



**UNIVERSIDADE ESTADUAL DA PARAÍBA  
CAMPUS VIII – ARARUNA  
CENTRO DE CIÊNCIAS, TECNOLOGIA E SAÚDE  
COORDENAÇÃO DE ENGENHARIA CIVIL**

**DIÓGENES LUIZ SILVA NASCIMENTO**

**OTIMIZAÇÃO TOPOLÓGICA EM JULIA ANALISANDO CASOS CLÁSSICOS DE  
VIGAS**

**ARARUNA – PB**

**2021**

**DIÓGENES LUIZ SILVA NASCIMENTO**

**OTIMIZAÇÃO TOPOLÓGICA EM JULIA ANALISANDO CASOS CLÁSSICOS DE  
VIGAS**

Trabalho de Conclusão de Curso apresentado à  
Coordenação do Curso de Graduação em  
Engenharia Civil da Universidade Estadual da  
Paraíba, como requisito parcial à obtenção do  
título de Bacharel em Engenharia Civil.

**Área de concentração:** Estruturas.

**Orientador:** Prof. Me. Anderson Viana do  
Nascimento.

**ARARUNA – PB**

**2021**

É expressamente proibido a comercialização deste documento, tanto na forma impressa como eletrônica. Sua reprodução total ou parcial é permitida exclusivamente para fins acadêmicos e científicos, desde que na reprodução figure a identificação do autor, título, instituição e ano do trabalho.

N244o Nascimento, Diogenes Luiz Silva.  
Otimização topológica em Julia analisando casos clássicos de vigas [manuscrito] / Diogenes Luiz Silva Nascimento. - 2021.  
73 p. : il. colorido.  
  
Digitado.  
Trabalho de Conclusão de Curso (Graduação em Engenharia Civil) - Universidade Estadual da Paraíba, Centro de Ciências, Tecnologia e Saúde , 2021.  
"Orientação : Prof. Me. Anderson Viana do Nascimento , Coordenação do Curso de Engenharia Civil - CCTS."  
1. Engenharia estrutural. 2. Análise de estrutura. 3. Viga. I.  
Título  
  
21. ed. CDD 624.1

**DIÓGENES LUIZ SILVA NASCIMENTO**

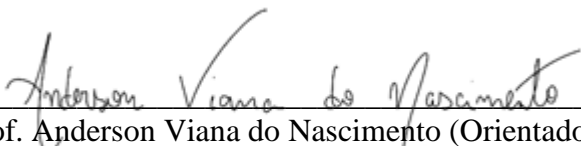
**OTIMIZAÇÃO TOPOLÓGICA EM JULIA ANALISANDO CASOS CLÁSSICOS DE  
VIGAS**

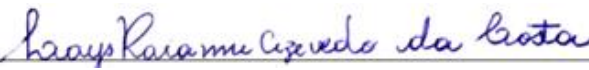
Trabalho de Conclusão de Curso apresentado à  
Coordenação do Curso de Graduação em  
Engenharia Civil da Universidade Estadual da  
Paraíba, como requisito parcial à obtenção do  
título de Bacharel em Engenharia Civil.

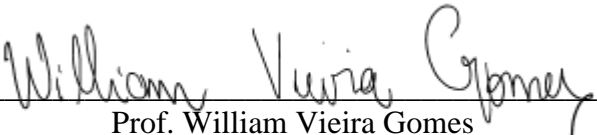
Área de concentração: Estruturas.

Aprovado em: 15/10/2021.

**BANCA EXAMINADORA**

  
Prof. Anderson Viana do Nascimento (Orientador)  
Universidade Estadual da Paraíba (UEPB)

  
Profa. Lays Raianne Azevedo da Costa  
Universidade Estadual da Paraíba (UEPB)

  
Prof. William Vieira Gomes  
Universidade Estadual da Paraíba (UEPB)

## **AGRADECIMENTOS**

Primeiramente, agradeço à Deus por sempre guiar meu caminho e me oportunizar a vencer mais esse desafio, me possibilitando condições físicas e intelectuais para a conclusão deste trabalho.

Ao meu orientador Prof. Me. Anderson Viana do Nascimento, pelo auxílio, disponibilidade de tempo, conhecimento e atenção prestada ao longo deste trabalho.

Agradeço aos colegas e amigos da engenharia por todos os momentos compartilhados, a todos os amigos conquistados nessa caminhada, com os apoios adquiridos, a jornada tornou-se mais branda.

Por fim, e não menos importante, agradeço a todos os familiares e pessoas próximas, que de alguma forma contribuíram e torceram para que eu vencesse mais essa etapa em minha vida, meu sincero agradecimento.

*“Nossa maior fraqueza é a desistência. O caminho mais certo para o sucesso é sempre tentar apenas uma vez mais.”*  
*(Thomas Alva Edison)*

## RESUMO

O presente trabalho tem por objetivo estudar problemas de engenharia estrutural através da otimização topológica. Ao otimizar o tamanho da estrutura, o principal objetivo é reduzir despesas e, simultaneamente, contemplar as exigências de segurança e especificações dos padrões convencionados. Nesse contexto, um método de otimização estrutural é introduzido para acelerar e possibilitar que uma solução possível e satisfatória seja obtida, proporcionando uma estrutura mais econômica. Porém, além dos benefícios econômicos, a utilização dessas tecnologias em projetos estruturais também pode analisar fatores relacionados ao impacto ambiental suscitado durante o procedimento e ao impacto ambiental gerado durante o processo após o projeto e a concepção estrutural. O desprovimento de recursos naturais e a crescente demanda por eles têm levado a comunidade científica a procurar por novas tecnologias para fornecer soluções eficazes na aplicação dos produtos produzidos a partir desses recursos. Entre essas tecnologias, o foco está na otimização, que tem sido acolhida com sucesso em diversos campos. Para isso, através da linguagem de programação Julia, aplica-se os métodos SIMP e BESO para resolução de dois problemas. Com base nisso, o presente trabalho tem por objetivo estudar problemas de engenharia estrutural através da otimização topológica através de dois problemas estruturais básicos. O primeiro trata de otimizar uma viga MBB, utilizando os dois métodos de otimização topológica citados. O segundo trata de uma viga engastada com carga distribuída na parte superior da extremidade livre. O caminho metodológico começa com uma revisão da literatura para compreender as questões relevantes e fundamentais na pesquisa de otimização de topologia (OT). São estudados o desenvolvimento de conceitos de OT, métodos SIMP e BESO. Tudo isso é feito adotando diferentes parâmetros de entrada, a fim de verificar a influência de cada parâmetro sobre o resultado final. Ao fim, os resultados mostram quais parâmetros de entrada são mais influentes e permitem avaliar a eficiência dos métodos utilizados, mostrando suas vantagens e dificuldades, e em todos os casos, a ferramenta provou ser capaz de se aproximar da topologia otimizada esperada.

**Palavras-Chave:** Otimização estrutural. Otimização topológica. Viga. SIMP. BESO.

## ABSTRACT

This work aims to study structural engineering problems through topological optimization. By optimizing the size of the structure, the main objective is to reduce expenses and, at the same time, meet the safety requirements and specifications of the agreed standards. In this context, a structural optimization method is introduced to accelerate and enable a possible and satisfactory solution to be obtained, providing a more economical structure. However, in addition to the economic benefits, the use of these technologies in structural projects can also analyze factors related to the environmental impact raised during the procedure and the environmental impact generated during the process after the design and structural conception. The lack of natural resources and the growing demand for them have led the scientific community to look for new technologies to provide effective solutions in the application of products produced from these resources. Among these technologies, the focus is on optimization, which has been successfully embraced in various fields. For this, using the Julia programming language, the SIMP and BESO methods are applied to solve two problems. Based on that, this work aims to study structural engineering problems through topological optimization through two basic structural problems. The first problem deals with optimizing an MBB beam, using the two mentioned topological optimization methods. The second problem deals with a crimped beam with load distributed on top of the free end. All this is done by adopting different input parameters in order to verify the influence of each parameter on the final result. At the end, the results show which input parameters are more influential and allow evaluating the efficiency of the methods used, showing their advantages and difficulties, and in all cases, the tool proved to be able to approach the expected optimized topology.

**Keywords:** Structural optimization. Topological optimization. Beam. SIMP. BESO.



## LISTA DE FIGURAS

<b>Figura 1-</b> otimização topológica-----	13
<b>Figura 2</b> - Domínio de projeto-----	16
<b>Figura 3</b> - Categorias de otimização estrutural. a) OD. b) OF. c) OT. À esquerda o domínio do problema e à direita a solução otimizada -----	17
<b>Figura 4</b> – Linha do tempo da OT sob a ótica da indústria -----	19
<b>Figura 5</b> – Aplicação da Fabricação Aditiva no Problema OT 3D -----	20
<b>Figura 6</b> – Procedimento típico de projeto estrutural por OT -----	24
<b>Figura 7</b> – Distribuição de material em um domínio de referência fixo -----	24
<b>Figura 8</b> – Tabuleiro de Xadrez -----	29
<b>Figura 9</b> – Benchmark, linguagem Julia-----	31
<b>Figura 10</b> – Viga MBB-----	33
<b>Figura 11</b> – Viga SC-----	35
<b>Figura 12</b> – Resultado da OT da solução encontrada por Sigmund (2001), e da topologia resultante com o uso da ferramenta criada dos métodos SIMP e BESO do Problema 1-----	36
<b>Figura 13</b> – Resultado da OT da solução encontrada por Sigmund (2001), e da topologia resultante com o uso da ferramenta criada dos métodos SIMP e BESO do Problema 2-----	37
<b>Figura 14</b> – Resultados do problema 1 para o método SIMP da viga MBB-----	38
<b>Figura 15</b> – Resultados do problema 1 para o método BESO da viga MBB-----	40
<b>Figura 16</b> – Resultados do problema 2 para o método SIMP da viga SC-----	41
<b>Figura 17</b> – Resultados do problema 2 para o método BESO da viga SC-----	42

## LISTA DE TABELAS

<b>Tabela 1</b> – Casos do problema 1, pelo método SIMP -----	34
<b>Tabela 2</b> – Casos do problema 1, pelo método BESO -----	34
<b>Tabela 3</b> – Casos do problema 2, pelo método SIMP -----	35
<b>Tabela 4</b> – Casos do problema 2, pelo método BESO -----	35
<b>Tabela 5</b> – Resultados do problema 1, pelo método SIMP -----	38
<b>Tabela 6</b> – Resultados do problema 1, pelo método BESO-----	39
<b>Tabela 7</b> – Resultados do problema 2, pelo método SIMP-----	40
<b>Tabela 8</b> – Resultados do problema 2, pelo método BESO-----	42

## LISTA DE ABREVIATURAS E SIGLAS

OT	Otimização topológica
SIMP	<i>Solid Isotropic Microstructure with Penalization</i>
ESO	<i>Evolutionary Structural Optimization</i>
ABNT	Associação Brasileira de Normas Técnicas
MEF	Método dos Elementos Finitos
DSA	<i>Design Sensitivity Analysis</i>
OD	Otimização dimensional
OF	otimização de forma
CAD	Desenho auxiliado por computador
AM	Manufatura Aditiva
OET	otimização estrutural em topologia
TSA	Topological Sensitivity Analysis
BESO	<i>Bi-directional Evolutionary Structural Optimization</i>
OC	Critério de otimização

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>12</b>
<b>1.1</b>	<b>Objetivo geral</b>	<b>14</b>
<b>1.2</b>	<b>Objetivos específicos</b>	<b>14</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>15</b>
<b>2.1</b>	<b>Otimização estrutural</b>	<b>15</b>
<b>2.2</b>	<b>Otimização Topológica</b>	<b>18</b>
<b>2.3</b>	<b>Método SIMP</b>	<b>26</b>
<b>2.4</b>	<b>Método BESO</b>	<b>28</b>
<b>2.5</b>	<b>Linguagem Julia de programação</b>	<b>30</b>
<b>3</b>	<b>METODOLOGIA</b>	<b>32</b>
<b>3.1</b>	<b>Método SIMP</b>	<b>32</b>
<b>3.2</b>	<b>Método BESO</b>	<b>32</b>
<b>3.3</b>	<b>Casos estudados</b>	<b>33</b>
3.3.1	Problema 1	33
3.3.2	Problema 2	34
<b>4</b>	<b>RESULTADOS E DISCUSSÃO</b>	<b>36</b>
<b>4.1</b>	<b>Validação dos códigos</b>	<b>36</b>
<b>4.1.1</b>	<b>Viga MBB</b>	<b>36</b>
<b>4.1.2</b>	<b>Viga SC</b>	<b>36</b>
<b>4.2</b>	<b>Comparativos</b>	<b>37</b>
4.2.1	Casos do problemas 1, método SIMP	37
4.2.2	Casos do problemas 1, método BESO	39
4.2.3	Casos do problemas 2, método SIMP	40
4.2.4	Casos do problemas 2, método BESO	41
<b>5</b>	<b>CONCLUSÃO</b>	<b>43</b>
<b>6</b>	<b>REFERÊNCIAS</b>	<b>44</b>
	<b>APÊNDICE A – Código SIMP para a Viga MBB</b>	<b>52</b>

<b>APÊNDICE B – Código SIMP para a viga SC</b> .....	<b>57</b>
<b>APÊNDICE C – Código BESO para a Viga MBB</b> .....	<b>62</b>
<b>APÊNDICE D – Código BESO para a Viga SC</b> .....	<b>68</b>

## 1 INTRODUÇÃO

Nas últimas décadas, a otimização das estruturas de edifícios tornou-se uma atividade crítica e desafiadora. O objetivo dos engenheiros é obter soluções ótimas para reduzir os custos do projeto de construção, ou seja, criar melhores projetos que reduzam despesas de recursos materiais e financeiros bem como o tempo (NAHRY; TIAHJONO; SATITI, 2013).

No entanto, projetos modernos possuem maior complexidade e abordam, frequentemente, grande número de variáveis e restrições. Essa complexidade inspirou inúmeros estudos em todo o mundo com o objetivo compartilhado de desenvolvimento de modelos para otimização dos problemas de engenharia civil (PRAYOGO; CHENG; PRAYOGO, 2017).

Otimização é o campo da ciência que busca conseguir a mais perfeita conformação para um agregado de incógnitas de projeto que apresentam uma assentada dificuldade. A otimização procura, por meio de concepções matemáticas, elevar ao máximo ou tornar mínimo uma função objetivo, que reproduz um determinado discernimento para aferir o melhor atributo para uma melhor solução. Dentre elas a otimização topológica é uma das metodologias que possibilita aos engenheiros descobrir respostas ótimas para elementos estruturais. Nas questões de engenharia, a função objetivo pode ser demonstrada por meios de termos de custo ou quantidade físicas, como a diminuição dos deslocamentos ou a maximização de como uma estrutura pode ficar rígida.

Dentre o que vai ser analisado nesse trabalho, as vigas são elementos estruturais de edifícios, são normalmente utilizadas em sistemas laje-viga-pilar para transferir a força vertical recebida da laje para o pilar ou para transferir cargas concentradas se for usado como suporte para o pilar. A viga transfere o peso das lajes e dos demais elementos (paredes, portas, etc.) às colunas. Este elemento estrutural é muito usado na maior parte das estruturas de concreto armado, como galpões, pontes, edifícios e várias edificações.

De modo geral, as vigas podem ser definidas como biapoiadas, quando possuem apenas um vão livre, e vigas contínuas, quando possuem vãos múltiplos. De acordo com (Mehta; Monteiro, 1994) devido o concreto ter uma resistência mínima em relação aos esforços de tração, resiste em torno de 10% em relação a compressão, o concreto armado é basicamente definido como a adição no concreto de armações longitudinais e transversais à face da viga para suprir o esforço de tração não resistido pelo concreto. Essas armações são vendidas em diâmetros comerciais e costumam ser chamadas de bitolas. Em problemas (OT), Figura 1, a incógnita do projeto é a semelhança entre os elementos estruturais e os nós. Isso ocorre devido à redistribuição e remoção de material de parte da estrutura.

**Figura 1:** otimização topológica

Fonte: Bendsøe e Sigmund (2002)

Várias técnicas foram desenvolvidas para otimização topológica, como por exemplo, o método de homogeneização (BENDSOE; KIKUCHI, 1988), o método do sólido isotrópico com penalização (SIMP do inglês *Solid Isotropic Microstructure with Penalization*) proposto por Bendsoe (1995) e método de otimização estrutural evolutiva (ESO do inglês *Evolutionary Structural Optimization*) proposto por Xie e Steven (1997).

No método de homogeneização, a solução é obtida sob a forma de material compósito perfurado. No entanto, como resultado, as estruturas não podem ser construídas diretamente, uma vez que nenhuma escala de comprimento está associada às microestruturas.

Nesse contexto a abordagem alternativa é um método SIMP no qual as propriedades do material são assumidas como sendo constantes dentro de cada elemento. Esses elementos são usados para discretizar o domínio do projeto e as variáveis são as densidades relativas do elemento. É necessária uma interpolação por lei da potência para penalizar as densidades intermediárias e obter-se soluções quase 0/1 de distribuição de material, onde 1 significa a existência total do material e 0 significa nenhum material (vazio). (HUANG E XIE, 2007).

