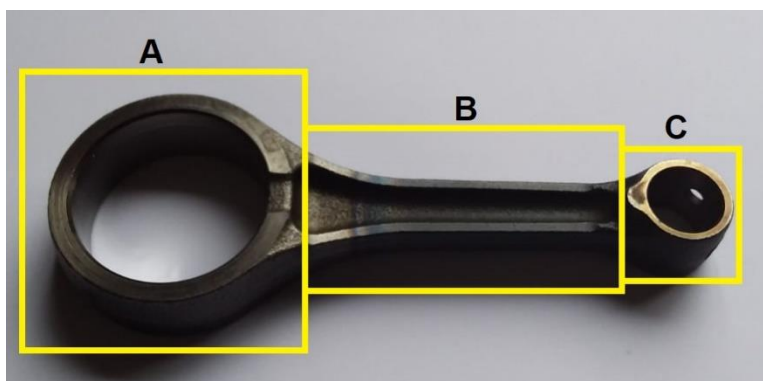


Figura 6.1.2. Biela dividida em três regiões: A) olhal maior; B) alma; C) olhal menor..

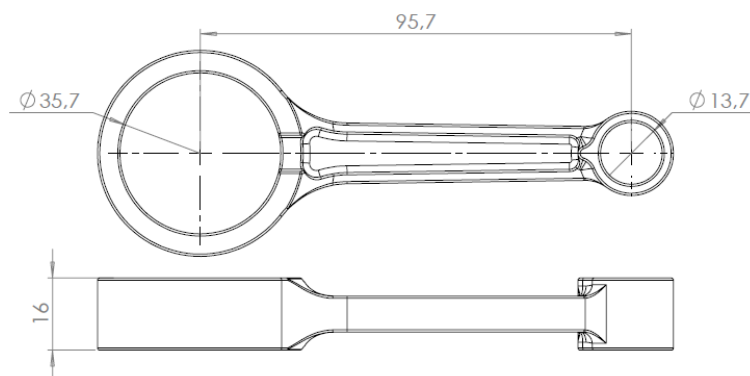


Figura 6.1.3. Dimensões da biela convencional utilizada.

## 6.2. ANÁLISE ESTÁTICA LINEAR DE UMA BIELA

No trabalho de Lima e Silva (2013), são realizadas três análises computacionais sobre uma biela, sendo uma análise para cada região da biela, neste trabalho será realizada apenas uma análise para avaliação da alma da biela, afim de comparar especificamente a diferença nos resultados que serão obtidos entre a biela otimizada e a biela convencional.

A análise será executada no software Ansys Student 2020 R1, simulando o contato do pino no olhal menor da biela, ou seja, a face interna do olhal menor será dividida com a finalidade de se aplicar uma carga distribuída em apenas uma região limitada, esta região está limitada à um arco de 120° partindo do centro do olhal menor. Os casos onde ocorrem as maiores tensões na alma da biela ocorrem durante a liberação de energia na câmara de combustão, gerando uma força de compressão na biela, sendo que esta força de compressão pode ter uma magnitude duas vezes maior do que no caso de maior tração, conforme os resultados de Lima e Silva (2013). O material a ser utilizado será o Structural Steel, um material padrão contido no Ansys Student 2020 R1. O volume da biela convencional tem aproximadamente 17670.19mm<sup>3</sup>.

Para a discretização da geometria, foi utilizado elementos tetraédricos, de segunda ordem, com um tamanho de elemento de 3mm. As Condições de Contorno estão demonstradas na Figura 6.2.1.

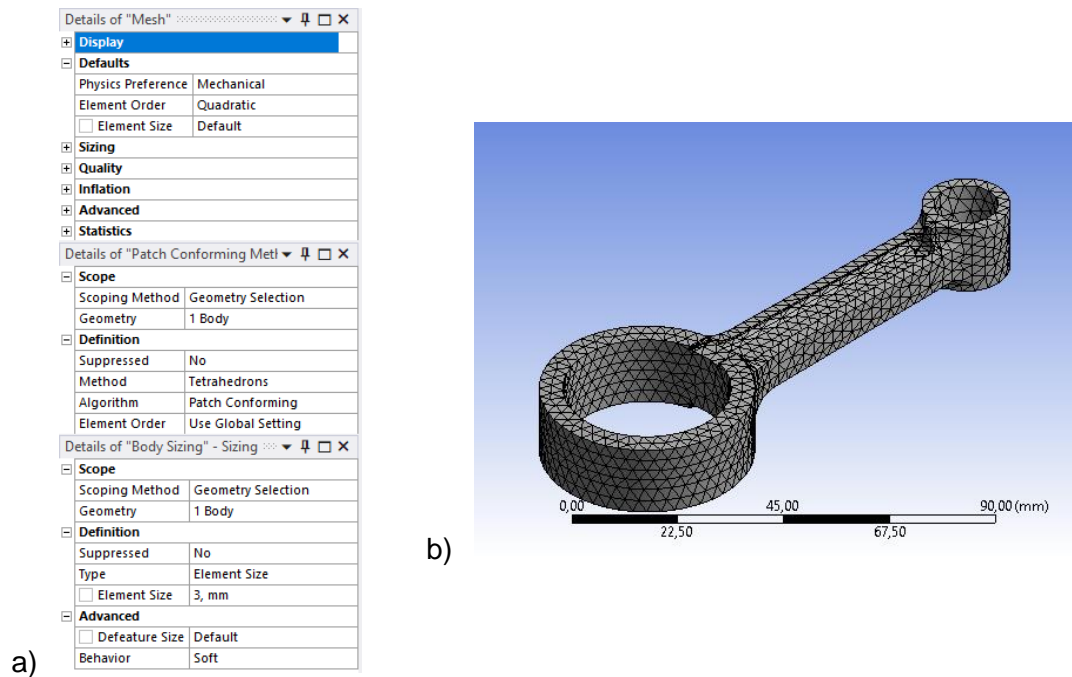


Figura 6.2.1. a) Parâmetros para a geração da malha; b) Malha utilizada para análise estática com 7371 elementos.

