

Instituto Federal de Educação, Ciência e Tecnologia de  
São Paulo – IFSP

Programa de Pós-Graduação do Mestrado

Acadêmico em Engenharia Mecânica

Departamento de Mecânica - Campus São Paulo



PAULO HENRIQUE LIXANDRÃO FERNANDO

**Estudo de Otimização Aplicado a Topologia de Furação de Longarina  
de Chassi Estrutural Veicular**

Dissertação

São Paulo, 2017

PAULO HENRIQUE LIXANDRÃO FERNANDO

**Estudo de Otimização Aplicado a Topologia de Furação de Longarina  
de Chassi Estrutural Veicular**

Dissertação do Programa de Pós-Graduação  
*Stricto Sensu* de Mestrado Acadêmico em En-  
genharia Mecânica apresentada ao Instituto  
Federal de Educação, Ciência e Tecnologia de  
São Paulo - *Campus* São Paulo como parte  
dos requisitos para a obtenção do título de  
Mestre em Engenharia Mecânica, área de con-  
centração: Processos de Fabricação.

Orientador: Prof. Dr. Francisco Yastami Nakamoto

São Paulo, 2017

Catalogação na fonte  
Biblioteca Francisco Montojos - IFSP Campus São Paulo  
Dados fornecidos pelo(a) autor(a)

F363e	<p>Fernando, Paulo Henrique Lixandrão Estudo de otimização aplicado a topologia de furação de longarina de chassi estrutural veicular / Paulo Henrique Lixandrão Fernando. São Paulo: [s.n.], 2017. 168 f.</p> <p style="text-align: center;">Orientador: Prof. Dr. Francisco Yastami Nakamoto</p> <p style="text-align: center;">Dissertação (Mestrado Acadêmico em Engenharia Mecânica) - Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, IFSP, 2017.</p> <p style="text-align: center;">1. Otimização Topológica. 2. Paretoworks. 3. Virtual.pyxis. 4. Matlab. 5. Simulação Estrutural. I. Instituto Federal de Educação, Ciência e Tecnologia de São Paulo II. Título.</p> <p>CDD 620.1</p>
-------	---

Dissertação defendida em 28 de *novembro* de 2017, com comissão avaliadora constituída pelos professores:

---

**Prof. Dr. Francisco Yastami  
Nakamoto**  
Orientador

---

**Professor Dr. Cícero Ribeiro de Lima**  
Membro convidado externo

---

**Professor Dr. Givanildo Alves dos  
Santos**  
Membro convidado interno

São Paulo, 2017



*Este trabalho é dedicado a disseminação do tema  
de otimização estrutural no Brasil.*

# Agradecimentos

Agradeço primeiramente à **Deus**.

À minha esposa, **PhD Kelly Cristina de Lira Lixandrão**, amor da minha vida, essência da minha inspiração, razão do meu viver e pessoa que sempre me motivou a buscar o melhor em meus estudos e pesquisas.

Aos meus pais, **Severino Félix Fernando** e **Rosangela Lixandrão Fernando**, aos meus sogros **Ivanete Luiza de Lira Silva** e **José Matos da Silva** e aos meus irmãos **Daniel** e **Vanessa Lixandrão** por toda a motivação e admiração.

Ao **Instituto Federal de Educação, Ciência e Tecnologia de São Paulo**, por meio do departamento de mecânica que possibilitaram a realização do presente trabalho por meio do programa de Pós-Graduação Stricto Sensu do Mestrado Acadêmico em Engenharia Mecânica.

Ao **Prof. Dr. Francisco Yastami Nakamoto**<sup>1</sup>, pela excelente orientação, conhecimento compartilhado e apoio no desenvolvimento do trabalho, e por sempre estar me incentivando a desenvolver novas pesquisas e artigos inovadores, acadêmicos e de estudos de aplicação na indústria.

Ao **Prof. Dr. Givanildo Alves dos Santos**<sup>2</sup>, por todo o suporte dado por meio das disciplinas e da estrutura do curso e por aceitar fazer parte da banca de qualificação e defesa contribuindo de forma significativa para os resultados deste texto de dissertação.