O método ESO, por outro lado, baseia-se no conceito de que removendo lentamente materiais ineficientes, a estrutura evolui para um ótimo. A otimização estrutural evolutiva bidirecional (BESO do inglês *Bi-directional Evolutionary Structural Optimization*) é uma extensão do ESO que permite a agregação de materiais eficientes à estrutura ao mesmo tempo que os ineficientes são removidos (YANG *et al.*, 1999). Os métodos ESO/BESO, diferentemente do método SIMP, fornecem um perfil sem área “cinza” (intermediária), e portanto, mais claro e fácil de fabricar (TANSKANEN, 2002).

Com base no exposto, este trabalho analisou vigas clássicas empregando os métodos SIMP e BESO em dois casos clássicos de vigas, visando obter e comparar diferentes parâmetros de entrada.

## 1.1 Objetivo geral

O presente trabalho tem por objetivo resolver problemas de otimização estrutural empregando os métodos SIMP e BESO, em casos clássicos de vigas utilizando algoritmos em linguagem Julia.

## 1.2 Objetivos específicos

- Desenvolver o conhecimento do estudante em otimização topológica para estudar e aplicar as técnicas clássicas de otimização de topologia em Julia;
- Implementar códigos de otimização, em Julia, com o objetivo de encontrar a solução mais econômica que atenda às restrições de projeto e solicitações impostas aos elementos estruturais tratados;
- Verificar a eficiência dos programas implementados através da comparação dos seus resultados com os resultados gerados a partir de outros trabalhos de otimização topológica em casos clássicos de vigas.



## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 Otimização estrutural

A otimização estrutural se dedica em descobrir a configuração ótima da distribuição de massa estrutural dentro da região exequível do projeto que atenda aos padrões de projeto específicos. A disposição dos materiais no domínio muda, e as mudanças em determinados desempenhos mecânicos da estrutura são analisados, como flexibilidade, frequência natural, tensões, etc. Estas são chamadas de restrições de projeto, que podem ser iguais ou desiguais, (Santana 2002; Bendsøe e Sigmund 2003; e Arora 2004). Para esse fim, a otimização estrutural utiliza instrumentos computacionais para estabelecer metodicamente a configuração ótima. Portanto, considerando as condições de carga limite e propriedades específicas do material como restrições para avaliar a disposição do material da estrutura.

De acordo com Haftka e Gürdal (1991), a otimização compreende a realização por meio da minimização ou maximização, melhor para a decorrência final da função. Sua estrutura consiste em uma função objetivo integrando uma ou mais incógnitas de projeto, tendo potencial ou não de ser restringida por equações e / ou desigualdades que suprimem a superfície de busca (região viável).

Há mais de um século, Michell (MICHELL, 1904) proporcionou o estudo inicial de otimização estrutural, em que seu artigo consistia em encontrar a melhor solução para uma viga simplesmente apoiada com carga vertical empregue no meio do vão, em sua parte inferior. Atualmente, estruturas do tipo Michell são usadas em muitos estudos para apreciar artifícios de otimização. Embora a pesquisa de Michell possa remontar a mais de 100 anos, a otimização estrutural ainda é um campo de pesquisa recente. Apenas em 1989 o método SIMP Solid Bendsøe (1989) apresentou um procedimento de otimização topológica em um estudo que trata desse problema ainda chamado de método de abordagem direta, como um problema de distribuição de material e proporcionou múltiplos artifícios possíveis de otimizar uma estrutura.

A função objetivo exprime a finalidade a ser minimizado ou maximizado. Por exemplo, em termos de otimização estrutural, consegue exprimir a quantidade de material utilizado ou o preço dos elementos executadas. Variáveis de projeto são critérios que mudam interiormente no sistema no decorrer da otimização. Como exemplos de variáveis de projeto em engenharia estrutural são a área de reforço de aço, a área e / ou volume de concreto e o tamanho das peças estruturais que vierem a ser otimizados.

Restrições são funções que retratam circunstâncias ruins (desigualdade) ou impostas (igualdade). No momento de se abordar as restrições pertinentes ao concreto armado, esses são os valores limites para carga, momentos, área de armadura, altura da viga, etc.

No geral, pondera-se as limitações indicadas de um modelo característico. Por outro lado, a região de busca é o local que contém todas as respostas capazes para o problema, e seus alcances são limitados pela desigualdade e igualdade.

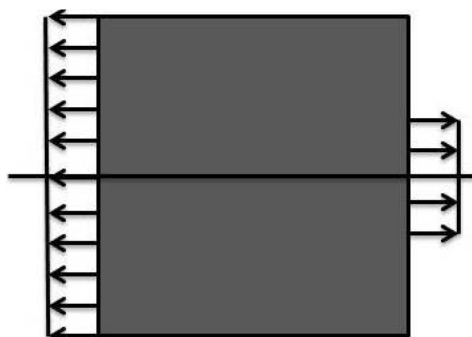
Já os modelos matemáticos utilizados para otimização podem ser lineares ou não lineares. Segundo Arora (2004), problemas que circundam funções objetivo e restrições lineares de otimização linear. Apesar de seu uso seja mais compreensível, sua empregabilidade a problemas de engenharia é limitada, pois a maioria possui funções objetivo e funções de restrição não lineares. Os modelos de otimização também há a possibilidade em ser divididos em duas categorias, são elas: métodos determinísticos e métodos probabilísticos.

Na abordagem determinística, o algoritmo de procura usa pelo menos a derivada inicial da função objetivo relacionada às variáveis de projeto. Portanto, torna-se um procedimento instantâneo e diminui as despesas computacionais, mas pode ser complexo descobrir a solução ótima global na função multimodal. Embora tenha suas restrições, esses procedimentos são competentes e de compreensível empregabilidade.

Por outro lado, os métodos probabilísticos são fundamentados em possíveis elucidações de descobertas "pseudo-aleatórias", devido as variáveis sejam originadas aleatoriamente do procedimento inicial de otimização e determinados dados sejam inventados randomicamente no processo, a solução de busca é guiada por regras probabilísticas (CARVALHO, 2014).

(BENDSOE; KIKUCHI, 1988) no final dos anos 1980 desenvolveram uma tecnologia pioneira na transformação de problemas de otimização de forma em problemas de distribuição, usa-se um método de homogeneização para processar o material. Com isso em mente, existem duas fases de composição: materiais e vazios. No estudo de (BENDSOE; KIKUCHI, 1988) considera uma estrutura que é adotada para lidar com o problema da elasticidade plana, sujeita a trações de superfície que é apontada na Figura 2.

**Figura 2:** Domínio de projeto



**Fonte:** Bendsoe e Kikuchi, 1988, p. 214.

Como posto por Oliveira Neto (2011), a distribuição ótima de material consiste em determinar quais regiões do domínio possuem material e quais estão vazias.

Para obter o projeto ótimo, avalie-o por meio da função objetivo, determinado por meio de um conglomerado de variáveis que representam o sistema estrutural, variáveis de projeto.

Matematicamente, o problema de otimização pode ser expresso do seguinte jeito:

Minimizar  $f(x)$  → Função custo ou função objetivo

Sujeita às  $\begin{matrix} g(x) \leq 0 \\ h(x) = 0 \end{matrix}$  → Restrições do problema (1)

Onde  $x \in R^t$ ,  $f \in R^r$  e  $h \in R^s$ , sendo,  $t$ ,  $r$  e  $s$  escalares que definem o número de variáveis, restrições de desigualdade e de igualdade.

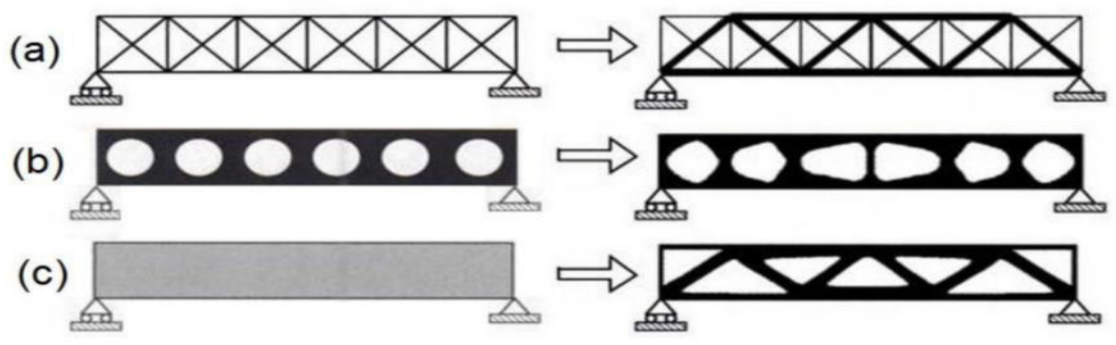
Estabelecendo o número de variáveis, desigualdades e escalares das restrições e de igualdade.

O objetivo da técnica de otimização é descobrir o valor extremo da função objetivo na região do projeto chamada da região viável ou factível. Esta região engloba pontos de design que satisfazem as restrições do problema e geralmente é dividida em restrições de igualdade e desigualdade.

Para alcançar uma estrutura ótima, geralmente são usados conhecimento de sensibilidade, sabendo-se que esta é a derivada da função objetivo associada a cada variável de projeto e as limitações atribuídas ao problema. Esta sensibilidade é muito importante porque após a análise, o algoritmo de otimização irá modificar as variáveis de projeto de acordo com seus próprios métodos.

De acordo com as variáveis de projeto, a otimização estrutural geralmente é dividida em paramétrica ou dimensional (OD), forma (OF) e topológica (OT). Na Figura 3, a estrutura alcançada pelos três tipos de otimização estrutural é mostrada esquematicamente.

**Figura 3:** Categorias de otimização estrutural. a) OD. b) OF. c) OT. À esquerda o domínio do problema e à direita a solução otimizada



Fonte: Bendsoe e Sigmund, 2003, p. 2.

A otimização paramétrica é uma camada que abrangem a otimização de dimensões, que procura a mais adequada resposta para parâmetros estruturais, como as características geométricas da seção transversal ou a disposição de cabos protendidos. Neste caso de otimização, a topologia é pré-definida e colada no procedimento, conseguindo ser autorizado a extinção de elementos estruturais, mas não é autorizada a invenção de novos elementos. A otimização de formas visa otimizar os delineamentos exteriores ou interiores de estruturas com uma topologia global determinada. As variáveis de projeto para este tipo de otimização são as coordenadas das regiões do contorno. Por fim, a Otimização de Topologia (OT) procura a melhor repartição de materiais em uma região espacial, onde a estrutura topológica da estrutura até este momento é uma incógnita (BENDSØE E SIGMUND, 2003).

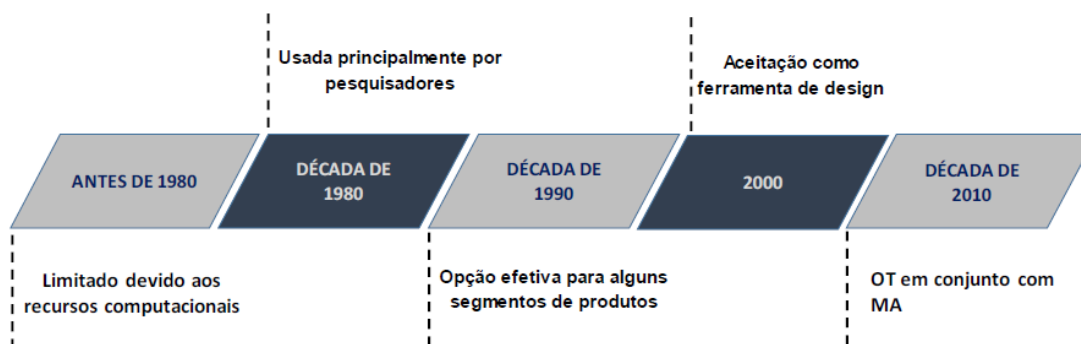
## 2.2 Otimização Topológica

A otimização da topologia é o meio mais comum de otimização estrutural, devido admitir que os elementos obtenham valores nulos para suas variáveis de projeto, portanto, a conexão dos nós da malha de elementos finitos seja modificável. Esta função admite que a topologia da estrutura seja modificada, enquanto a otimização da forma e da dimensão necessita conservar a conectividade de todos os seus elementos, para que se fique mais forte (BENDSØE; KIKUCHI, 1988; CHRISTENSEN; KLARBRING, 2009).

Os softwares iniciais para fins comerciais de OT foram desenvolvidos no final dos anos 80 e início dos anos 90. Entre esses softwares, pode-se citar o Optishape, o GENESIS, o OptiStruct, o MSC Nastran, entre outros.

Por causa das grandes diferenças na natureza dos algoritmos de otimização e suas especificações de implementação, não é fácil escolher o melhor procedimento para desenvolver softwares simples e didático, portanto para que os usuários não necessitem ser especialistas em teoria de otimização (VANDERPLAAST, 2006).

Baseado em Saitou *et al.* (2005), do ponto de vista da indústria, a otimização estrutural pode ser exibida em ordem cronológica para evidenciar as diferentes características dessa análise ao longo dos anos, conforme mostrado na Figura 4.

**Figura 4** – Linha do tempo da OT sob a ótica da indústria

**Fonte:** Pereira (2018)

- Antes de 1980: Por causa dos entraves de soluções computacionais, a solução numérica de problemas de otimização estrutural não era exequível.

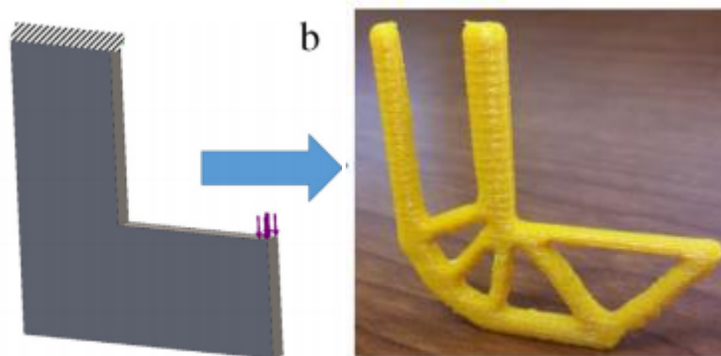
- Década de 1980: a análise estrutural tornou-se um instrumento de design iterativa / exploratória. Ainda que o aumento gradual do interesse, a otimização estrutural era na maioria um instrumento utilizado por pesquisadores.

- Década de 1990: Combinada com o Desenho auxiliado por computador (CAD), a otimização estrutural tornou-se um instrumento com um enorme potencial na análise de projetos. A otimização estrutural tornou-se uma escolha eficaz para determinados repartições de mercado de produtos.

- Década de 2000: a simulação estrutural foi usada no lugar dos testes físicos de certos tipos de produtos. A otimização estrutural está se tornando cada vez mais popular em aplicações industriais.

- Década de 2010: Graças à tecnologia de Manufatura Aditiva (AM), a otimização da topologia é frequentemente usada nos mais diversos campos. Em comparação com os processos de produção clássicas, a fabricação por meio da impressão 3D tem muitos benefícios. Na Figura 5, é apresentada uma empregabilidade simples deste procedimento adquirido por Mirzendehtel e Suresh (2016).

**Figura 5-**Aplicação da Fabricação Aditiva no Problema OT 3D



**Fonte:** Mirzendehtel e Suresh (2016)

A fundamentação matemática da otimização foi determinada e tem potencial ser empregada a diferentes campos, como dinâmica de fluidos, mecânica de sólidos, biomecânica, indústria automotriz e aeroespacial, etc., como Oktay, Akay e Merttopcuoglu (2011), trabalho de Kutuk e Gov (2013), Cavazzuti *et al.* (2011), Zhu, Zhang e Xia (2016), Zakhm, (2007), respectivamente.

OT é geralmente utilizada na fase de projeto conceitual para fornecer meios para novas estruturas inovadoras. Especialmente por meio de análise de otimização, como por exemplo, eles devem maximizar a rigidez da estrutura (minimizar a flexibilidade) ou minimizar o peso total (volume) da estrutura, mas reféns a determinadas restrições no volume total, rigidez total, deslocamento máximo e / ou tensão, etc. Geralmente, a discretização do domínio é realizada pelo método dos elementos finitos (BENDSOE; SIGMUND, 2003; OÑATE, 2009).

A literatura propõe duas categorias básicas de otimização da topologia, discreta e contínua, e existem diversos métodos em OET (otimização estrutural em topologia). Notadamente, para Emmendoerfer e Fancello (2014), otimização da topologia o método de estrutura contínua tem interessado a atenção de pesquisadores no caso de restrições de volume, a rigidez é maximizada.

Para Eschenauer e Olhoff (2001), a otimização contínua da topologia pode ser dividida em dois métodos: o denominado abordagem do material ou micro abordagem, e o método geométrico ou macro. No primeiro método, a seleção permite que a microestrutura porosa no domínio do projeto distribua de maneira uniforme o material. O segundo método envolve a colocação iterativa de adição ou exclusão de furos em pontos específicos do domínio.

O trabalho inicial para resolver o problema de otimização de topologia no plano de fundo de estruturas contínuas com restrições de tensão é Duysinx e Bendesoe (1998). O autor usa a apreciação de microestrutura SIMP, para “penalizar” alvos de densidade média.