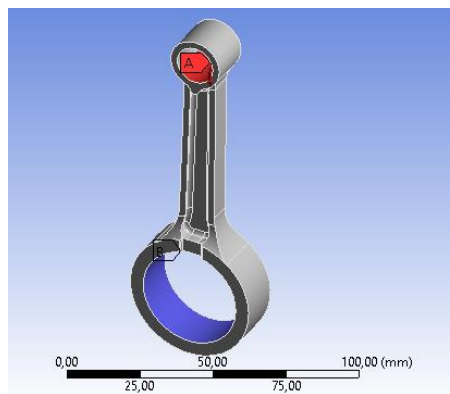


Figura 6.2.2. Condições de Contorno utilizadas para a análise estática: A) 1000N; B) Engaste.

### 6.3. OTIMIZAÇÃO TOPOLÓGICA DE UMA BIELA

Para realizar a Otimização Topológica, as mesmas Condições de Contorno foram aplicadas sobre o domínio computacional. Duas análises serão realizadas, uma para obter uma geometria que tenha o mesmo volume da biela convencional, e outra para obter uma geometria com o volume reduzido em relação à biela convencional. O volume do domínio computacional tem  $56364.42\text{mm}^3$ , porém para a análise da Otimização

Topológica foi utilizada uma condição de simetria, sendo necessário utilizada apenas metade deste domínio.

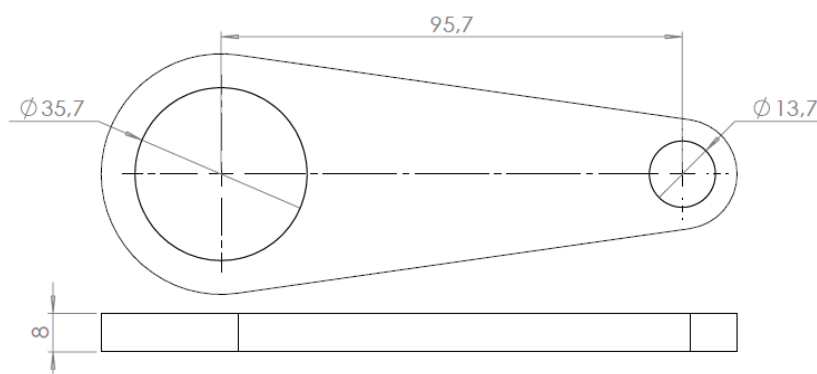


Figura 6.3.1. Dimensões do domínio computacional desenvolvido.

O percentual de volume que a biela convencional possui em relação ao domínio desenvolvido para a realização da Otimização Topológica, é aproximadamente 31%, sendo este valor, a fração de volume desejada para se obter uma geometria otimizada com o mesmo volume da geometria convencional. Além desse caso, foi realizada uma segunda análise utilizando um percentual menor para a obtenção de uma geometria mais leve do que a convencional.

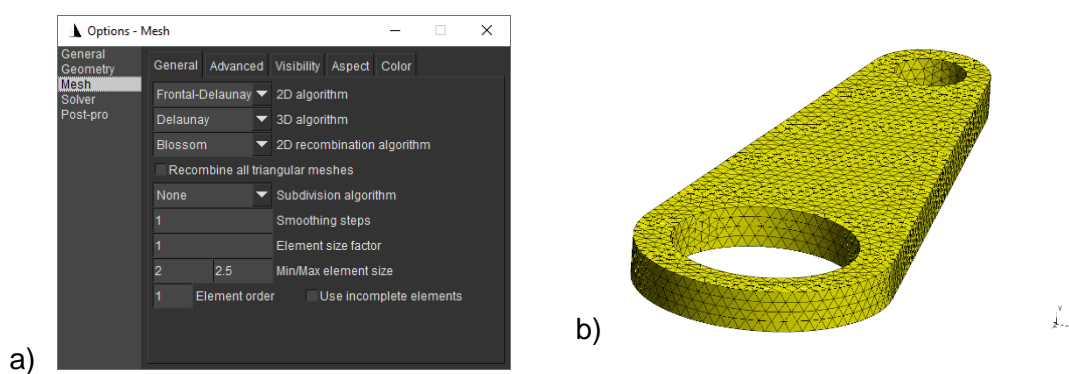


Figura 6.3.2. a) Parâmetros para a geração da malha; b) Malha utilizada para a realização da Otimização Topológica com 10717 elementos.

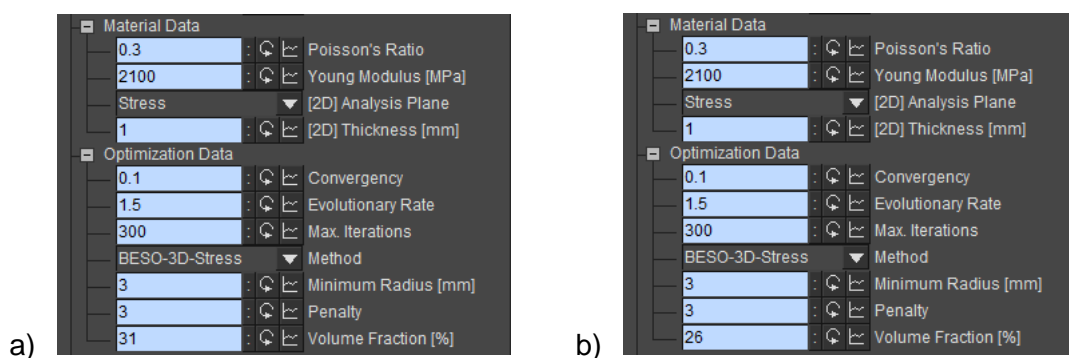


Figura 6.3.3. a) Parâmetros para obtenção de uma geometria com o mesmo volume da biela convencional; b) Parâmetros para obtenção de uma geometria com menos volume do que a biela convencional.