Ao **Prof. Dr. Cícero Ribeiro de Lima**<sup>3</sup>, por aceitar o convite de professor de banca externa tanto na qualificação, quanto na defesa do mestrado e por contribuir na qualificação com novas ideias para o trabalho, propondo melhorias e implementações na área de otimização topológica, sempre me incentivando nos desenvolvimentos desta ainda jovem e inovadora área da mecânica.

Ao **Prof. Dr. Eduardo Guy Perpetuo Bock - IFSP-SPO** e ao **Prof. Dr. Osvaldo Luís Asato - IFSP-SZN**, por aceitarem participar da banca como membros suplentes, e por contribuírem com o aprendizado de disciplinas relacionados ao texto e também no auxílio a correção e revisão de artigos.

---

<sup>1</sup> Diretor do Departamento de Mecânica do IFSP - SP <<http://www.ifsp.edu.br/>>

<sup>2</sup> Coordenador do Programa de Mestrado em Engenharia Mecânica do IFSP - SP <<http://www.ifsp.edu.br/>>

<sup>3</sup> Professor do Centro de Engenharia, Modelagem e Ciências Sociais Aplicadas (CECS) da Universidade Federal do ABC (UFABC) <<http://www.ufabc.edu.br/>>

Ao **Michel Santos Queiroz**, diretor de TI do campus SP do IFSP, que me auxiliou com as licenças dos softwares CAE acadêmicos.

Como suporte externo obtive o apoio de várias pessoas, empresas e universidades: **Prof. Dr. Krishnan Suresh** <sup>4</sup> juntamente com o **Eng.º Praveen Yadav** <sup>5</sup>, por me apoiar com as pesquisas internacionais na área de otimização topológica e por meio da empresa Sciart disponibilizar os recursos do programa computacional ParetoWorks.

**Eng.º Leandro Garbin** <sup>6</sup> e **Eng.º Paulo Nigro** <sup>7</sup>, por contribuírem pelo fornecimento do programa computacional Virtual.PYXIS para uso afins de realizar o trabalho e também pelo incentivo em novas pesquisas na área de otimização e também ao **Lucas Ramos** e **Igor Saveljevas** <sup>8</sup> pelo suporte na instalação do programa e ajuda com resultados.

**Eng.º Wilson Molina Ribas** <sup>9</sup>, que sempre me incentivou aos estudos de mestrado e disponibilizou horário flexível na empresa para que eu pudesse realizar as disciplinas.

**Eng.º Hugo Nicioli Cordeiro de Souza** <sup>10</sup>, pela discussão em aplicações para chassi estrutural com relação ao caminhão aplicado no estudo deste trabalho.

Ao **Eng.º MSc. Rafael Santos Iwamura** <sup>11</sup> por auxiliar na produção do algoritmo e sempre que eu o acionei procurou me mostrar o melhor caminho para o desenvolvimento e implemento dos algoritmos.

Pela amizade ainda agradeço aos colegas do IFSP-SP, em especial ao grupo de Automação e Integração de Sistemas ao qual eu me baseei inicialmente por não existir um grupo específico de otimização estrutural, que são **Ricardo Hideki Kubo**, **Diego Nogueira Guirro**, **Camila Neves**, **Irineu Carmelino da Silva Junior**, **Uanderson Rodrigues Costa Mazzoni**, **Alex** e **Plínio Sérgio Pires Monteiro**, além de outros colegas do curso que trabalham com simulação FEA como o **Carlos Eduardo Lino** e o professor e aluno **Daniel Pezza Tchernov**, e alunos de outros grupos como o **Raul Gaspari Santos**, **Wellington de Souza Lima**, **Rogério Lima de Souza** e o professor e aluno **Wagner de Campos Sabor**.