Pereira, Fancello e Barcellos (2004) empregam algoritmos de otimização topologia com o principal foco de tornar mínimo as limitações de massa e tensão. Utilizando o mesmo método de material SIMP e função de penalidade de densidade medianas. O autor também utiliza uma função de regularização para o gradiente que foi projetado para eliminar a aparência de *checkerboard*. Os Métodos para a resolução de problemas de otimização não linear é o método Lagrangiano para limites de tensão envolvendo falha de material, utilizando o critério de falha de von Mises.

Além disso, Amstutz e Novotny (2010) empregaram um algoritmo de otimização de topologia para minimizar as restrições de massa e tensão usando um método de gradiente de topologia fundamentado no critério de falha de von Mises. Da mesma forma, o mesmo autor e Souza Neto (2012) utilizam o mesmo método, mudando apenas os critérios de falha de Drucker-Prager.

Guest e Smith Genut (2010) adotaram um método de otimização contínua da topologia para diminuir o número de variáveis de projeto. Seu método está no contexto de fórmulas contínuas tradicionais e fórmulas discretas usando algoritmos genéticos.

Bruggi e Duysinx (2012) aplicaram um algoritmo de otimização topológica que visa minimizar a qualidade por meio de restrições de tensão, incluindo o critério de falha de Drucker-Prager.

Holmberg, Torstenfelt e Klarbring (2013) debatem problemas de otimização em topologia desenvolvendo métodos de relaxamento de tensão. O autor usa o método fundamentado na norma  $p$  ou a função Kreisselmeier-Sinehauser (K-S) para reduzir o número de limites de tensão. O autor não considera a limitação uma medida local, mas global. Luo, Wang e Kang (2013) inseriram o parâmetro K-S para diminuir a região restrita no problema de otimização de topologia.

Qian (2013) propôs um método para alcançar a densidade ótima em um problema de otimização de topologia fundamentado na função B-spline, que limita a distribuição de densidade.

Guo, Zhang e Zhong (2014) utilizam o método de *level set* para exigir restrições ao valor extremo que apresenta a topologia ótima final da estrutura.

Clausen, Aage e Sigmund (2014) propuseram uma fórmula de otimização de estrutura topológica fundamentada em um método de combinação, que visa minimizar a flexibilidade e

restrições de volume por meio de um subproblema principal, e outro subproblema menor que visa minimizar a introdução de interferência em a estrutura quando a área estiver vaga.

Emmendoerfer e Fancello (2014) tratam do problema de qualidade mínima limitada pelo estresse de acordo com o critério de falha de von Mises. O autor emprega o método de conjunto de níveis para criar um algoritmo eficaz que pode identificar áreas de concentração de tensão e obter a melhor topologia.

Gao *et al.* (2015) empregam métodos de otimização na topologia e integrar com sistemas de estrutura de múltiplos componentes para impedir muitas restrições. O autor usa um método de agregação de restrição adaptativo baseado na função Kreisselmeier-Steinhauser (KS).

Duan, Li e Qin (2015) discutiram problemas de otimização na topologia de escoamento de fluidos, com o objetivo de obter melhor resolução de interface com mínimo custo computacional. Eles propuseram um método de refinamento adaptativo em otimização. O autor usa o método de critério ótimo para provar que é um método de problema de otimização eficaz e robusto com um grande número de variáveis de projeto. O método auxiliar é empregado à análise de sensibilidade.

Segundo Eschenauer e Olhoff (2001), a estrutura contínua da OT pode ser fragmentada em dois segmentos:

Aproximação por microestrutura (material): O procedimento OT é fundamentado na instauração da semelhança entre rigidez e densidade relacionadas ao domínio, tendo em vista que a densidade é capaz de assumir qualquer valor no entre 0-1, geralmente 0 (vazio) e 1 (material sólido) são usados. Entretanto, existe uma terceira alternativa, que é um elemento poroso de densidade média. Sabendo os valores das variáveis, densidades e algoritmos baseados em microestrutura visa descobrir o melhor arranjo de materiais para minimizar ou maximizar a função de custo. Alguns procedimentos pertencentes a este grupo incluem: SIMP, *homogeneous porous materials*, etc.

Aproximação por macroestrutura (geometria): Nesta situação, a topologia da estrutura é alterada inserindo furos no domínio. Como exemplo os métodos baseados na inserção de furos em sistemas contínuos, como o TSA-Topological Sensitivity Analysis e Bubble Method, que pode ser consultado em Pereira e Bittencourt (2010) e Eschenauer, Kobelev e Schumacher (1994). Na situação da introdução de buracos de furos em sistemas discretos, como principais métodos temos: métodos baseados em algoritmos evolutivos, como ESO e BESO.

Os resultados descobertos pelos dois métodos geralmente precisam ser explicados (suavização de contornos e definição de elementos sólidos) para que a estrutura seja executável



na edificação, pois o nível de refinamento da malha e mesmo a densidade intermediária possibilita a implicação às vezes impraticável de se fabricar. Por conseguinte, o esclarecimento dos resultados, uma nova apreciação da estrutura é necessária para averiguar se a mudança na topologia não atrapalha o desempenho da estrutura otimizada.

Pereira (2006) faz uma explanação sobre a OT em problemas de elasticidade envolvendo não-linearidade geométrica (grandes deslocamentos e rotações) e não-linearidade de material (hiperelasticidade não-linear quase-incompressível), aplicando o conceito de Análise de Sensibilidade Topológica (TSA) através de uma formulação Lagrangiana Total.

Porto e Pavanello (2007) estudaram a dos parâmetros da otimização estrutural topológica, fundamentado na teoria da homogeneização em seus resultados de otimização. Neste trabalho, uma única unidade de base quadrada com um vazio central retangular foi estudada, e suas propriedades mecânicas foram verificadas pelo método de homogeneização e MEF. O tensor elástico é descrito como para cada elemento finito do modelo estrutural global e obter a solução ótima do problema de minimização de energia potencial total através de um algoritmo iterativo baseado no critério de otimalidade.

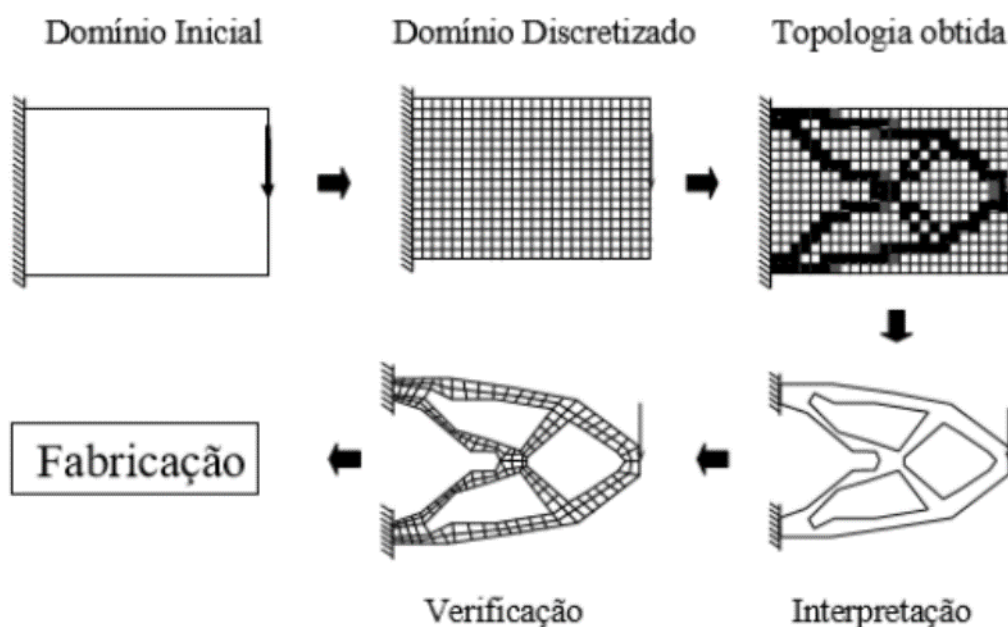
Rozvany (2009) expôs os atributos e fragilidades dos procedimentos de otimização de topologia mais conhecidos. Sigmund e Maute (2013) explanam sobre os desenvolvimentos atuais dos métodos de otimização de topologia mais importante.

A OT trabalha com determinados procedimentos de cálculo estrutural para alcançar o valor extremo da função objetivo e atender às restrições atribuídas ao problema. De acordo com (CHRISTENSEN; KLARBRING, 2009) esta é a maneira mais comum de otimização estrutural. A OT consente o uso de estruturas sem formas pré-estabelecidas, o que permite ao método descobrir arranjos inovadores e de alto desempenho (LIU; TOVAR, 2014). (SILVA, 2003) recomenda os subsequentes passos usando o projeto estrutural da OT, (Figura 6):

- Domínio inicial: O primeiro passo é determinar o domínio estrutural.
- Domínio discretizado: onde o domínio é discretizado em um número finito de elementos onde são executadas as condições finitas e de contorno;
- Topologia Obtida: relatar dados de domínio para software OT, a partir de um processo iterativo encontra-se a topologia que optimize a função objetivo. As regiões pretas retratam regiões com material e regiões brancas indica uma área sem material. Nesta topologia, as regiões também podem aparecer com escala de cinza, significa a existência de material intermediário;
- Interpretação do resultado: A imagem obtida pela OT precisa ser interpretada para que se obtenha o projeto final;

- Verificação de resultados: aqui você pode verificar a estrutura final usando MEF;
- Fabricação de estruturas: tecnologias como a prototipagem rápida podem ser usados para fabricar estruturas de formas complexas.

**Figura 6:** Procedimento típico de projeto estrutural por OT.

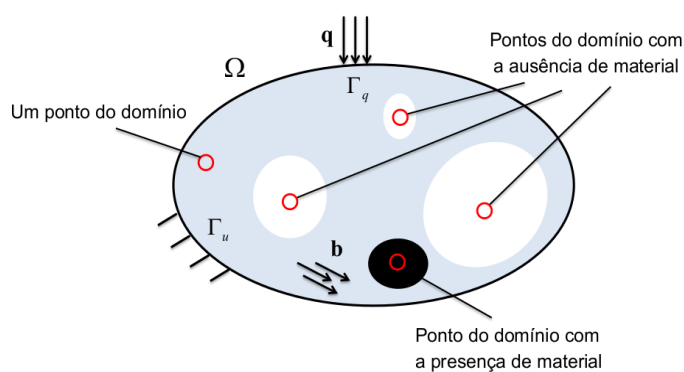


Fonte: Silva, 2003, p. 142.

Levando em conta um material mecânico, conforme mostrado na Figura 7, ocupando um domínio bidimensional, as propriedades de seu limite são:

$$\begin{aligned} \tau &= \tau_u \cup \tau_q, \\ \tau_u \cap \tau_q &= \emptyset \end{aligned} \quad (2)$$

**Figura 7:** Distribuição de material em um domínio de referência fixo.



Fonte: Chiesa, 2016

O principal foco da OT é descobrir um subdomínio de  $m$  que contenha material neste domínio ou nano-subdomínios sem materiais que atendam aos padrões estabelecidos, não há uma decisão a priori sobre sua conectividade (BECKERS, 1999). O formato da OT pode, compreensivelmente, resumir-se a um problema de distribuição de material, em um domínio fixo expansão, para definir em que pontos existe ou não existe material, a forma de configuração é um problema binário de 1  $\mp$  0. Do aspecto visual, a adversidade da OT pode ser retratada por uma imagem em preto e branco, onde o ponto  $(x; y)$  do domínio onde o material existe é definido por preto e o ponto  $(x; y)$  do domínio, e o local sem o material é determinado pelo branco. Os pontos  $(x; y)$  que contém ou não contém material no domínio pode ser representado pela função característica  $X(x; y)$ :

$$X(x,y) = \begin{cases} 1, se & (x,y) \in \Omega_m \\ 0, se & (x,y) \in \Omega \setminus \Omega_m \end{cases} \quad (3)$$

Levando em conta a má localização do problema de otimização, a existência do resultado pode ser alcançada por meio do "relaxamento" do problema (BENDSOE; KIKUCHI, 1988). O "Relaxamento" admite que a densidade assuma um valor intermediário entre 0 e 1. A inclusão desses valores pode ser executada selecionando um modelo de material. O modelo de material troca a função característica  $X(x,y)$  por uma função contínua. Por ser de fácil implementação e não aumentar o número de variáveis do problema, o modelo de material usado em larga escala na literatura é o modelo de material SIMP.

O principal foco do método OT é encontrar o layout ou estrutura topológica de uma quantidade fixa de material que atenda a certas restrições do projeto estrutural, com o objetivo de minimizar ou maximizar a função objetivo.

Como foi proposto por Eschenauer e Olhoff (2001), a OT pode ser dividido em dois tipos, discreto ou contínuo, obedecendo ao tipo de estrutura. Por exemplo, no caso de uma OT com uma estrutura modesta, o número total de barras, as devidas junções e sua posição são definidos. Além de analisar a maximização da rigidez, como visto em muitos trabalhos, este método pode ser executado para maximizar a frequência de ressonância, maximizar a carga de flambagem, minimizar a resposta de frequência da estrutura e maximizar a energia de impacto. A aplicabilidade da estrutura discreta de OT pode ser examinada no trabalho de Achtziger

(1997). Entretanto, a estrutura contínua da OT visa definir a presença ou ausência de materiais no domínio, seus respectivos formatos e a junção entre os elementos.

No vasto estudo dos métodos numéricos para OT, destacam-se dois métodos, nomeadamente o método SIMP e o método ESO, este último deu origem ao método BESO.

### 2.3 Método SIMP

O Método SIMP, a denominação do método de otimização de método direto SIMP foi usado pela primeira vez em um artigo de 1992 (ROZVANY *et al.*, 1992), que pesquisou a otimização da topologia sem homogeneização. O método SIMP é um dos procedimentos de OT mais conhecidas, inicialmente exposto por Bendsoe (1989) com o nome abordagem direta ou densidade artificial, e o nome SIMP foi inicialmente falado pelos autores Rozvany, Zhou e Birker (1992).

O principal foco desse método, é um procedimento que pode lidar com a existência de materiais por meio de materiais fictícios, e pode chegar a alcançar desempenhos intermediários entre o sólido e o vazio. Este procedimento pertence a um grupo de estrutura contínua com aproximação de microestrutura, devido que os algoritmos fundamentados em microestrutura se dispõem a descobrir o melhor arranjo de materiais a fim de minimizar ou maximizar a função objetivo.

O modelo é baseado na concepção de materiais com microestruturas isotrópicas e imaginárias. O conceito original do método SIMP foi inicialmente exposto por Bendsøe (1989), por conseguinte diversos trabalhos foram concebidos por meio deste conceito. Por exemplo, pode-se mencionar Bendsøe e Sigmund (1999), Fancello e Pereira (2003), Rozvany (2009), Bruggi e Duysinx (2012), Bruggi e Taliercio (2013) e Luo *et al.* (2013).

(ROSVANY, 2008) concluiu que o método SIMP necessita comparativamente um pequeno número de iterações e é apropriado para uma compatibilidade de várias condições de contorno, cargas e sistemas grandes (geralmente espaciais). Portanto, o método SIMP foi examinado quantitativamente, apresentando a convergência numérica da topologia de Michell. Kerdal e Nethercott (1984) apresentaram uma pesquisa sobre o modo de falha de vigas casteladas. Nesta pesquisa, os autores tentaram descobrir um método levemente não complexos para antecipar o modo de falha de vigas com aberturas na alma.

Os métodos mais comumente usados na otimização da topologia são fundamentados em métodos de densidade, que abrangem os mais conhecidos (SIMP). Os métodos fundamentados em densidade atuam em um domínio fixo, na maioria das vezes usando métodos de elementos

finitos (TAYLOR, ZIENKIEWICZ, 2000; OÑATE, 2009 E RAO, 2011) para localizar elementos que serão indicados como materiais sólidos ou vazios para minimizar a função objetiva. Geralmente, em OT, a flexibilidade é usada como a função objetivo e como um limite para a quantidade de material que pode ser usada (DEATON; GRANDHI, 2014).

Nesse procedimento, a presença ou ausência de um material é analisada por meio de um modelo de material fictício, onde a região tem um valor de densidade que varia entre os limites de 0 e 1, e o material vazio e o material sólido são conformados. Portanto, quando um procedimento chamado SIMP é usada, o desempenho constitutivo de materiais intermediários (artificial) é caracterizado por uma função parâmetro  $\rho$  relacionada à densidade do material. Os materiais sólidos são representados por uma densidade igual a 1 e os vazios são retratados por uma densidade 0. A densidade relacionada a cada ponto no domínio é usada para definir o tensor constitutivo do material, de acordo com Bendsoe (1995) e Bendsoe e Sigmund (1999).

O procedimento SIMP é fundamentado com base na heurística entre densidade do elemento e módulo de elasticidade, de acordo com a equação (4) abaixo. O fator de penalidade é usado para minimizar a densidade intermediária, que é a região onde é difícil explicar se há material no processo final. Esta região da mesma forma também é chamada de escala de cinza, pois a densidade tem um valor intermediário. Sabe-se que uma densidade com valor 0 significa vazia e está marcada com branco, enquanto uma densidade com valor 1 significa sólida, que é simbolizada por preto. Portanto, a densidade está entre 0 e 1 representando o valor da mediana, e não está incluído em nenhuma das opções acima, o que dificulta a compreensão da resolução.