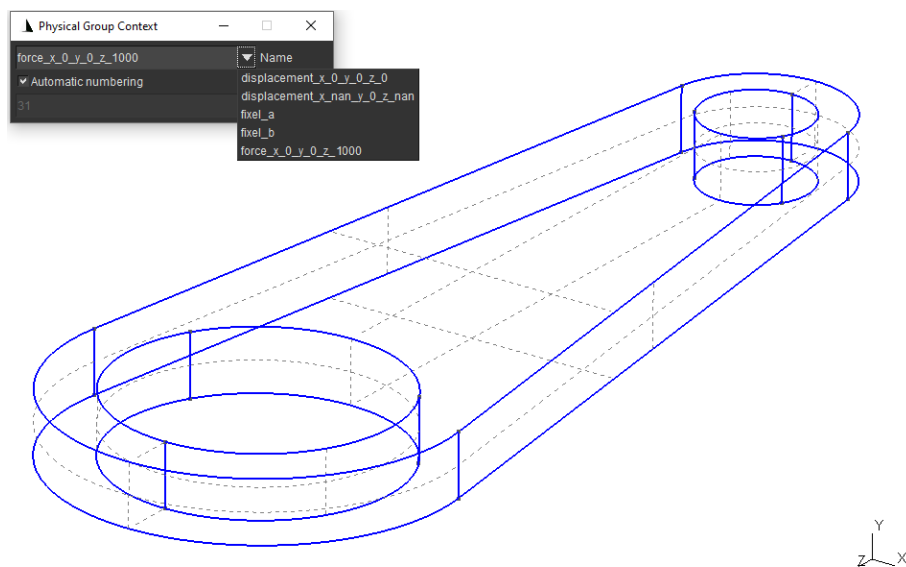


Figura 6.3.4. Condições de Contorno utilizadas para a realização da Otimização Topológica.



## 7. RESULTADOS

O cálculo para determinação de um percentual do volume do domínio computacional utilizado na Otimização Topológica em relação à biela convencional foi realizado através de uma regra de 3, conforme:

$$\begin{array}{l} 100\% \rightarrow 56364.42 \\ x \rightarrow 17670.19 \end{array} \quad x = \frac{17670.19 \cdot 100}{56364.42} = 31.35\%$$

### 7.1. RESULTADOS DA OTIMIZAÇÃO TOPOLÓGICA

#### 7.1.1. GEOMETRIA OBTIDA COM 31% DA FRAÇÃO DE VOLUME

A geometria obtida após o procedimento de Otimização Topológica pelo software desenvolvido é apresentada na Figura 7.1.1.1, esta geometria foi posteriormente refinada, e apresentou um volume de 17329.48mm<sup>3</sup>. É notável a diferença da geometria obtida com a biela convencional, possuindo uma região vazada na alma da biela, e a estrutura possui a mesma espessura do que os olhais.

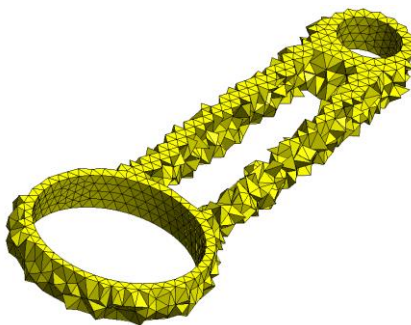


Figura 7.1.1.1. Geometria obtida após o processo de Otimização Topológica.

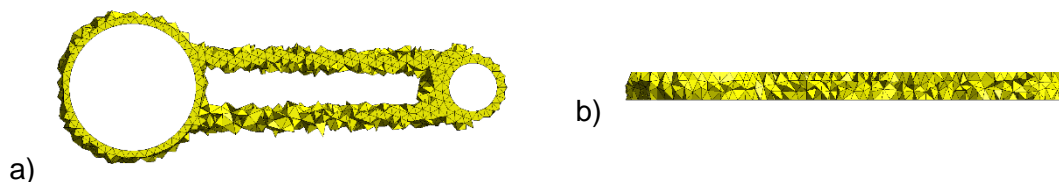


Figura 7.1.1.2. a) Vista superior da geometria obtida; b) Vista lateral.

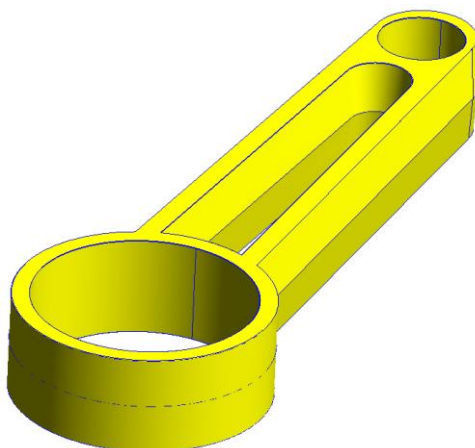


Figura 7.1.1.3. Geometria refinada.

#### 7.1.2. GEOMETRIA OBTIDA COM 26% DA FRAÇÃO DE VOLUME

A geometria obtida após o procedimento de Otimização Topológica pelo software desenvolvido é apresentada na Figura 7.1.2.1, esta geometria foi posteriormente refinada, e apresentou um volume de 14709.56mm<sup>3</sup>, houve uma redução de aproximadamente 16.7% do volume da biela convencional. Esta geometria manteve algumas características da obtida anteriormente, a redução do volume foi realizada com a diminuição da espessura da estrutura, e ao mesmo tempo houve um aumento de distribuição de material na região onde houve aplicação da carga distribuída.

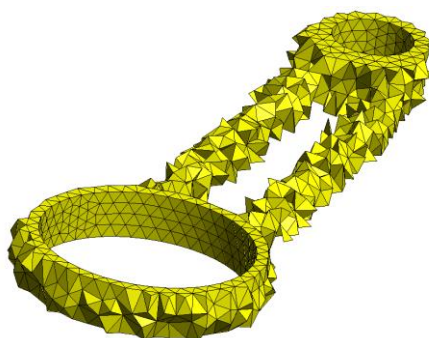


Figura 7.1.2.1. Geometria obtida após o processo de Otimização Topológica.

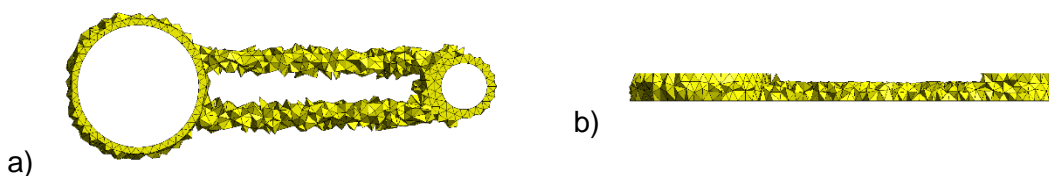


Figura 7.1.2.2. a) Vista superior da geometria obtida; b) Vista lateral.