---

<sup>4</sup> Professor da Universidade de Winsconsin <<http://www.ersl.wisc.edu/>>

<sup>5</sup> Diretor da empresa Sciart <<https://www.sciartsoft.com/>>

<sup>6</sup> Diretor da empresa Virtual CAE <<http://www.virtualcae.com.br/>>

<sup>7</sup> Coordenador de Projetos da empresa Virtual CAE

<sup>8</sup> Equipe de suporte técnico da empresa Virtual CAE

<sup>9</sup> Diretor da empresa Lavrita Engenharia <<https://www.lavrita.com.br/>>

<sup>10</sup> Gerente de Produto da Scania

<sup>11</sup> Engenheiro de desenvolvimento de produto da Embraer

*“A ciência nunca resolve um problema sem criar pelo menos outros dez”*  
*(George Bernard Shaw)*

# Resumo

LIXANDRÃO FERNANDO, P. H. **Estudo de Otimização Aplicado a Topologia de Furação de Longarina de Chassi Estrutural Veicular**. 2017. 168p. Dissertação (Programa de Pós-Graduação em Engenharia Mecânica) - Instituto Federal de Educação, Ciência e Tecnologia São Paulo, São Paulo, 2017.

A necessidade atual de redução de custos de fabricação na indústria automotiva é fortemente solicitada aos engenheiros de produto e processo. Por este fato, engenheiros e projetistas buscam em outras áreas as técnicas modernas que podem ser introduzidas ou utilizadas em seus projetos, como por exemplo, os métodos computacionais para otimização aplicadas ao desenho das peças. Esta técnica atual é conhecida como “Otimização topológica”. Como a longarina do chassi do veículo é a ligação entre a cabine, motor, transmissão e eixos, este componente é de grande importância na construção do veículo. Para gerar bons produtos manufaturados, engenheiros das companhias tem elevados custos com tempo de desenvolvimento, assim, buscam desenvolver e projetar um produto com redução de peso do material suportado para a aplicação. A longarina do chassi é um dos mais importantes itens do veículo porque é onde é suportado todas as partes da estrutura no veículo automotivo. Este trabalho procurou identificar a otimização topológica aplicada na longarina do chassi de um caminhão, com o objetivo de desenvolver nova topologia para esta longarina pela técnica de otimização topológica estrutural. Comparou-se os resultados com alguns programas computacionais comerciais e alguns algoritmos propostos para a aplicação para validar o desenvolvimento da nova topologia, por meio de recursos dos programas computacionais como o do ParetoWorks (Sciart) e Virtual.PYXIS (Virtual CAE), além disto, demonstrou-se a programação de algoritmos para esta aplicação com o programa computacional MATLAB® (MathWorks). Como resultado houve uma possibilidade redução de massa de 38,58%, sendo que o critério de análise baseou-se apenas na análise estática do objeto de estudo. Os resultados contribuíram para uma nova aplicação no meio acadêmico e uso industrial.

**Palavras-chave:** Otimização Topológica. ParetoWorks. Virtual.PYXIS. MATLAB®.

# Abstract

LIXANDRÃO FERNANDO, P. H. **Study of Optimization Applied in Topology of Vehicle Structural Stringer Chassi Drilling**. 2017. 168p. Master Thesis (Programa de Pós-Graduação em Engenharia Mecânica) - Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, São Paulo, 2017.

The necessity to reduce production costs in the automotive industry is strongly investigated by product engineers. For this reason, engineers and designers search in other areas modern techniques that can be improved or used in their projects, using computational methods for optimization applied to the part design. This actual technique is well-known with Topology Optimization. As the vehicle chassis is the link between the cabin, engine, transmission and axles, it has a great importance in the construction of vehicle. For the development of production, engineering companies have high costs due to the time it takes to get a better product, and it is very difficult to develop and design a product with the lightweight of the material that suits to the application. The stringer chassis is one of the most important items of the vehicle because that is where it supports all parts of the structure in the propellant vehicle. This work searched to identify such optimization technology, and applies it in the analysis of a chassis of a truck, with the objective of developing a drilling plan for this stringer by overlapping planes using the topological optimization technique. It will be comparing the results with some commercial software and with others algorithms proposed for the application to validate the development of this drilling plan, using for comparison some computer program that have some optimization type incorporated such as ParetoWorks (Sciart) and Virtual.PYXIS (VirtualCAE), demonstrating the programming of the algorithms for this application with the use of the MATLAB® (MathWorks) computer program. As a result with possibility there was make a mass reduction of 38.58%, and the analysis criterion was based only on the static analysis of the object of study. The results can contribute to a new application in academic and industrial use.