Em seguida, está a relação entre a densidade da unidade e o módulo de elasticidade.

$$E_e = x_e^p(E_0), \quad x_e \in [0,1] \quad (4)$$

Em Bendsøe e Sigmund (1999), a densidade relativa é estabelecida como uma variável contínua  $\rho$ , que assume um valor entre 0 e 1. Empregue um pequeno limite inferior,  $0 < \rho_{\min} \leq \rho$ , para impedir singularidades ao utilizar MEF. A semelhança entre esta densidade e o tensor material  $C_{ijkl}$  pode ser escrita como

$$C_{ijkl}(\rho) = \rho^p C_{ijkl}^0 \quad (5)$$

Levando em conta que o tensor  $C_{ijkl}$  é arbitrário e isotrópico, para qualquer densidade em relação a  $\rho$ , o material é isotrópico, ou seja,  $C_{ijkl}^0$  tem exclusivamente duas propriedades

modificáveis: Módulo de Young e coeficiente de Poisson. Além disso, a equação (3.11) satisfaz as relações

$$C_{ijkl}(0) = 0 \text{ e } C_{ijkl}(1) = C_{ijkl}^0 \quad (6)$$

Isso quer dizer que o projeto final contém uma densidade relativa punitiva, isso significa, existe um esforço de orientar a densidade a zero ou um em todos os locais. Por esta, admitindo-se um valor  $p$  grande o suficiente para ter desenho 0-1; em geral,  $p \geq 3$  é imprescindível (BENDSØE; SIGMUND, 1999).

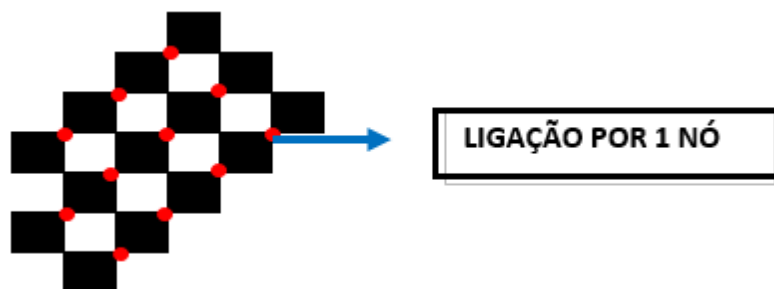
## 2.4 Método BESO

O procedimento de otimização de estrutura evolucionária BESO é fundamentado em um método heurístico no qual uma determinada concentração de material é selecionada para ser retirada ou acrescentada aos poucos (HUANG E XIE, 2010). Esse acréscimo ou retirada acontece devido à sensibilidade do elemento. Portanto, a quantidade de adição (AR) é um limite, que juntamente com a razão de adição máxima, ARmax, controla o número máximo de elementos que precisam ser acrescentados à estrutura. Portanto, quando  $AR > AR_{max}$ , apenas alguns elementos com números de sensibilidade mais altos são adicionados para definir  $AR = AR_{max}$ . Assim, o elemento com o menor número de sensibilidade é extraído para atender ao volume. Ainda é possível adicionar elementos considerando sua vizinhança, analisando que o valor de sensibilidade do elemento vazio que está próximo ao elemento de alta demanda não é zero, e a estrutura sob a condição de elemento entidade é retornada. (PICELLI, VICENTE e PAVANELLO, 2014).

O procedimento BESO foi desenvolvido por meio do método proposto por Xie e Steven (1996) e é denominado (ESO), que é usado para otimização de topologia estrutural. O desenvolvimento inicial do método ESO foi restrito a somente admitir que o material fosse removido da estrutura, e os elementos extraídos não puderam ser recuperados na evolução porvindoura, é fundamentado na concepção de que ao extrair vagarosamente materiais não muito eficaz da estrutura, a forma residual se desenvolverá em direção a uma estrutura melhor. A primeira exposição do ESO Bidirecional (BESO) foi proposta por Querin, Steven e Xie (1998) e Young *et al.* (1999). Desde esse momento, o método ESO tem sido continuamente aprimorado e desenvolvido em muitas particularidades, e tem muitas aplicações no campo da arquitetura e engenharia.

Sanches (2011), da utilização de grades hexagonais na otimização de estruturas evolutivas (método BESO), o principal incentivo é extinguir inconstâncias numéricas denominadas tabuleiros de xadrez, que são constituídos por regiões de materiais com densidade 1 alternando com áreas sem material onde sua densidade é 0. O benefício de utilizar elementos hexagonais é impedir a conexão de apenas um nó entre os elementos, conforme mostrado na Figura 8, a conexão forma o padrão de tabuleiro de xadrez.

**Figura 8** – Tabuleiro de Xadrez



Fonte: Pereira (2018)

Os artifícios circundados no método BESO para remover e acrescentar materiais seguem a técnica de eliminar o elemento sólido com o menor número de sensibilidade da estrutura e converter o elemento vazio com o maior número de sensibilidade em um sólido (Huang e Xie, 2010). O esquema exposto acima ocorre por meio de dois parâmetros não relacionados: a taxa de rejeição (RR) e a taxa de inclusão (IR).

Pode haver muitos padrões seguidos como princípio de retirada de materiais na estrutura, Young et al. (1999) e Liang, Xie e Steven (2000), tais como: padrões de tensão e deslocamento. Para o critério de estresse, use a expressão dada pela seguinte desigualdade,

$$\frac{\sigma_e^{vm}}{\sigma_{MAX}^{vm}} < RR_I \quad (7)$$

Onde,  $\sigma_e^{vm}$  é a tensão de von Mises do elemento,  $\sigma_{MAX}^{vm}$  é a máxima tensão de von Mises na estrutura e  $RR_I$  é a razão de rejeição.

Este procedimento pertence a um grupo de estruturas contínuas com aproximação macroestrutura, que se caracteriza por inserir furos no domínio discreto para realizar um processo de otimização da topologia.

Embora o método ESO seja amplamente utilizado, Huang e Xie (2010) apontaram que o material extraído nas primeiras iterações pode precisar fazer parte do projeto ideal

posteriormente. Todavia, o algoritmo ESO não pode reconstituir o material que pode ter sido extraído da estrutura de forma antecipada ou incorreta. Portanto, a solução otimizada final pode não ser absolutamente ideal. Devido às carências do método ESO, os autores Querin, Steven e Xie (1998) e Yang *et al.* (1999) desenvolveram um algoritmo melhor, denominado como método BESO. A principal desarmonia entre eles é que o primeiro só pode remover elementos que não atendam a um determinado padrão. Nesse método, tem a possibilidade de adicionar e remover elementos não muito eficientes.

O método ESO é bastante vulnerável à taxa de remoção dos elementos da malha, o que acarreta determinadas inconveniências. Um deles é o aparecimento de bordas desiguais e interconexões estruturais, resultando em uma concentração de mecanismo e tensão, Coutinho (2006).

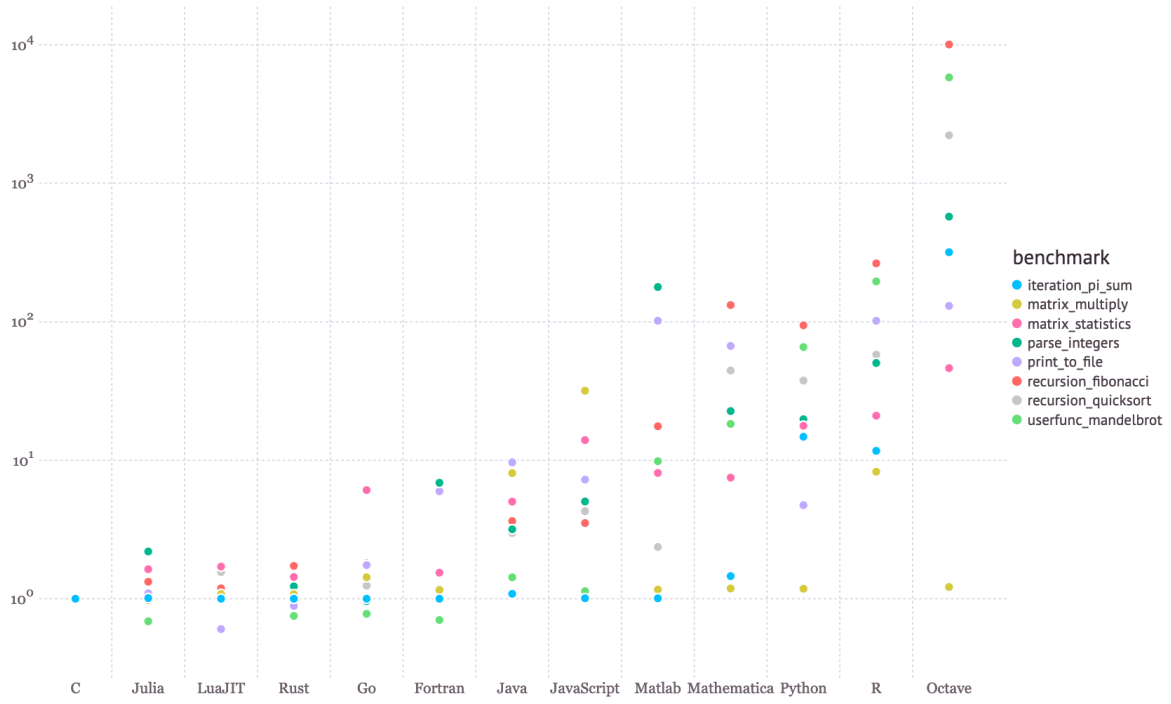
## **2.5 Linguagem Julia de programação**

A linguagem de programação Julia foi criada em 2012 por Alan Edelman, Stefan Karpinski, Jeff Bezanson e Viral Shah (BEZANSON *et al.*, 2012). Trata-se de uma linguagem gratuita e de código aberto, com bibliotecas próprias, além das bibliotecas C e Fortran. Sua performance é igual a de uma linguagem estaticamente compilada, como C e Fortran, com o comportamento interativo/dinâmico e produtividade de linguagens como Python e Ruby.

A Figura 9 mostra um teste de benchmark a fim de comparar a eficiência da linguagem Julia com a das principais linguagens de programação. A ordenada representa a ordem de grandeza do tempo gasto nos processamentos de teste.



**Figura 9** - Benchmark, linguagem Julia



Fonte: Julia (2021)

### 3 METODOLOGIA

O percurso metodológico iniciou com uma revisão da literatura com o objetivo de entender questões relacionadas e fundamentais na pesquisa de otimização de topologia (OT). Foi pesquisado o desenvolvimento de conceitos de otimização, métodos OT, métodos SIMP e BESO. A linguagem Julia tem o seu próprio REPL (do inglês *Read Eval Print Loop*), mas, devido a não possuir um “plotador” nativo, precisou-se de um ambiente de programação, aqui optado por utilizar o ATOM, que é um *software* também livre, gratuito, de código aberto e multiplataforma, que permite fazer a integração da linguagem Julia.

Conforme apresentado, são utilizados dois métodos de OT abundantemente utilizados na comunidade científica, nomeadamente os métodos BESO e SIMP. Na Seção 2.3 e 2.4, as questões relacionadas aos dois métodos são apresentadas com mais detalhes.

#### 3.1 Método SIMP

Sigmund (2001) elaborou um código OT fundamentado no método SIMP para minimizar a flexibilidade de carregar estaticamente uma estrutura bidimensional no ambiente MATLAB, denominado top99, usado como base para a escrita em linguagem Julia aqui utilizada com a integração do ATOM. O projeto auxiliou a disseminar o método e demonstrou a capacidade da OT porque ele pode se adaptar a distintos problemas e restrições.

Os dados fundamentais para iniciar o processo de otimização no programa são: o número de elementos, a fração de volume (a relação entre o volume final do domínio e o volume inicial), o coeficiente de penalidade e o tamanho do filtro.

Considerando que a topologia ótima da estrutura é obtida por meio de um processo iterativo, as variáveis de projeto precisam ser atualizadas. No código desenvolvido por Sigmund (2001), os critérios de otimização (OC) são utilizados para o otimizador, mas, de acordo mostrado na seção 2, possuem outras sugestões que podem ser usadas.

#### 3.2 Método BESO

Com base no código top99 (Sigmund, 2001), os pesquisadores Huang e Xie (2010) implementaram o método de otimização de topologia BESO Soft Kill no MATLAB. A densidade dos métodos SIMP e BESO Soft Kill são diferentes uns dos outros porque são variáveis contínuas e variáveis discretas. Desse modo, para o método BESO, apenas o critério de otimalidade (OC) é usado para atualizar as variáveis. O código elaborado por Huang e Xie (2010) serviu de base para o algoritmo BESO implementado nesse trabalho.

### 3.3 Casos estudados

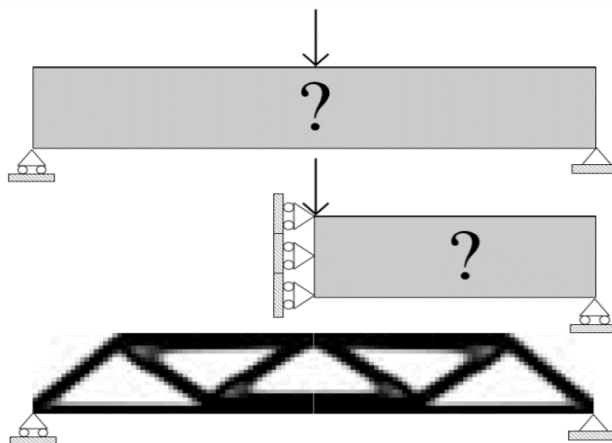
O rápido crescimento do poder e disponibilidade de computadores têm levado ao desenvolvimento de importantes ferramentas numéricas computacionais, que possibilitam a resolução de diversos problemas da engenharia. A linguagem Julia, possibilita a criação de codificações genéricas e de funções externas, sendo ambos extensíveis; possui capacidade de despachos múltiplos, o que permite combinações de argumentos bastante variadas; sua forma é otimizada e extensiva para tipos numéricos; tem tipagem dinâmica. O primeiro caso de todos os problemas foram os dados utilizados por Sigmund (2001), e a mudança nos parâmetros analisados, devido não haver parâmetros na literatura, adotou-se a mudança de 50%, ou próximo a isso, seja nos incrementos ou decrementos dos dados para observar e fazer as avaliações dos resultados.

Os problemas resolvidos no presente trabalho são detalhados a seguir.

#### 3.3.1 Problema 1

O primeiro problema trata de uma viga biapoiada com carga centrada aplicada no meio do vão, na parte superior da viga, conforme pode ser visto na Figura 13. Procura-se minimizar a flexibilidade da viga. Devido à simetria do problema, é feita uma simplificação utilizando-se apenas metade da viga com as condições de contorno mostradas também na Figura 10.

**Figura 10** – Viga MBB



**Fonte:** Sigmund (2001)

Tal problema é conhecido como viga MBB (Messerschmitt-Bölkow-Blohm), mais detalhes sobre tal podem ser vistos em Sigmund (2001).

Para este problema, utiliza-se o programa top que faz uso do método SIMP. Em sua execução de tal programa, é necessário definir alguns parâmetros de entrada, são eles: a

distribuição da quantidade de elementos na direção x (nelx) e direção y (nely), a fração de volume (volfrac), o parâmetro de penalização (penal), e o tamanho do filtro (rmin). Para o método BESO, ao executar tal programa, alguns parâmetros de entrada precisam ser definidos, são eles: a distribuição do número de elementos na direção x (nelx) na direção y (nely), a fração de volume (volfrac), a razão de evolução (er) e tamanho de filtragem (rmin).

A fim de avaliar a influência de cada parâmetro de entrada sobre a densidade do material de composição da viga MBB, testa-se o uso de diferentes valores para esses parâmetros. As combinações de parâmetros adotadas no presente problema são mostradas na Tabela 1.

**Tabela 1** – Casos do Problema 1, pelo método SIMP

Messerschmitt-Bolkow-Blohm (MBB-beam)					
Caso	nelx	nely	volfrac	penal	rmin
S-MBB-1	60	20	0,5	3	1,5
S-MBB-2	30	10	0,5	3	1,5
S-MBB-3	90	30	0,5	3	1,5
S-MBB-4	60	20	0,4	3	1,5
S-MBB-5	60	20	0,6	3	1,5
S-MBB-6	60	20	0,5	3	1
S-MBB-7	60	20	0,5	3	2

**Fonte:** autor (2021)

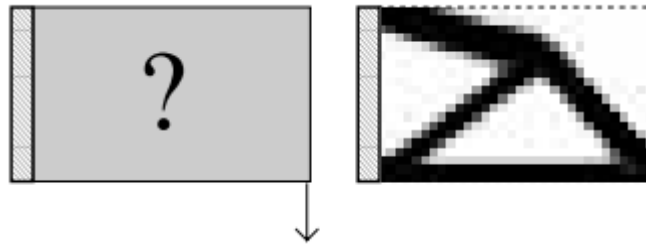
**Tabela 2** – Casos do Problema 1, pelo método BESO

Short Cantilever (SC-beam)					
Caso	nelx	nely	volfrac	penal	rmin
S-SC-1	32	20	0,4	3	1,2
S-SC-2	48	30	0,4	3	1,2
S-SC-3	64	40	0,4	3	1,2
S-SC-4	32	20	0,3	3	1,2
S-SC-5	32	20	0,5	3	1,2
S-SC-6	32	20	0,4	3	2

**Fonte:** autor (2021)

### 3.3.2 Problema 2

Neste problema procura-se otimizar, através de um dos métodos anteriores, uma viga curta engastada com carga distribuída na parte inferior da extremidade livre, conforme a Figura 11.