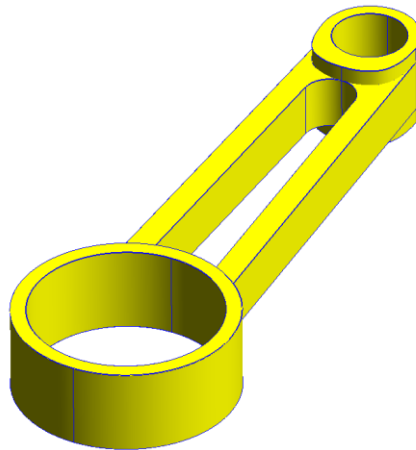


Figura 7.1.2.3. Geometria refinada.

## 7.2. RESULTADOS DA ANÁLISE ESTÁTICA LINEAR

As Condições de Contorno, e os parâmetros utilizados para a geração da malha, foi o mesmo nas três análises, conforme o Capítulo 6.2.

### 7.2.1. RESULTADOS DA BIELA CONVENCIONAL

A biela convencional apresentou na análise um deslocamento máximo de 0.0048mm na região do olhal menor, e uma tensão equivalente de Von Mises máxima de 20.9MPa bem localizada, abaixo do olhal menor. Pode ser observado que a alma da biela convencional se mantém tensionada em toda a sua extensão, bem como os valores mais elevados de energia interna.

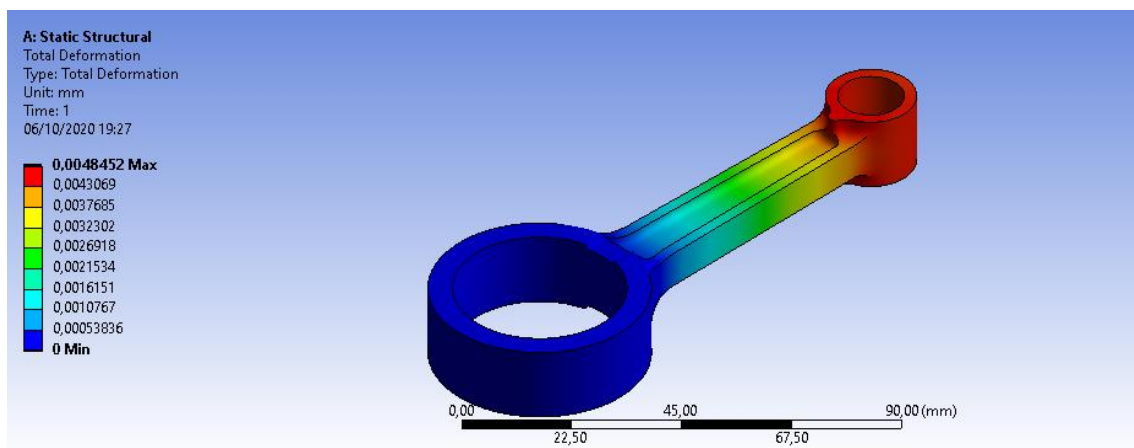


Figura 7.2.1.1. Deslocamento.

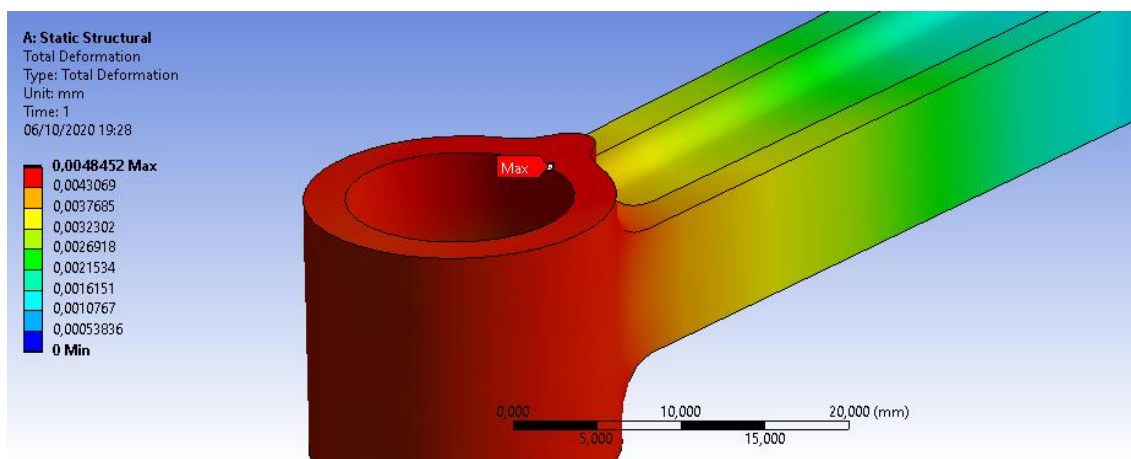


Figura 7.2.1.2. Deslocamento Máximo.

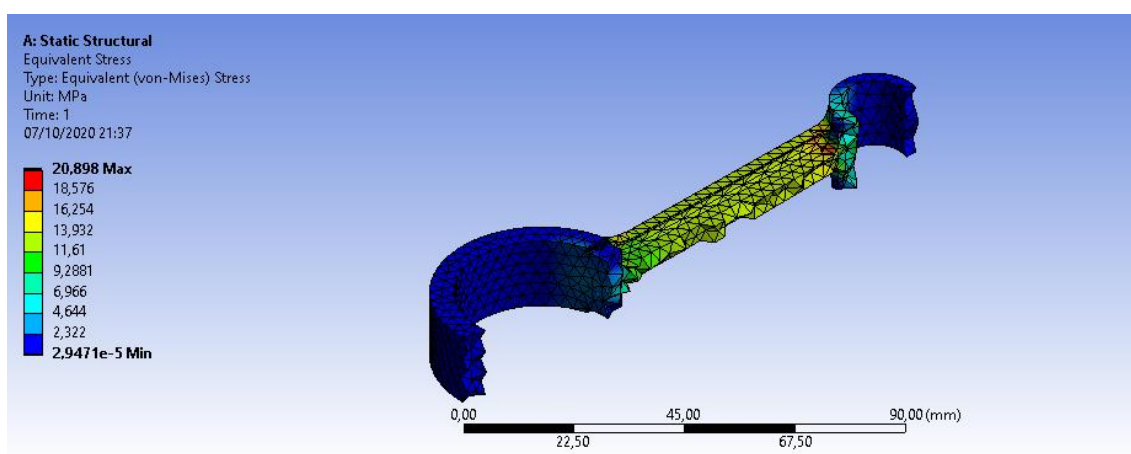


Figura 7.2.1.3. Tensões equivalente de Von Mises.