**Figura 11** – Viga SC

Fonte: Sigmund (2001)

Tal problema é conhecido como viga SC, mais detalhes sobre tal podem ser vistos em Sigmund (2001).

A malha computacional para este problema ( $n_{elx}$  e  $n_{ely}$ ) é fixa. São adotados 32 x 20 elementos em cada dimensão da viga, conforme é mostrado na Figura 10.

**Tabela 3** – Casos do Problema 2, pelo método SIMP

Messerschmitt-Bolkow-Blohm (MBB-beam)					
Caso	$n_{elx}$	$n_{ely}$	volfrac	er	rmin
B-MBB-1	60	20	0,5	0,05	1,5
B-MBB-2	30	10	0,5	0,05	1,5
B-MBB-3	90	30	0,5	0,05	1,5
B-MBB-4	60	20	0,4	0,05	1,5
B-MBB-5	60	20	0,6	0,05	1,5
B-MBB-6	60	20	0,5	0,05	1
B-MBB-7	60	20	0,5	0,05	2

Fonte: autor (2021)

**Tabela 4** – Casos do Problema 2, pelo método BESO

Short Cantilever (SC-beam)					
Caso	$n_{elx}$	$n_{ely}$	volfrac	er	rmin
B-SC-1	32	20	0,4	0,01	1,2
B-SC-2	48	30	0,4	0,01	1,2
B-SC-3	64	40	0,4	0,01	1,2
B-SC-4	32	20	0,3	0,01	1,2
B-SC-5	32	20	0,5	0,01	1,2
B-SC-6	32	20	0,4	0,01	2

Fonte: autor (2021)

Os parâmetros de entrada serão adotados de acordo com análises feitas nos problemas anteriores, a fim de proporcionar o melhor resultado.

## 4 RESULTADOS E DISCUSSÃO

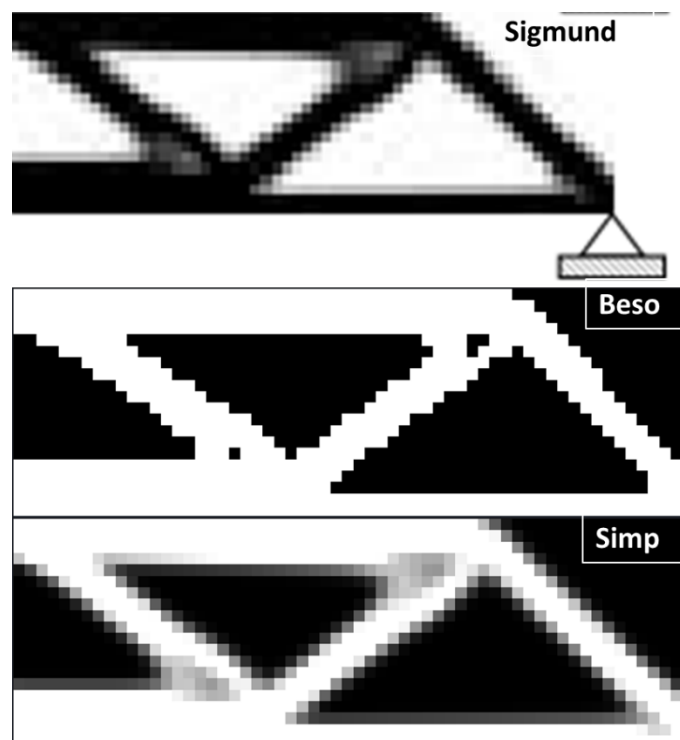
### 4.1 Validação dos códigos

Na sequência serão apresentados os resultados da implementação computacional para a resolução dos problemas 1 e 2. A metodologia utilizada tem como base os trabalhos de SIGMUND (2001) e os algoritmos foram implementados na linguagem Julia.

#### 4.1.1 Viga MBB

Na Figura 12 abaixo, mostram-se os resultados obtidos considerando os métodos de OT, SIMP e BESO, para o primeiro caso. Pode-se verificar que, visualmente, existe grande semelhança entre o resultado publicado por Sigmund (2001) e aquele encontrado pelo código desenvolvido neste trabalho, quando as mesmas condições de contorno são aplicadas, utilizando os mesmos parâmetros definidos para a viga, o algoritmo gerou uma topologia com distribuição de densidades bem próxima daquela encontrada por Sigmund (2001).

**Figura 12** – Resultado da OT da solução encontrada por Sigmund (2001), e da topologia resultante com o uso da ferramenta criada dos métodos SIMP e BESO do Problema 1.

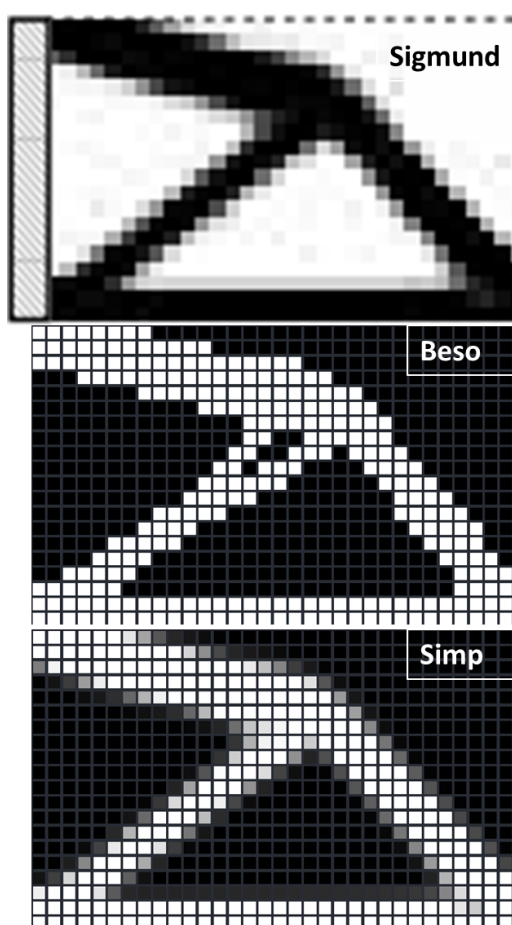


Fonte: autor (2021)

#### 4.1.2 Viga SC

A Figura 13 a seguir exhibe as implicações alcançadas considerando os métodos OT, SIMP e BESO no segundo caso. É possível perceber na imagem as decorrências publicadas por Sigmund (2001) e os resultados descobertos no código desenvolvido neste trabalho também proporcionam grandes analogias, no momento que as condições análogas de contorno são justapostas, empregados os parâmetros análogos determinados para a viga. A distribuição da densidade topológica gerada por este algoritmo é muito similar à descoberta por Sigmund (2001).

**Figura 13** – Resultado da OT da solução encontrada por Sigmund (2001), e da topologia resultante com o uso da ferramenta criada dos métodos SIMP e BESO do Problema 2



Fonte: autor (2021)

## 4.2 Comparativos

### 4.2.1 Casos do problema 1, método SIMP

Os valores dos resultados do presente problema 1 para o método SIMP da viga MBB são mostrados na Tabela 5. Em cada caso, a partir do segundo, apresenta a variação de algum dos parâmetros.

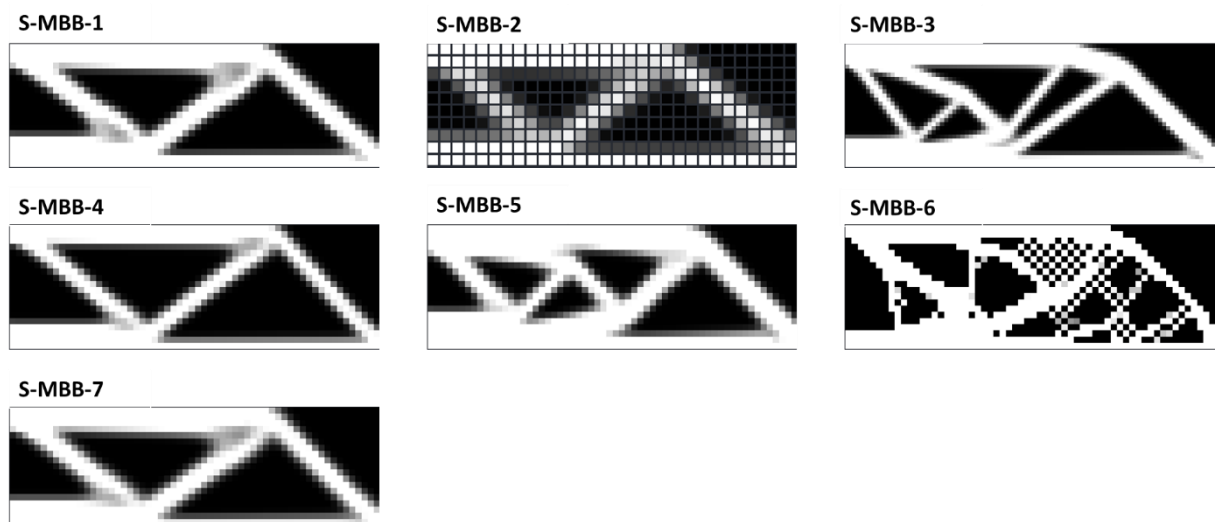
**Tabela 5** – Resultados do Problema 1, pelo método SIMP

Messerschmitt-Bolkow-Blohm (MBB-beam)							
Caso	nelx	nely	volfrac	penal	rmin	iterações	obj
S-MBB-1	60	20	0,5	3	1,5	94	203,3
S-MBB-2	30	10	0,5	3	1,5	72	225,6
S-MBB-3	90	30	0,5	3	1,5	379	196,6
S-MBB-4	60	20	0,4	3	1,5	63	255,35
S-MBB-5	60	20	0,6	3	1,5	79	171,6
S-MBB-6	60	20	0,5	3	1	65	203,2
S-MBB-7	60	20	0,5	3	2	54	209,3

Fonte: autor (2021)

Percebe-se, analisando a tabela acima, primeiramente, que o incremento de malha computacional reduz a função objetivo (flexibilidade), porém, aumenta o custo computacional, já que além de cada iteração custar mais tempo, aumentou-se também o número de iterações. A variação da fração de volume final também teve grande influência na função objetivo, o que já era esperado, devido quando se tem menos materiais construtivos na viga vai se ter uma flexibilidade maior, e vai diminuir a rigidez. Por fim a variação do tamanho do filtro teve efeito significativo na redução do número de iterações, tanto com o aumento tanto com a redução deste parâmetro, observamos a diminuição do número de iterações, porém a função objetivo não teve grandes mudanças.

A Figura 14 mostra os resultados do problema 1 para o método SIMP, conforme os casos mostrados na Tabela 5.

**Figura 14** – Resultados do problema 1 para o método SIMP da viga MBB

Fonte: autor (2021)



Na figura acima pôde-se observar que com a diminuição do refinamento, da fração de volume e com o aumento do filtro, não houve mudanças consideráveis no formato da viga. Contudo, o aumento do refinamento da malha e da fração de volume aumentaram o número de barras que compõem a viga, e a redução do filtro resultou em um formato não construtivo, apresentando problema do “tabuleiro de xadrez”.

#### 4.2.2 Casos do problema 1, método BESO

Os valores dos resultados deste problema 2 para o método BESO da viga MBB são apresentados na Tabela 6. Em cada condição, a partir do segundo, exibiu determinadas mudanças de parâmetros.

**Tabela 6** – Resultados do Problema 1, pelo método BESO

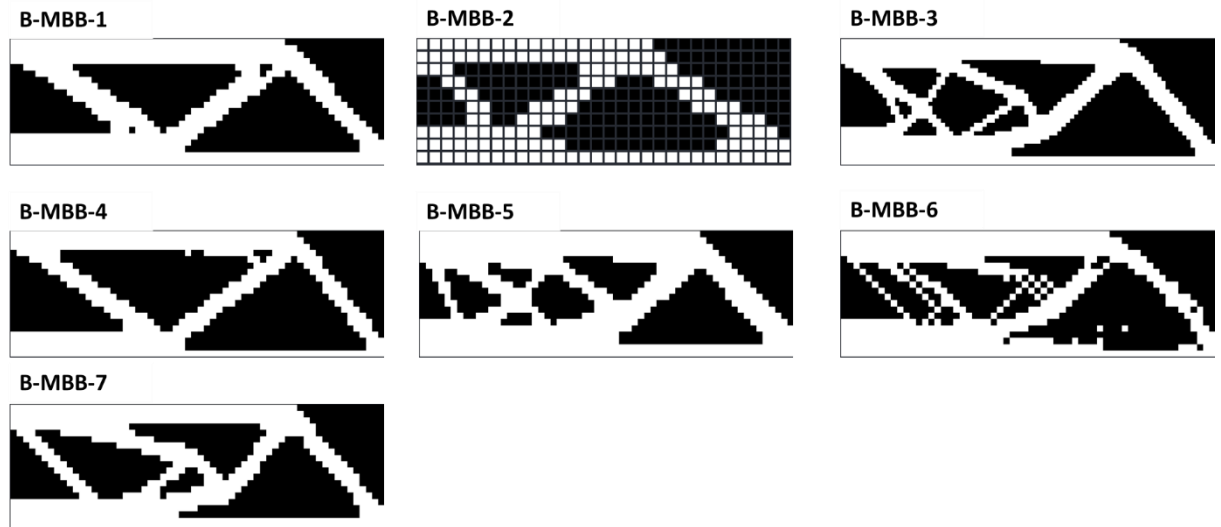
Messerschmitt-Bolkow-Blohm (MBB-beam)							
Caso	nelx	nely	volfrac	er	rmin	iterações	obj
B-MBB-1	60	20	0,5	0,05	1,5	35	95,3
B-MBB-2	30	10	0,5	0,05	1,5	32	100,9
B-MBB-3	90	30	0,5	0,05	1,5	23	94,5
B-MBB-4	60	20	0,4	0,05	1,5	28	116,6
B-MBB-5	60	20	0,6	0,05	1,5	35	81,2
B-MBB-6	60	20	0,5	0,05	1	23	109,6
B-MBB-7	60	20	0,5	0,05	2	38	95,5

**Fonte:** autor (2021)

Observa-se, por meio da tabela 6, inicialmente, que o acréscimo da malha computacional de cálculo se manteve relativamente constante a função objetivo (flexibilidade) sem alteração considerável. Entretanto, houve variações no número de iterações em função da mudança de malha. A mudança da fração de volume final conjuntamente tem importante contribuição na função objetivo, o que é previsto, devido quando há menos materiais de construção na viga, a flexibilidade será maior e a rigidez diminuirá.

Por último, a mudança do tamanho do filtro tem uma implicação expressiva do número de iterações, seja com o acréscimo ou redução deste parâmetro. Assim, pode-se notar que o número de iterações decresce ou aumenta, mas a função objetivo não altera expressivamente.

A Figura 15 apresenta os resultados do problema 1 para o método BESO, de acordo indicado na Tabela 6.

**Figura 15** – Resultados do problema 1 para o método BESO da viga MBB

Fonte: autor (2021)

Foi possível constatar na figura acima que conforme o grau de refinamento reduz, conforme a fração de volume reduz, o formato da viga não altera significativamente. Porém, o refinamento da malha e o aumento da fração de volume aumentou o número de vigas que a compõem, e a redução dos filtros leva a formas não edificante.

#### 4.2.3 Casos do problema 2, método SIMP

Os valores das implicações do método SIMP para a viga SC neste problema 2 é mostrado na Tabela 7. Em cada condição, a partir da segunda, mostrou-se uma determinada alteração de parâmetro.

**Tabela 7** – Resultados do Problema 2, pelo método SIMP

Short Cantilever (SC-beam)							
Caso	nelx	nely	volfrac	penal	rmin	iterações	obj
S-SC-1	32	20	0,4	3	1,2	71	57,3
S-SC-2	48	30	0,4	3	1,2	134	57,2
S-SC-3	64	40	0,4	3	1,2	110	56,1
S-SC-4	32	20	0,3	3	1,2	42	81,1
S-SC-5	32	20	0,5	3	1,2	69	46,3
S-SC-6	32	20	0,4	3	2	45	63,1

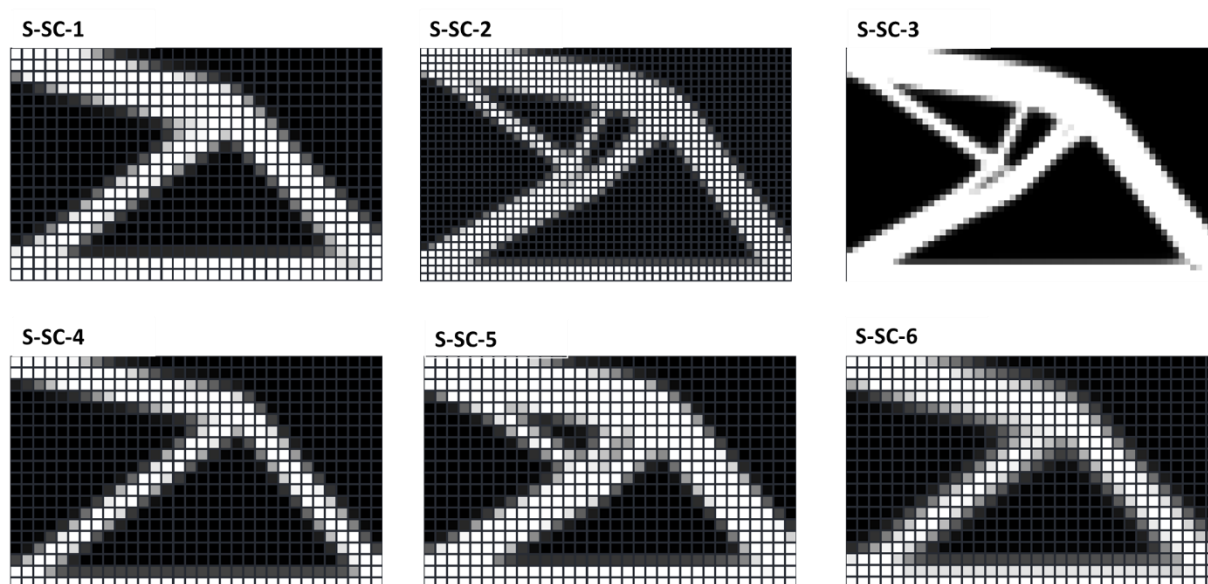
Fonte: autor (2021)

Analisando a Tabela 7, pode-se observar que a adição da malha de cálculo inicialmente calculada mantém uma função objetivo (flexibilidade) relativamente constante sem alterações

significativas, porém, devido às alterações da malha, o número de iterações apresentou alterações relevantes. A mudança da fração de volume final tem uma importante contribuição no número de iterações e na função objetivo. Por fim, alterar o tamanho do filtro tem um significado considerável para reduzir o número de iterações. Pode-se notar que o número de iterações diminui, mas a função objetivo não muda significativamente.

A Figura 16 mostra os resultados do problema 2 do método SIMP, conforme mostrado na Tabela 7.

**Figura 16** – Resultados do problema 2 para o método SIMP da viga SC



**Fonte:** autor (2021)

Como pode ser visto na figura acima, quando o grau de refinamento diminui e o filtro aumenta, não houve alterações na viga, diferentemente quando a fração de volume aumenta houve uma pequena mudança no formato. Assim como, o refinamento da malha e o aumento da fração de volume aumentam o número de vigas que a estruturam.

#### 4.2.4 Casos do problema 2, método BESO

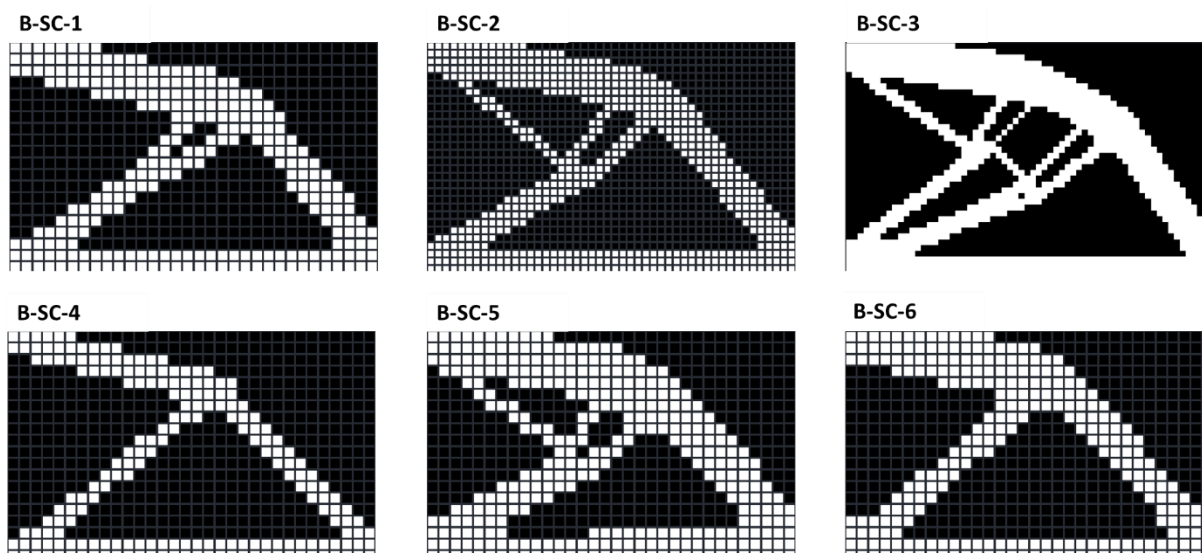
Os valores dos resultados do método BESO da viga SC nesta questão 2 é apontado na Tabela 8. Em cada categoria, a partir da segunda, é exibida uma certa modificação de parâmetro.

**Tabela 8** – Resultados do Problema 2, pelo método BESO

Short Cantilever (SC-beam)							
Caso	nelx	nely	volfrac	er	rmin	iterações	obj
B-SC-1	32	20	0,4	0,01	1,2	102	26,67
B-SC-2	48	30	0,4	0,01	1,2	100	26,6
B-SC-3	64	40	0,4	0,01	1,2	101	26,6
B-SC-4	32	20	0,3	0,01	1,2	132	35,7
B-SC-5	32	20	0,5	0,01	1,2	79	21,8
B-SC-6	32	20	0,4	0,01	2	103	26,9

Fonte: autor (2021)

Através da Tabela 8, é possível verificar que a malha de cálculo adicionada ao cálculo inicial conserva uma função objetivo (flexibilidade) e o número de iterações constante sem nenhuma mudança considerável. Entretanto aumentando a fração de volume final tem uma imposição relevante para a redução do número de iterações e da função objetivo, mas diminuindo a fração de volume ocorreu um aumento do número de iterações e das funções objetivo. Por último, com a modificação do tamanho do filtro não houve alterações, pode-se notar que o número de iterações, e a função objetivo não mudou relevantemente.

**Figura 17** – Resultados do problema 2 para o método BESO da viga SC

Fonte: autor (2021)

De acordo com a figura 17, quando o grau de refinamento é reduzido e o filtro é aumentado, a viga não muda, ao contrário de quando a fração de volume sofre um acréscimo. Da mesma forma, o refinamento da malha e o aumento da fração de volume aumentam o número de viga que a compõem.

## 5 CONCLUSÃO

O principal objetivo do presente trabalho é otimizar através do método de otimização topológica (OT): SIMP (Solid Isotropic Material with Penalization) e BESO (Bi-Directional Evolutionary Structural Optimization). Para a pesquisa, foi desenvolvido o código implementado em ambiente JULIA, e feito o comparativo entre os resultados do SIMP e BESO nos problemas 1 e 2 com cada método, permitindo observar melhor o que foi comentado.

Pôde-se observar que a escolha dos parâmetros certos é fundamental para a obtenção de bons resultados na otimização topológica. Dependendo do método, há parâmetros com maior ou menor influência sobre os resultados. Por esse motivo, sempre é válido realizar um estudo comparativo antes da escolha final dos parâmetros.

Para os casos testados, o método BESO mostrou ser mais rápido que o método SIMP, além de apresentar topologias bem definidas. Além disso, a presença de material intermediário com interpretação dúbia no método SIMP (regiões cinzentas) pode atrapalhar bastante no momento de execução de projetos.

Pelo que pôde ser visto, a principal diferença entre os resultados dos métodos utilizados nos problemas anteriores é a existência de material intermediário. Percebe-se que no método BESO os resultados de topologia são bem definidos, enquanto método SIMP os resultados possuem regiões com material dúbio, onde não se sabe se realmente há material.

Percebe-se ainda que o método BESO tendeu a locar menos furos que o método SIMP, contudo, foi capaz de obter excelentes valores de flexibilidade. Vale lembrar que topologias mais simples são construtivamente mais atrativas. Há um efeito de “serrilhamento” muito mais perceptível nos resultados obtidos com o método BESO, contudo, o aumento no refinamento da malha computacional suaviza esse efeito e o torna menos expressivo.

Por fim, conclui-se que de maneira geral, a partir da escolha dos parâmetros certos, tanto o método SIMP, quanto o método BESO, são capazes de apresentar resultados satisfatórios.

## 6 REFERÊNCIAS

ACHTZIGER, W. **Topology Optimization of Discrete Structures**. Topology Optimization in Structural Mechanics, v. 347, p. 57-100, 1997.

AMSTUTZ, S.; NOVOTNY, A. A. Topological optimization of structures subject to Von Mises stress constraints. **Structural and Multidisciplinary Optimization**, v. 41, p. 407 – 420, 2010.

AMSTUTZ, S.; NOVOTNY, A. A.; SOUZA NETO, E. A. Topological derivative-based topology optimization of structures subject to Drucker–Prager stress constraints. **Computer Methods in Applied Mechanics and Engineering**, v. 233–236, p. 123– 136, 2012.

ANDREASSEN, E; SIGMUND, O; CLAUSEN, A; SCHEVENELS, M; LAZAROV, B. S. **Efficient topology optimization in MATLAB using 88 lines of code**. Struct. Multidisc. Optim., v. 43, p. 1-16, 2011.

ARORA, J.S., “**Introduction to Optimum Design**”, Elsevier Inc., USA, (2004)

BECKERS, M. Topology Optimization Using a Dual Method with Discrete Variables.

BENDSØE, M. P.; SIGMUND, O. Material interpolation schemes in topology optimization. **Archive of Applied Mechanics**, v. 69, p. 635 – 654, 1999.

BENDSOE, M. P.; SIGMUND, O. **Topology optimization: Theory, Methods and Applications**. Berlin; Heidelberg; New York; Barcelona; Hong Kong; London; Milan; Paris; Tokyo: Springer, 2003.

BENDSØE, M.P. E KIKUCHI, N., “**Generating optimal topologies in structural design using a homogenization method**”, Computer Methods in Applied Mechanics and Engineering, Vol. 71, pp. 197-224, (1988).

BENDSØE, M.P. E SIGMUND, O., **Topology Optimization**, Springer, Berlin, (2002).

BENDSOE, M.P. **Optimal shape design as a material distribution problem**. Structural Optimization, v. 1, p. 193-202, 1989.

BENDSOE, M.P. **Optimization of structural topology, shape, and material**. 1. ed. Berlin, Heidelberg, New York; Springer, 1995.

BENDSOE, M.P., KIKUCHI, N., **Generating optimal topologies in structural design using a homogenization method**. *Comput. Methods Appl. Mech. Eng*, v. 71, p. 197–224, 1988.

BENDSOE, M.P; SIGMUND, O. **Material interpolation schemes in topology optimization**. Archive of Applied Mechanics, v. 69, p. 635-654, 1999.

BENDSØE, MARTIN P.; KIKUCHI, Noboru. Generating optimal topologies in structural design using a homogenization method. **Computer Methods in Applied Mechanics and Engineering**, [s. l.], v. 71, p. 197–224, 1988.

BERKE L., KHOT N.S. (1987) **Structural Optimization Using Optimality Criteria**. In: Mota Soares C.A. (eds) *Computer Aided Optimal Design: Structural and Mechanical Systems*. NATO ASI Series (Series F: Computer and Systems Sciences), vol 27. Springer, Berlin, Heidelberg. [https://doi.org/10.1007/978-3-642-83051-8\\_7](https://doi.org/10.1007/978-3-642-83051-8_7)

BEZANSON, J.; KARPINSKI, S.; SHAH, V. B.; EDELMAN, A. Julia: A Fast Dynamic Language for Technical Computing. *arXiv*, v. 1, 2012.

BRUGGI, M.; DUYSINX, P. Topology optimization for minimum weight with compliance and stress constraints. *Structural and Multidisciplinary Optimization*, v. 46, p. 369–384, 2012.

BRUGGI, M.; TALIERCIO, A. Topology optimization of the fiber-reinforcement retrofitting existing structures. *International Journal of Solids and Structures*, v. 50, n. 12, p. 121–136, 2013.

CANFIELD, S. E FRECKER, M., **Topology Optimization of Compliant Mechanical Amplifiers for Piezoelectric Actuators**, *Struct Multidisc Optim*, p.269-279, (2000).

CARVALHO, G. C. G. **Otimização de seções caixão de concreto armado e protendido pelo método dos algoritmos genéticos**. 2014. 126p. Dissertação (Mestrado em Engenharia Civil) – Universidade Federal do Rio de Janeiro – Rio de Janeiro. 2014.

CAVAZZUTI, M; BALDINI, A; BERTOCCHI, E; COSTI, D; TORRICELLI, E; MORUZZI, P. **High performance automotive chassis design: a topology optimization based approach**. *Struct. Multidisc. Optim.*, v. 44, p. 45-56, 2011.

CHIESA, D. **Otimização Topológica Aplicada em um Braço Robótico Flexível**. 2016. 115 f. Dissertação (mestrado) - Programa de Pós-Graduação em Modelagem Matemática da Universidade Federal de Pelotas, Brasil, 2016.

CHO, S. E CHOI, J., **Efficient Topology Optimization of Thermo-Elasticity Problems Using Coupled Field Adjoint Sensitivity Analysis Method**, *Finite Element in Analysis and Design*, (2005).

CHRISTENSEN, PETER; KLARBRING, ANDERS. **An Introduction to Structural Optimization**. Dordrecht: Springer Netherlands, 2009. ISSN 0925-0042. (Solid Mechanics and Its Applications).v. 153 Disponível em: <https://doi.org/10.1007/978-1-4020-8666-3>

CHRISTIE, W. C., BETTESS, P. E BULL, J. W., **Self-Designing Structures: A Practical Approach**, *Engineering Computations*, v. 15, n. 1, p. 35-48, (1998).

CHU, D.N., XIE, Y.M., HIRA, A. AND STEVEN, G.P., **“Evolutionary structural optimization for problems with stiffness constraints”**, *Finite Elements in Analysis and Design*, Vol. 21, pp. 239-251, (1996).

CLAUSEN, A.; AAGE, N.; SIGMUND, O. Topology optimization with flexible void area. *Structural and Multidisciplinary Optimization*, v. 50, p. 927 – 943, 2014.

COUTINHO, K. D., **Método de Otimização Topológica em Estruturas Tridimensionais**, Dissertação de Mestrado, Universidade Federal do Rio Grande do Norte, Programa de Pós Graduação em Engenharia Mecânica, Natal, RN, (2006).

DEATON, J. D; GRANDHI, R. V. **A survey of structural and multidisciplinary continuum topology optimization: post 2000**. Struct. Multidisc. Optim., v. 49, p. 1-38, 2014.

DUAN, X. B.; LI, F. F.; QIN, X. Q. Adaptive mesh method for topology optimization of fluid flow. **Applied Mathematics Letters**, v. 44, p. 40 – 44, 2015.

DUYSINX, P.; BENDSØE, M. P. Topology optimization of continuum structures with local stress constraints. **International Journal for Numerical Methods in Engineering**, v. 43, p. 1453 – 1478, 1998.

EMMENDOERFER, H. JR.; FANCELLO, E. A. A level set approach for topology optimization with local stress constraints. **International Journal for Numerical Methods in Engineering**, v. 99, p. 129 – 156, 2014.

ESCHENAUER, H. A.; OLHOFF, N. Topology optimization of continuum structures: A review\*. **Applied Mechanics Reviews**, v. 54, p. 331 – 390, 2001.

ESCHENAUER, H. A; KOBELLEV, V.V; SCHUMACHER, A. **Bubble method for topology and shape optimization of structures**. Structural Optimization, v. 8, p. 42-51, 1994.

ESCHENAUER, H. A; OLLOFF, N. **Topology optimization of continuum structures: A review**. Appl. Mech., v. 54, 2001.

FANCELLO, E. A.; PEREIRA, J. T. Structural topology optimization considering material failure constraints and multiple load conditions. **Latin American Journal of Solids and Structures**, v. 1, p. 3 - 24, 2003.

GAO, H. H. et al. An improved adaptive constraint aggregation for integrated layout and topology optimization. **Computer Methods in Applied Mechanics and Engineering**, v. 289, p. 387 – 408, 2015.

GUEST, J. K.; SMITH GENUT, L. C. Reducing dimensionality in topology optimization using adaptive design variable fields. **International Journal for Numerical Methods in Engineering**, V. 81(8), p.1019 - 1045, 2010.

GUO, X.; ZHANG, W.; ZHONG, W. Explicit feature control in structural topology optimization via level set method. **Computer Methods in Applied Mechanics and Engineering**, v. 272, p. 354 – 378, 2014.

HAFTKA, R. GÜRDAL, Z. **Elements of structural optimization**. 3 ed. Holland: Kluwer Academic Publishers, 1991. 481p.

HAFTKA, R.T. E GANDHI, R.V., “**Structural Shape Optimization – A survey**”, Computer Methods in Applied Mechanics and Engineering, v. 57, pp. 91-106, (1986).



HINTON, E. AND SIENZ, J., “**Fully stressed topological design of structures using an Evolutionary Procedure**”. *Engineering Computations*, v. 12, p. 229-44, (1995).

HOLMBERG, E.; TORSTENFELT, B.; KLARBRING, A. Stress constrained topology optimization. **Structural and Multidisciplinary Optimization**, v 48, p. 33 – 47, 2013.

HUANG, X., XIE, Y.M., **Convergent and mesh-independent solutions for the bi-directional evolutionary structural optimization method**. *Finite Elements in Analysis and Design*, 43:1039–1049, 2007.