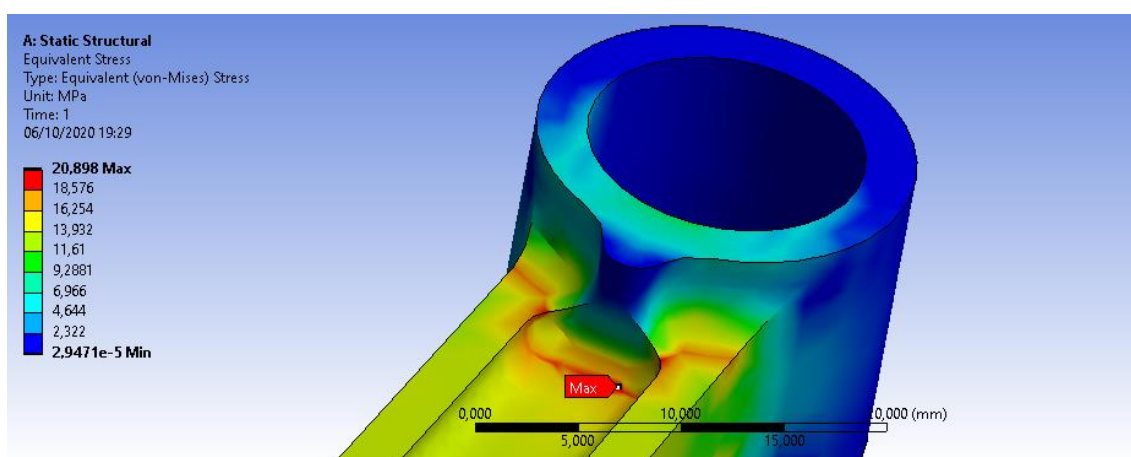


Figura 7.2.1.4. Máxima tensão equivalente de Von Mises.

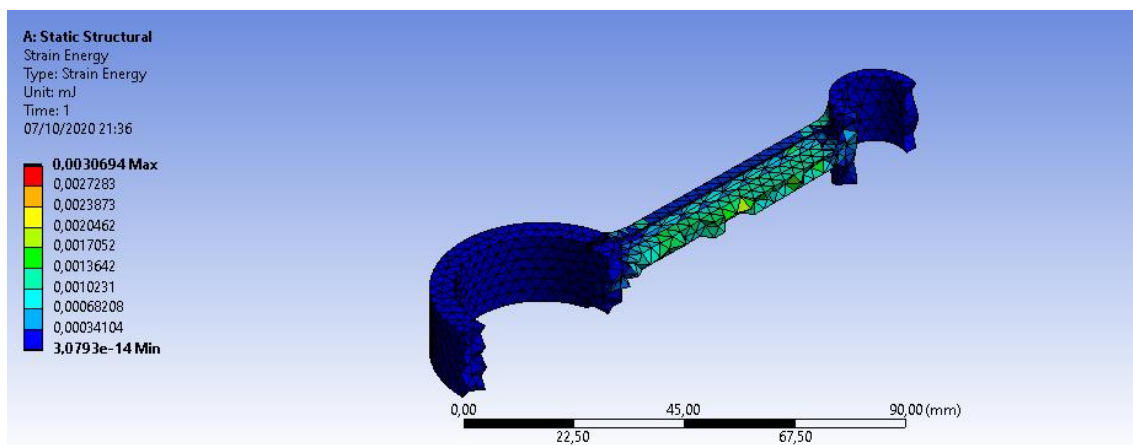


Figura 7.2.1.5. Energia interna.

## 7.2.2. RESULTADOS DA BIELA OTIMIZADA COM 31% DE FRAÇÃO DO VOLUME

A biela otimizada com o volume próximo ao da biela convencional apresentou na análise um deslocamento máximo de 0.0039mm na região onde ocorreria parte do contato entre o pino e o olhal menor, e uma tensão equivalente de Von Mises máxima de 32.24MPa na mesma região. Pode ser observado uma diminuição nos valores de tensão e da energia interna ao longo de toda a estrutura da biela otimizada em relação à biela convencional.

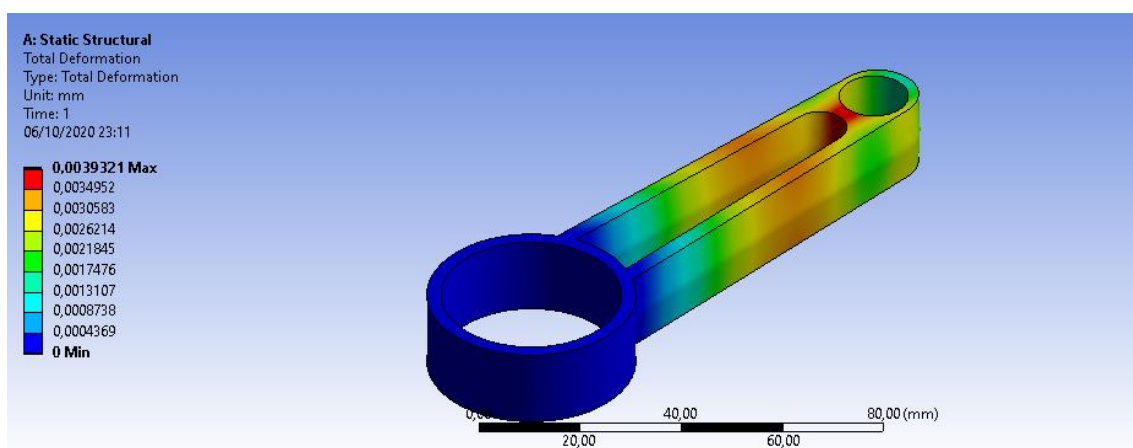


Figura 7.2.2.1. Deslocamento.