HUANG, X.; XIE, Y.M. **Evolutionary topology optimization of continuum structures**. 1. ed. New Delhi, India: WILEY, 2010.

Julia. **Julia Micro-Benchmarks**. Disponível em: <<https://julialang.org/benchmarks/>>. Acesso em 04 de março de 2021.

KERDAL, D. AND NETHERCOTT, D. A. **Failure Modes for Castellated Beams**. **J. Construct.** *Steel Research* 4 (1984): 295-315.

KIRSCH, U., “**On Singular Topologies in Optimum Structural Design**”, *Structural Optimization*, vol. 2, pp.133-142, (1990).

KUTUK, M. A; GOV, I; **Application of topology optimization to the tibial osteotomy fixation plates**. *Applied Bionics and Biomechanics*, v. 10, 125-133, 2013.

LIANG, Q.Q; XIE, Y.M; STEVEN, G.P. **Optimal topology selection of continuum structures with displacement constraints**. *Computers and Structures*, v. 77, 635-644, 2000.

LIU, KAI; TOVAR, ANDRÉS. An efficient 3D topology optimization code written in Matlab. **Structural and Multidisciplinary Optimization**, [s. l.], v. 50, n. 6, p. 1175–1196, 2014. Disponível em: <https://doi.org/10.1007/s00158-014-1107-x>

LUO, Y.; WANG, M. Y.; KANG, Z. An enhanced aggregation method for topology optimization with local stress constraints. **Computer Methods in Applied Mechanics and Engineering**, v. 254, n. 5, p. 31- 41, 2013.

METHA, P.K., MONTEIRO, P.J.M. **Concreto – Estrutura, Propriedades e Materiais**. São Paulo – SP, Editora PINI, 1994.

MICHELL, A.G.M. (1904) **The Limits of Economy of Material in Frame Structures**. *Philosophical Magazine*, Series 6, 8, 589-597. <http://dx.doi.org/10.1080/14786440409463229>

MIRZENDEHDEL, A.M, SURESH, K. **Support structure constrained topology optimization for additive manufacturing**. *Computer-Aided Design*, v. 81, p. 1-13, 2016.

NAHRY, TIAHJONO, T., SATITI, Y.J., **The Optimization Model of Runway and Gate Assignment**. *Civil Engineering Dimension*, 15:129-136, 2013.

OCHOA, M.A., VALDEZ, S.I., BOTELLO, S., CARDOSO, V., **Topology optimization benchmark I**. CIMAT, 2016.

OKTAY, E; AKAY, H.U; MERTTOPCUOGLU, O. **Parallelized structural topology optimization and CFD coupling for design of aircraft wing structures**. *Computers & Fluids*, v. 49, p. 141-145, 2011.

OLIVEIRA NETO, J. A. **Otimização Topológica de Estruturas Termoelásticas**. 122 f. Dissertação Universidade Federal do Rio Grande do Norte. Programa de Pós-Graduação em Engenharia Mecânica. Natal, 2011.

OÑATE, E. **Structural analysis with the finite element method linear statics**. 1. Ed. Espanha, Springer, 2009.

PEREIRA, C. E. L; BITTENCOURT, M.L. **Topological sensitivity analysis for a two-parameter MooneyRivlin hyperelastic constitutive model**. *Latin American Journal of Solids and Structures*, v. 7, p. 391-411, 2010.,

PEREIRA, C.E.L. **Análise de Sensibilidade Topológica em Problemas de Não-Linearidade Geométrica e Hiperelasticidade Não-Linear Quase-Incompressível**, Tese de Doutorado, Universidade Estadual de Campinas, Campinas, São Paulo, (2006).

PEREIRA, J. T.; FANCELLO, E. A.; BARCELLOS, C. S. Topology optimization of continuum structures with material failure constraints. **Structural and Multidisciplinary Optimization**, v. 26, p.50 – 66, 2004.

PEREIRA, R. **Otimização topológica de sistema de contraventamento em edificações, considerando os efeitos do vento**. 2018. 131 f. Dissertação (mestrado em engenharia civil) - Universidade Federal de Pernambuco, Brasil, 2018.

PICELLI, R; VICENTE, W.N; PAVANELLO, R. **Bi-directional evolutionary structural optimization for design-dependent fluid pressure loading problems**. *Engineering Optimization*, v. 47, p. 1324-1343, 2014.

PORTO, E. D. E PAVANELLO, R., **Influencia dos Parâmetros de Homogeneização sobre a solução estrutural topológica ótima**, VIII Congresso Ibero Americano de Engenharia Mecânica, Peru, (2007).

PRAYOGO, D., CHENG, M.Y., PRAYOGO, H., **A Novel Implementation of Nature-inspired Optimization for Civil Engineering: A Comparative Study of Symbiotic Organisms Search**. *Civil Engineering Dimension*, 19:36-43, 2017.

QIAN, X. Topology optimization in B-spline space. **Computer Methods in Applied Mechanics and Engineering**, v. 265, n. 2, p. 15 – 35, 2013.

QUERIN, O. M., STEVEN, G.P. E XIE, Y.M., **Evolutionary Structural Optimization using Additive Algorithm, Finite Element Analysis and Design**, v. 34, p. 291-308, (2000a).

QUERIN, O. M., YOUNG, V., STEVEN, G.P. E XIE, Y.M., **Computational Efficiency and Validation of Bi-Directional Evolutionary Structural Optimization**, *Comput. Meth. Appl. Mech. Engng.*, v. 189, p. 559-73, (2000b).

QUERIN, O.M; STEVEN, G.P; XIE, Y.M, **Evolutionary structural optimisation (ESO) using a bidirectional algorithm**, Eng. Comput., v. 15, p. 1031-1048, 1998.

RAO, S. S. **The finite element method in engineering**. 5. Ed, Florida, Butterworth-Heinemann, 2011.

REYNOLDS D., MCCONNACHIE J., BETTESS P., CHRISTIE W. C, BULL J. W., **Reverse adaptivity – a new evolutionary tool for structural optimization**, Int. J. Numer. Meth. Engrg., v. 45(5), p. 529-52, (1999).

RONG, J.H., XIE, Y.M., YANG, X.Y. AND LIANG, Q.Q., **“Topology optimization of structures under dynamic response constraints”**, *Journal of Sound and Vibration*, Vol. 234, No. 2, pp.177-189, (2000).

ROZVANY, G. I. N. **A critical review of established methods of structural topology optimization**. Structural and Multidisciplinary Optimization, v. 37, p.217 – 237, 2009.

ROZVANY, G.I.N; ZHOU, M; BIRKER.T. **Generalized shape optimization without homogenization**. Structural Optimization, v. 4, p. 250-252, 1992.

SAITOU, K; IZUI, K; NISHIWAKI, S; PAPALAMBROS, P. **A survey of structural optimization in mechanical product development**. Journal Of Computing and Information Science in Engineering. v. 5, p. 214-226, 2005.

SALAGAME, R. R. E BELEGUNDU, A. D., **“Shape Optimization with p-Adaptivity”**, *American Institute of Aeronautics and Astronautics Journal*, vol. 33 (12), pp. 2399-2405, 1995.

SANCHES, R. P. **Otimização estrutural evolucionária usando malhas hexagonais**. 2011. Dissertação (Engenharia Mecânica) - Universidade Estadual de Campinas, 2011.

SANTANA, H.M., **Otimização topológica de estruturas bidimensionais contínuas submetidas a restrições de flexibilidade e tensão**. Dissertação (Mestrado). Porto Alegre: Universidade Federal do Rio Grande do Sul, (2002).

SCHMIT, L.A., **STRUCTURAL SYNTHESIS – its genesis and development**. AIAA J. 19, 1249–1263, (1981).

SIGMUND, O. **A 99 line topology optimization code written in matlab**. Struct. Multidisc. Optim., v.21, p. 120-127, 2001.

SIGMUND, O.; MAUTE, K. **“Topology optimization approaches”**. Structural and Multidisciplinary Optimization, Springer Berlin Heidelberg, v. 48, n. 6, p. 1031–1055, 2013. ISSN 1615-147X. Disponível em: <<http://dx.doi.org/10.1007/s00158-013-0978-6>>.

SILVA, E. C. N. **Otimização Aplicada ao Projeto de Sistemas Mecânicos**. [S.l.]: Escola Politécnica da Universidade de São Paulo, 2003. Online; accessed em 16- junho-2015.

SILVA, EMÍLIO CARLOS NELLI. **Otimização aplicada ao projeto de sistemas mecânicos**. São Paulo: Departamento de Engenharia Mecatrônica e Sistemas Mecânicos da Escola Politécnica da USP, 2000. 44 p.

SIMONETI, H. **Otimização topológica de estruturas elásticas 2d via mec baseada em uma nova abordagem evolucionária.** 2016. 162 f. Tese (doutorado) - programa de Pós-Graduação do Departamento de Engenharia Civil da Escola de Minas da Universidade Federal de Ouro Preto, Brasil, 2018.

SOH,C.K.;YANG,J.,**Fuzzy controlled genetic algorithm search for shape optimization,** *Journal of Computing in Civil Engineering*, v. 10, n. 2, p. 143-214, (1996).

SONMEZ, F.O.. **Structural Optimization Using Simulated Annealing,** *Departamento of Mechanical Engineering*, Bogazici University, Turkey, pp. 282-306, (2008).

STEFFENS, L.M., **Desenvolvimento de uma metodologia Integrada para Otimização de Forma e Mecânica de Fluidos,** Dissertação de Mestrado, Universidade Federal de Santa Catarina,Florianópolis, (2005).

TANSKANEN, P., **The evolutionary structural optimization method: theoretical aspects.** *Comput. Methods Appl. Mech. Eng.*, 191:5485–5498, 2002.

TAYLOR R, L; ZIENKIEWICZ, O.C. **The finite element method: The basis.** 5. Ed. Oxford, Butterworth Heinemann, 2000. v.1.

VANDERPLAAST, G. N. **Structural optimization for statics, dynamics and beyond.** International Symposium on Dynamic Problems of Mechanics, Ouro Preto. MG, v. 18, 2006.

VANDERPLAATS, G. N. E SALAJEGHEH, E., “**New Approximation Method for Stress Constraints in Structural Synthesis**”, *AIAA Journal*, vol. 27, n° 3, pp. 352-358, (1987).

VOTH, B., **Using automatically generated shape variables to optimize stamped plates.** Altair Internal Report Report, Troy, MI (1999).

WU, S. J. E CHOW, P. T., **Genetic algorithms for nonlinear mixed discrete- integer optimization problems via meta-genetic parameter optimization,** *Engineering Optimization*, 24, pp. 137–159,(1995).

XIE, Y.M, STEVEN, G.P. **Evolutionary structural optimization for dynamic problems.** *Comp. Struct.*, v. 58, p. 1067–1073, 1996.

XIE, Y.M., STEVEN, G.P., *Evolutionary Structural Optimization.* Springer, 1997.

YANG, X.Y., XIE, Y.M., STEVEN, G.P., QUERIN, O.M., **Bidirectional evolutionary method for stiffness optimization,** *AIAA Journal*, 37:1483–1488, 1999.

YOUNG, V; QUERIN, O.M; STEVEN, G.P; XIE, Y.P. **3D and multiple load case bi-directional evolutionary structural optimization (BESO).** *Structural Optimization*, v.18, p. 183-192, 1999.

ZAKHAMA, R; ABDALLA, M; GURDAL, Z; SMAOUI, H. **Wind load effect in topology optimization problems.** *Journal of Physics*, v. 75, n.1, p.12-48, 2007.

ZHOU, M., PAGALDIPT, N., THOMAS, H.L., SHYY, Y.K. **An integrated Approach to Topology, Sizing and Shape Optimization, Structural and Multidisciplinary Optimization**, v.26, p.308-318, (2004).

ZHU, J. H; ZHANG, H.Z; XIA, L. **Topology optimization in aircraft and aerospace structures design**. Arch. Computat. Methods Eng., v. 23, p. 595–622, 2016.

## APÊNDICE A – Código SIMP para a Viga MBB

```

# Programa "top", otimização topológica da viga MBB, Método SIMP
# Baseado em A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLE SIGMUND, 1999
# Entrada de dados
nelx = 60
nely = 20
volfrac = 0.5
penal = 3
rmin = 1.5
#
x=zeros(nely,nelx)
for q=1:nely
  for r=1:nelx
    global x[q,r]=volfrac
  end
end
loop = 0
change = 1
while change > 0.01
  global loop = loop + 1
  xold = x
# FE-ANALYSIS
E = 1
nu = 0.3
k=[ 1/2-nu/6 1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 [
-1/4+nu/12 -1/8-nu/8 nu/6 1/8-3*nu/8]]
KE = E/(1-nu^2)*[ k[1] k[2] k[3] k[4] k[5] k[6] k[7] k[8]
k[2] k[1] k[8] k[7] k[6] k[5] k[4] k[3]
k[3] k[8] k[1] k[6] k[7] k[4] k[5] k[2]
k[4] k[7] k[6] k[1] k[8] k[3] k[2] k[5]
k[5] k[6] k[7] k[8] k[1] k[2] k[3] k[4]
k[6] k[5] k[4] k[3] k[2] k[1] k[8] k[7]
k[7] k[4] k[5] k[2] k[3] k[8] k[1] k[6]

```

```

k[8] k[3] k[2] k[5] k[4] k[7] k[6] k[1]]
K = zeros(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1))
F = zeros(2*(nely+1)*(nelx+1),1)
U = zeros(2*(nely+1)*(nelx+1),1)
for elx = 1:nelx
  for ely = 1:nely
    n1 = (nely+1)*(elx-1)+ely
    n2 = (nely+1)*elx+ely
    edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1; 2*n2+2; 2*n1+1; 2*n1+2]
    for q=1:length(edof)
      for r=1:length(edof)
        contq = convert(Int, edof[q])
        contr = convert(Int, edof[r])
        K[contq,contr] = K[contq,contr]+(x[ely,elx]^penal)*KE[q,r]
      end
    end
  end
end
end
F[2,1] = -1
aux = convert(Int64, ceil(nely+2))
uni = zeros(aux)
for q=1:ceil(nely+1)
  uni[convert(Int64, q)]=1+2*(q-1)
end
uni[aux]=2*(nelx+1)*(nely+1)
fixeddofs = uni
alldofs = zeros(2*(nely+1)*(nelx+1))
for q=1:2*(nely+1)*(nelx+1)
  alldofs[q]=q
end
freedofs = setdiff(alldofs,fixeddofs)
Uaux=zeros(length(freedofs))
Faux=zeros(length(freedofs))
Kaux=zeros(length(freedofs),length(freedofs))

```

```

for q=1:length(freedofs)
    contq = convert(Int, freedofs[q])
    Faux[q]=F[contq]
    for r=1:length(freedofs)
        contr = convert(Int, freedofs[r])
        Kaux[q,r]=K[contq,contr]
    end
end
# SOLVING
Uaux=Kaux\Faux
for q=1:length(Uaux)
    contq = convert(Int, freedofs[q])
    U[contq] = Uaux[q]
end
for q=1:length(fixeddofs)
    contq = convert(Int, fixeddofs[q])
    U[contq] = 0
end
c=0
dc = zeros(nely,nelx)
for ely=1:nely
    for elx=1:nelx
        n1=(nely+1)*(elx-1)+ely
        n2 = (nely+1)*elx+ely
        Ue = U[[2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1;2*n2+2; 2*n1+1;2*n1+2],1]
        c = c+x[ely,elx]^penal*Ue'*KE*Ue
        dc[ely,elx] = -penal*x[ely,elx]^(penal-1)*Ue'*KE*Ue
    end
end
#
dcn=zeros(nely,nelx)
for i = 1:nelx
    for j = 1:nely
        sum=0

```



```

for k = max(i-floor(rmin),1):min(i+floor(rmin),nelx)
  for l = max(j-floor(rmin),1):min(j+floor(rmin),nely)
    fac = rmin-sqrt((i-k)^2+(j-l)^2)
    sum = sum+max(0,fac)
    k = convert(Int, k)
    l = convert(Int, l)
    dcn[j,i] = dcn[j,i] + max(0,fac)*x[l,k]*dc[l,k]
  end
end
dcn[j,i] = dcn[j,i]/(x[j,i]*sum)
end
end
dc = dcn
l1 = 0
l2 = 100000
move = 0.2
while l2-l1 > 1e-4
  lmid = 0.5*(l2+l1)
  global xnew=zeros(nely,nelx)
  for q=1:nely
    for r=1:nelx
      xnew[q,r]=max(0.001,max(x[q,r]-move,min(1,min(x[q,r]+move,x[q,r][
        ]*sqrt(-dc[q,r]/lmid))))))
    end
  end
  if sum(sum(xnew)) - volfrac*nelx*nely > 0
    l1 = lmid
  else
    l2 = lmid
  end
end
end
global x = xnew
#
global change=0;

```

```
for q=1:nely
    for r=1:nelx
        aux = x[q,r]-xold[q,r]
        aux=abs(aux)
        if aux>change
            change=aux
            global change
        end
    end
end
#
end
# Informações impressas a cada iteração
display('\n')
display("It.:")
display(loop)
display("Obj.: ")
display(c)
display("Vol.: ")
display(sum(sum(x))/(nelx*nely))
display("ch.: ")
display(change)
end
# Plotagem
using Colors
plotd = Gray.(x)
```