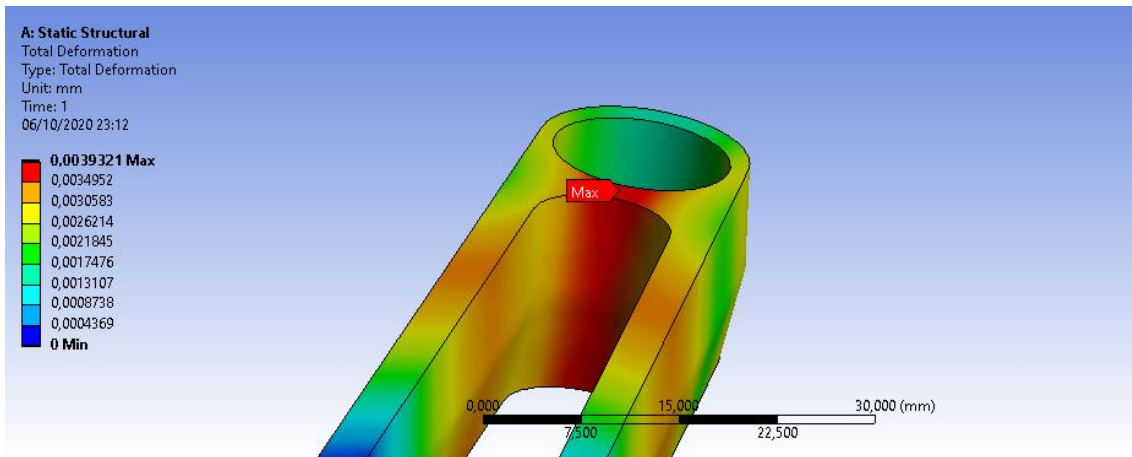


Figura 7.2.2.2. Deslocamento Máximo.

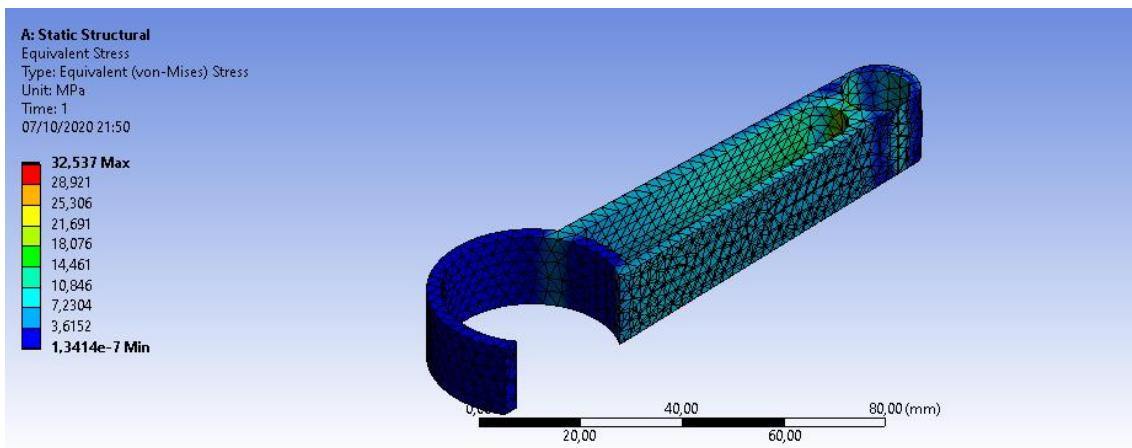


Figura 7.2.2.3. Tensões equivalente de Von Mises.

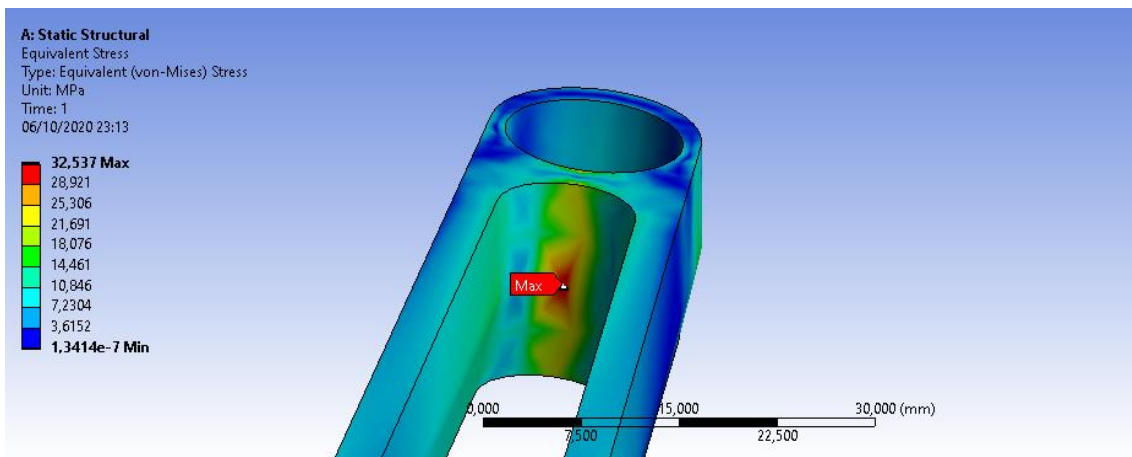


Figura 7.2.2.4. Máxima tensão equivalente de Von Mises.



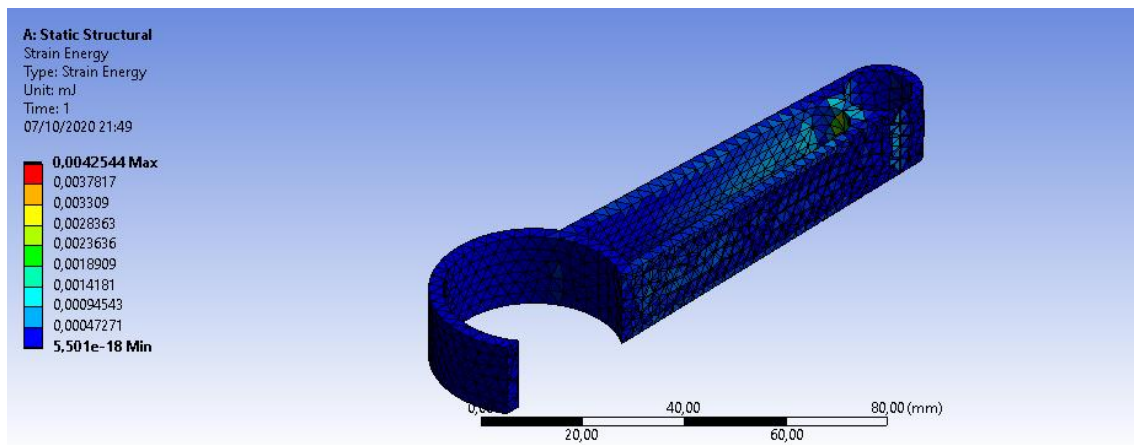


Figura 7.2.2.5. Energia interna.

### 7.2.3. RESULTADOS DA BIELA OTIMIZADA COM 26% DE FRAÇÃO DO VOLUME

A biela otimizada com menor volume apresentou na análise um deslocamento máximo de 0.0049mm na região do olhal menor, e uma tensão equivalente de Von Mises máxima de 27.46MPa na região mesma região. Pode ser observado que mesmo sendo reduzida de forma significativa a quantidade de material, o deslocamento máximo é muito próximo ao da biela convencional. A maior diferença está na região onde há a maior tensão.

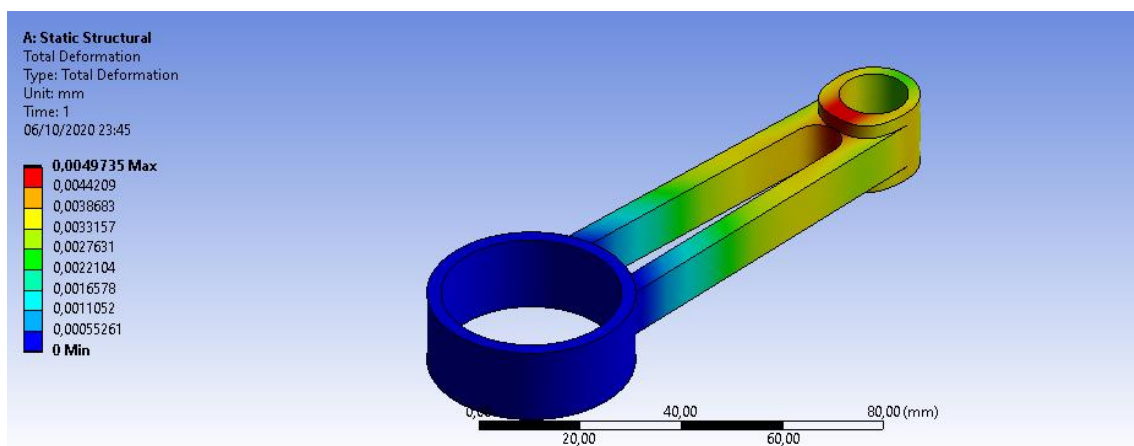


Figura 7.2.3.1. Deslocamento.

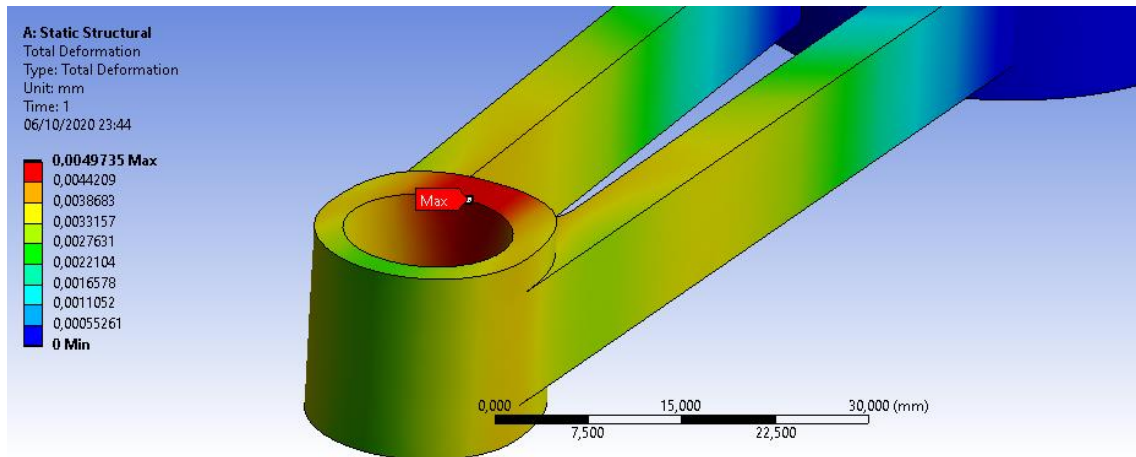


Figura 7.2.3.2. Deslocamento Máximo.

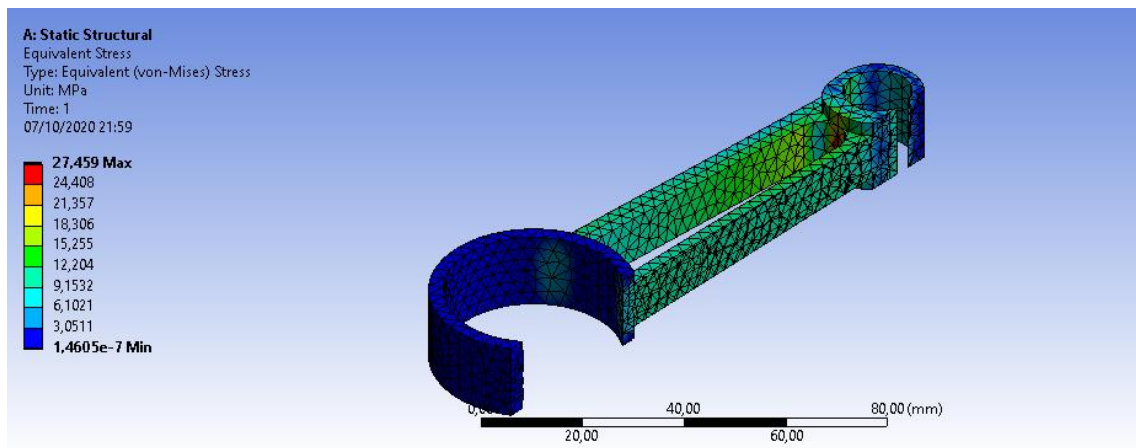


Figura 7.2.3.3. Tensões equivalente de Von Mises.