## APÊNDICE B – Código SIMP para a Viga SC

```

# Programa "top", otimização topológica da viga MBB, Método SIMP
# Baseado em A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLE SIGMUND,
2000
# Entrada de dados
nelx = 32
nely = 20
volfrac = 0.4
penal = 3
rmin = 1.2
#
x=zeros(nely,nelx)
for q=1:nely
  for r=1:nelx
    global x[q,r]=volfrac
  end
end
loop = 0
change = 1
while change > 0.01
  global loop = loop + 1
  xold = x
# FE-ANALYSIS
E = 1
nu = 0.3
k=[ 1/2-nu/6 1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 [
-1/4+nu/12 -1/8-nu/8 nu/6 1/8-3*nu/8]]
KE = E/(1-nu^2)*[ k[1] k[2] k[3] k[4] k[5] k[6] k[7] k[8]
k[2] k[1] k[8] k[7] k[6] k[5] k[4] k[3]
k[3] k[8] k[1] k[6] k[7] k[4] k[5] k[2]
k[4] k[7] k[6] k[1] k[8] k[3] k[2] k[5]
k[5] k[6] k[7] k[8] k[1] k[2] k[3] k[4]
k[6] k[5] k[4] k[3] k[2] k[1] k[8] k[7]

```

```

k[7] k[4] k[5] k[2] k[3] k[8] k[1] k[6]
k[8] k[3] k[2] k[5] k[4] k[7] k[6] k[1]]
K = zeros(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1))
F = zeros(2*(nely+1)*(nelx+1),1)
U = zeros(2*(nely+1)*(nelx+1),1)
for elx = 1:nelx
    for ely = 1:nely
        n1 = (nely+1)*(elx-1)+ely
        n2 = (nely+1)*elx+ely
        edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1; 2*n2+2; 2*n1+1; 2*n1+2]
        for q=1:length(edof)
            for r=1:length(edof)
                contq = convert(Int, edof[q])
                contr = convert(Int, edof[r])
                K[contq,contr] = K[contq,contr]+(x[ely,elx]^penal)*KE[q,r]
            end
        end
    end
end
F[convert(Int, 2*(nelx+1)*(nely+1)),1] = -1
fixeddofs=zeros(2*(nely+1))
for i=1:2*(nely+1)
    fixeddofs[i] = i
end
alldofs = zeros(2*(nely+1)*(nelx+1))
for q=1:2*(nely+1)*(nelx+1)
    alldofs[q]=q
end
freedofs = setdiff(alldofs,fixeddofs)
Uaux=zeros(length(freedofs))
Faux=zeros(length(freedofs))
Kaux=zeros(length(freedofs),length(freedofs))
for q=1:length(freedofs)
    contq = convert(Int, freedofs[q])

```

```

Faux[q]=F[contq]
for r=1:length(freedofs)
    contr = convert(Int, freedofs[r])
    Kaux[q,r]=K[contq,contr]
end
end
# SOLVING
using TickTock
tick()
Uaux=Kaux\Faux
tock()
for q=1:length(Uaux)
    contq = convert(Int, freedofs[q])
    U[contq] = Uaux[q]
end
for q=1:length(fixeddofs)
    contq = convert(Int, fixeddofs[q])
    U[contq] = 0
end
c=0
dc = zeros(nely,nelx)
for ely=1:nely
    for elx=1:nelx
        n1=(nely+1)*(elx-1)+ely
        n2 = (nely+1)*elx+ely
        Ue = U[[2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1;2*n2+2; 2*n1+1;2*n1+2],1]
        c = c+x[ely,elx]^penal*Ue'*KE*Ue
        dc[ely,elx] = -penal*x[ely,elx]^(penal-1)*Ue'*KE*Ue
    end
end
end
#
dcn=zeros(nely,nelx)
for i = 1:nelx
    for j = 1:nely

```

```

sum=0
for k = max(i-floor(rmin),1):min(i+floor(rmin),nelx)
  for l = max(j-floor(rmin),1):min(j+floor(rmin),nely)
    fac = rmin-sqrt((i-k)^2+(j-l)^2)
    sum = sum+max(0,fac)
    k = convert(Int, k)
    l = convert(Int, l)
    dcn[j,i] = dcn[j,i] + max(0,fac)*x[l,k]*dc[l,k]
  end
end
dcn[j,i] = dcn[j,i]/(x[j,i]*sum)
end
end
dc = dcn
l1 = 0
l2 = 100000
move = 0.2
while l2-l1 > 1e-4
  lmid = 0.5*(l2+l1)
  global xnew=zeros(nely,nelx)
  for q=1:nely
    for r=1:nelx
      xnew[q,r]=max(0.001,max(x[q,r]-move,min(1,min(x[q,r]+move,x[q,r][
        ]*sqrt(-dc[q,r]/lmid))))))
    end
  end
  if sum(sum(xnew)) - volfrac*nelx*nely > 0
    l1 = lmid
  else
    l2 = lmid
  end
end
global x = xnew
#

```

```
global change=0;
for q=1:nely
    for r=1:nelx
        aux = x[q,r]-xold[q,r]
        aux=abs(aux)
        if aux>change
            change=aux
        global change
    end
end
#
end
# Informações impressas a cada iteração
display('\n')
display("It.:")
display(loop)
display("Obj.: ")
display(c)
display("Vol.: ")
display(sum(sum(x))/(nelx*nely))
display("ch.: ")
display(change)
end
# Plotagem
using Colors
plotd = Gray.(x)
```

### APÊNDICE C – Código BESO para a Viga MBB

```

# Programa "softbeso" com alteração para viga MBB, Método Beso
# Baseado em A SOFT-KILL BESO CODE BY X. HUANG and Y.M. Xie, 2007
# Entrada de dados
nelx = 60
nely = 20
volfrac = 0.5
er = 0.05
rmin = 1.5
#
penal = 3
x=zeros(nely,nelx)
for q=1:nely
  for r=1:nelx
    global x[q,r]=1
  end
end
vol = 1
i = 0
change = 1
c=zeros(10000)
while change > 0.001
  global i = i + 1
  global vol = max(vol*(1-er),volfrac)
  if i > 1
    olddc = dc
  end
# FE-ANALYSIS
E = 1
nu = 0.3
k=[ 1/2-nu/6 1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 [
-1/4+nu/12 -1/8-nu/8 nu/6 1/8-3*nu/8]]
KE = E/(1-nu^2)*[ k[1] k[2] k[3] k[4] k[5] k[6] k[7] k[8]

```



```

k[2] k[1] k[8] k[7] k[6] k[5] k[4] k[3]
k[3] k[8] k[1] k[6] k[7] k[4] k[5] k[2]
k[4] k[7] k[6] k[1] k[8] k[3] k[2] k[5]
k[5] k[6] k[7] k[8] k[1] k[2] k[3] k[4]
k[6] k[5] k[4] k[3] k[2] k[1] k[8] k[7]
k[7] k[4] k[5] k[2] k[3] k[8] k[1] k[6]
k[8] k[3] k[2] k[5] k[4] k[7] k[6] k[1]]
K = zeros(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1))
F = zeros(2*(nely+1)*(nelx+1),1)
U = zeros(2*(nely+1)*(nelx+1),1)
for elx = 1:nelx
  for ely = 1:nely
    n1 = (nely+1)*(elx-1)+ely
    n2 = (nely+1)*elx+ely
    edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1; 2*n2+2; 2*n1+1; 2*n1+2]
    for q=1:length(edof)
      for r=1:length(edof)
        contq = convert(Int, edof[q])
        contr = convert(Int, edof[r])
        K[contq,contr] = K[contq,contr]+(x[ely,elx]^penal)*KE[q,r]
      end
    end
  end
end
end
F[2,1] = -1
aux = convert(Int64, ceil(nely+2))
uni = zeros(aux)
for q=1:ceil(nely+1)
  uni[convert(Int64, q)]=1+2*(q-1)
end
uni[aux]=2*(nelx+1)*(nely+1)
fixeddofs = uni
alldofs = zeros(2*(nely+1)*(nelx+1))
for q=1:2*(nely+1)*(nelx+1)

```

```

    alldofs[q]=q
end
freedofs = setdiff(alldofs,fixeddofs)
Uaux=zeros(length(freedofs))
Faux=zeros(length(freedofs))
Kaux=zeros(length(freedofs),length(freedofs))
for q=1:length(freedofs)
    contq = convert(Int, freedofs[q])
    Faux[q]=F[contq]
    for r=1:length(freedofs)
        contr = convert(Int, freedofs[r])
        Kaux[q,r]=K[contq,contr]
    end
end
end
# SOLVING
Uaux=Kaux\Faux
for q=1:length(Uaux)
    contq = convert(Int, freedofs[q])
    U[contq] = Uaux[q]
end
for q=1:length(fixeddofs)
    contq = convert(Int, fixeddofs[q])
    U[contq] = 0
end
global dc = zeros(nely,nelx)
for ely=1:nely
    for elx=1:nelx
        n1=(nely+1)*(elx-1)+ely
        n2 = (nely+1)*elx+ely
        global Ue = U[[2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1;2*n2+2; 2*n1+1;2*n1+2],1]
        global c[i] = c[i]+0.5*x[ely,elx]^penal*Ue'*KE*Ue
        global dc[ely,elx] = 0.5*x[ely,elx]^(penal-1)*Ue'*KE*Ue
    end
end
end

```

```

#
global dcf=zeros(nely,nelx)
for i = 1:nelx
  for j = 1:nely
    sum=0
    for k = max(i-floor(rmin),1):min(i+floor(rmin),nelx)
      for l = max(j-floor(rmin),1):min(j+floor(rmin),nely)
        fac = rmin-sqrt((i-k)^2+(j-l)^2)
        sum = sum+max(0,fac)
        k = convert(Int, k)
        l = convert(Int, l)
        global dcf[j,i] = dcf[j,i] + max(0,fac)*dc[l,k]
      end
    end
    global dcf[j,i] = dcf[j,i]/sum
  end
end
global dc = dcf
# Estabilização do processo evolucionário
if i > 1
  global dc = (dc+olddc)/2
end
#
l1 = dc[1,1]
l2 = dc[1,1]
for i=1:nely
  for j=1:nelx
    if l1 > dc[i,j]
      l1 = dc[i,j]
    end
  end
end
end
for i=1:nely
  for j=1:nelx

```

```

    if l2 < dc[i,j]
        l2 = dc[i,j]
    end
end
end
while (l2-l1)/l2 > 1e-5
    global th = (l2+l1)/2
    for q=1:nely
        for r=1:nelx
            global x[q,r] = max(0.001,sign(dc[q,r]-th))
        end
    end
    if sum(sum(x)) - vol*nelx*nely > 0
        l1 = th
    else
        l2 = th
    end
end
#
if i>10
    aux1 = convert(Int, (i-9))
    aux2 = convert(Int, (i-5))
    aux3 = convert(Int, (i-4))
    aux4 = sum(c[aux1:aux2])
    aux5 = sum(c[aux3:i])
    aux6 = sum(c[aux3:i])
    global change=abs(aux4-aux5)/aux6
end
#
# Informações impressas a cada iteração
display("\n")
display("It.:")
display(i)
display("Obj.: ")

```

```
display(c[i])
display("Vol.: ")
display((sum(sum(x)))/(nelx*nely))
display("ch.: ")
display(change)
end
# Plotagem
using Colors; Gray.(x)
```

## APÊNDICE D – Código BESO para a Viga SC

```

# Programa "softbeso" com alteração para viga TLSC, Método Beso
# Baseado em A SOFT-KILL BESO CODE BY X. HUANG and Y.M. Xie, 2007
# Entrada de dados
nelx = 32
nely = 20
volfrac = 0.4
er = 0.01
rmin = 1.2
#
penal = 3
x=zeros(nely,nelx)
for q=1:nely
  for r=1:nelx
    global x[q,r]=1
  end
end
vol = 1
i = 0
change = 1
c=zeros(10000)
while change > 0.001
  global i = i + 1
  global vol = max(vol*(1-er),volfrac)
  if i > 1
    olddc = dc
  end
# FE-ANALYSIS
E = 1
nu = 0.3
k=[ 1/2-nu/6 1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 [
-1/4+nu/12 -1/8-nu/8 nu/6 1/8-3*nu/8]]
KE = E/(1-nu^2)*[ k[1] k[2] k[3] k[4] k[5] k[6] k[7] k[8]

```

```

k[2] k[1] k[8] k[7] k[6] k[5] k[4] k[3]
k[3] k[8] k[1] k[6] k[7] k[4] k[5] k[2]
k[4] k[7] k[6] k[1] k[8] k[3] k[2] k[5]
k[5] k[6] k[7] k[8] k[1] k[2] k[3] k[4]
k[6] k[5] k[4] k[3] k[2] k[1] k[8] k[7]
k[7] k[4] k[5] k[2] k[3] k[8] k[1] k[6]
k[8] k[3] k[2] k[5] k[4] k[7] k[6] k[1]]
K = zeros(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1))
F = zeros(2*(nely+1)*(nelx+1),1)
U = zeros(2*(nely+1)*(nelx+1),1)
for elx = 1:nelx
    for ely = 1:nely
        n1 = (nely+1)*(elx-1)+ely
        n2 = (nely+1)*elx+ely
        edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1; 2*n2+2; 2*n1+1; 2*n1+2]
        for q=1:length(edof)
            for r=1:length(edof)
                contq = convert(Int, edof[q])
                contr = convert(Int, edof[r])
                K[contq,contr] = K[contq,contr]+(x[ely,elx]^penal)*KE[q,r]
            end
        end
    end
end
F[convert(Int, 2*(nelx+1)*(nely+1)),1] = -1
fixeddofs=zeros(2*(nely+1))
for i=1:2*(nely+1)
    fixeddofs[i] = i
end
alldofs = zeros(2*(nely+1)*(nelx+1))
for q=1:2*(nely+1)*(nelx+1)
    alldofs[q]=q
end
freedofs = setdiff(alldofs,fixeddofs)

```

```

Uaux=zeros(length(freedofs))
Faux=zeros(length(freedofs))
Kaux=zeros(length(freedofs),length(freedofs))
for q=1:length(freedofs)
    contq = convert(Int, freedofs[q])
    Faux[q]=F[contq]
    for r=1:length(freedofs)
        contr = convert(Int, freedofs[r])
        Kaux[q,r]=K[contq,contr]
    end
end
# SOLVING
display("antes")
Uaux=Kaux\Faux
display("depuis")
for q=1:length(Uaux)
    contq = convert(Int, freedofs[q])
    U[contq] = Uaux[q]
end
for q=1:length(fixeddofs)
    contq = convert(Int, fixeddofs[q])
    U[contq] = 0
end
global dc = zeros(nely,nelx)
for ely=1:nely
    for elx=1:nelx
        n1=(nely+1)*(elx-1)+ely
        n2 = (nely+1)*elx+ely
        global Ue = U[[2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1;2*n2+2; 2*n1+1;2*n1+2],1]
        global c[i] = c[i]+0.5*x[ely,elx]^penal*Ue'*KE*Ue
        global dc[ely,elx] = 0.5*x[ely,elx]^(penal-1)*Ue'*KE*Ue
    end
end
end
#

```



```

global dcf=zeros(nely,nelx)
for i = 1:nelx
    for j = 1:nely
        sum=0
        for k = max(i-floor(rmin),1):min(i+floor(rmin),nelx)
            for l = max(j-floor(rmin),1):min(j+floor(rmin),nely)
                fac = rmin-sqrt((i-k)^2+(j-l)^2)
                sum = sum+max(0,fac)
                k = convert(Int, k)
                l = convert(Int, l)
                global dcf[j,i] = dcf[j,i] + max(0,fac)*dc[l,k]
            end
        end
        global dcf[j,i] = dcf[j,i]/sum
    end
end
global dc = dcf
# Estabilização do processo evolucionário
if i > 1
    global dc = (dc+olddc)/2
end
#
l1 = dc[1,1]
l2 = dc[1,1]
for i=1:nely
    for j=1:nelx
        if l1 > dc[i,j]
            l1 = dc[i,j]
        end
    end
end
end
for i=1:nely
    for j=1:nelx
        if l2 < dc[i,j]

```

```

    l2 = dc[i,j]
end
end
end
while (l2-l1)/l2 > 1e-5
    global th = (l2+l1)/2
    for q=1:nely
        for r=1:nelx
            global x[q,r] = max(0.001,sign(dc[q,r]-th))
        end
    end
    if sum(sum(x)) - vol*nelx*nely > 0
        l1 = th
    else
        l2 = th
    end
end
#
if i>10
    aux1 = convert(Int, (i-9))
    aux2 = convert(Int, (i-5))
    aux3 = convert(Int, (i-4))
    aux4 = sum(c[aux1:aux2])
    aux5 = sum(c[aux3:i])
    aux6 = sum(c[aux3:i])
    global change=abs(aux4-aux5)/aux6
end
#
# Informações impressas a cada iteração
display("It.:")
display(i)
display("Obj.:")
display(c[i])
display("Vol.: ")

```

```
display((sum(sum(x)))/(nelx*nely))
display("ch.: ")
display(change)
end
# Plotagem
using Colors; Gray.(x)
plotd = Gray.(x)
```