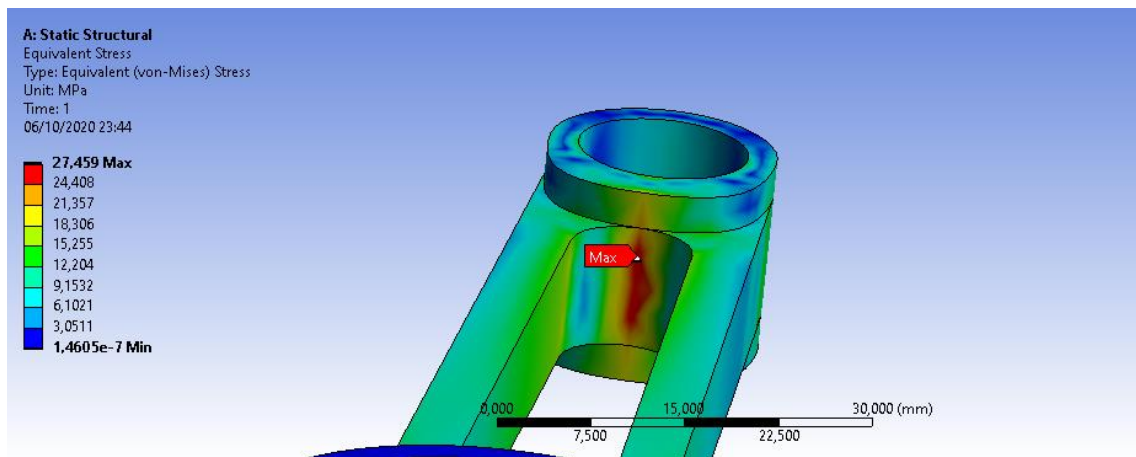


Figura 7.2.3.4. Máxima tensão equivalente de Von Mises.



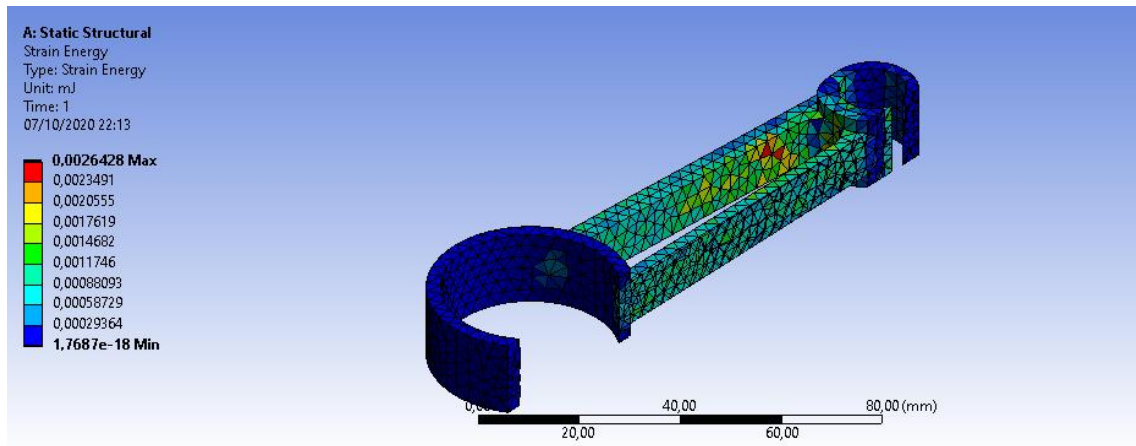


Figura 7.2.3.5. Energia interna.

## 8. CONCLUSÕES

Utilizando uma linguagem de alto nível como o Python, é possível desenvolver um software com a capacidade de realizar análises através do Método dos Elementos Finitos, incluindo nele diversas funções comuns em softwares comerciais.

A utilização da Otimização Topológica para o desenvolvimento de produtos mostrou ser uma ferramenta eficiente, apresentando conceitos finais da geometria final do produto que podem possuir melhorias em relação aos conceitos convencionais. A realização de uma comparação dos conceitos obtidos após o procedimento de otimização com os conceitos convencionais é importante para validação, tornando possível verificar se realmente há uma ou não melhoria no desempenho mecânico.

As bielas otimizadas tiveram um desempenho superior nas análises de compressão realizadas em relação à biela convencional, ficou evidente a redução da energia interna, e dos deslocamentos na biela otimizada com mesmo volume da biela convencional. Já no caso da biela com volume reduzido, os resultados obtidos foram próximos ao da biela convencional, sendo este caso uma demonstração de como a Otimização Topológica pode influenciar no projeto final de um produto.

Apesar da biela otimizada com o mesmo volume da convencional apresentar uma tensão equivalente de Von Mises máxima maior, é visível uma redução dessas tensões em toda a sua extensão, inclusive ao se analisar a biela otimizada com menor volume, a tensão equivalente máxima de Von Mises é menor, devido ao aumento de material na região onde o valor de tensão máxima está presente.

## REFERÊNCIAS BIBLIOGRÁFICAS

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

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

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

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

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

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

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

C. Geuzaine and J.-F. Remacle. (2009) Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, Volume 79, Issue 11, pages 1309-1331,

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

Lima e Silva, R. A. (2013) *Projeto Mecânico de Biela Automotiva Baseado em Otimização Estrutural*, Universidade Estadual de Campinas, Campinas

Harris, C.R., Millman, K.J., van der Walt, S.J. et al. (2020) Array programming with NumPy. *Nature* 585, 357–362 DOI: 10.1038/s41586-020-2649-2