**Key-words:** Topology optimization. ParetoWorks. Virtual.PYXIS. MATLAB®.

# Lista de ilustrações

Figura 1	Otimização estrutural obtida por Michell (1904). Fonte: (RAMOS, 2007)	28
Figura 2	Classes de otimização estrutural. Fonte: (BENDSØE; SIGMUND, 2003).	30
Figura 3	Resumo dos tipos atuais de otimização estrutural. Fonte: Do autor com base em Jenkins (2015), Amigo (2013) Sigmund e Maute (2013).	34
Figura 4	Princípio de otimização topológica com região vazia, clara (0), região cheia, escura (1). Fonte: (SILVA, 2003b)	39
Figura 5	Rigidez Relativa vs densidade de volume para o material SIMP para diferentes valores de penalização. Fonte: (ESCHENAUER; OLHOFF, 2001).	43
Figura 6	Fatores utilizados no LS. Fonte: (ALPHEN, 2017)	46
Figura 7	(a) Tabuleiro de xadrez, (b) Solução em ilhas, (c) Dependência de malha (240x40 elementos), (d) Dependência de malha (180x30 elementos). Fonte: (CARVALHO; ALMEIDA, 2013)	49
Figura 8	Problema do <i>checkerboard</i> (SILVA, 2003b)	50
Figura 9	Ótimo local ou global. Fonte: Adaptado de Rao (2009).	52
Figura 10	Principais otimizadores baseados em gradientes para otimização estrutural. Fonte: (GRIVA; NASH; SOFER, 2009).	54
Figura 11	Principais métodos de otimização. Fonte: Adaptado de Rao (2009).	57
Figura 12	Tipos de malhas bidimensionais. Fonte (SURESH, 2016)	59
Figura 13	Formato dos elementos bidimensionais. Fonte: (LIMA, 2002)	59
Figura 14	Elementos finitos quadriláteros nas coordenadas locais ( $\xi, \eta$ ). Fonte: (AHMED, 2013).	60
Figura 15	Funções de otimização no MATLAB® Fonte: Adaptado de Messac (2015).	68
Figura 16	Sintetizações com CO. (a)Viga MBB Simétrica, (b)Viga curta em balanço, (c)Viga em balanço com furo fixado. Fonte: (AHMED, 2013).	70
Figura 17	(a)Viga MBB Simétrica, (b)Viga curta em balanço, (c)Viga em balanço com furo fixado. Fonte: (AHMED, 2013).	70
Figura 18	(a)topologia tradicional, (b)formulação proposta. Fonte: Adaptado de Zhao (2016).	71
Figura 19	Tipos de domínio do projeto para a viga MBB. Fonte: (BENDSØE; SIGMUND, 2003).	73
Figura 20	Programa principal. Fonte: (SIGMUND, 2001).	74
Figura 21	Rotina CO. Fonte: (SIGMUND, 2001).	75

Figura 22	Programa do Filtro de independência de malha. Fonte: (SIGMUND, 2001). . . . .	75
Figura 23	Distribuição dos elementos. Fonte: (ANDREASSEN et al., 2011). . . .	76
Figura 24	Código para o MEF. Fonte: (SIGMUND, 2001). . . . .	77
Figura 25	Programa Principal para o LS. Fonte: (CHALLIS; GUEST, 2009). . . .	79
Figura 26	Reinicialização da função LS. Fonte: (CHALLIS; GUEST, 2009). . . .	79
Figura 27	Atualização do domínio de projeto de LS. Fonte: (CHALLIS; GUEST, 2009). . . . .	80
Figura 28	Evolução da função de LS. Fonte: (CHALLIS; GUEST, 2009). . . . .	80
Figura 29	Implementação da análise FEA. Fonte: (CHALLIS; GUEST, 2009). . .	81
Figura 30	Matriz de rigidez dos elementos. Fonte: (CHALLIS; GUEST, 2009). . .	81
Figura 31	Tipos de carrocerias aplicados no chassi. Fonte: Adaptado de (SCANIA(a), 2016). . . . .	85
Figura 32	Chassi 4x4. Fonte: Adaptado de (SCANIA(a), 2016). . . . .	85
Figura 33	Encurtamento ou prolongamento de entre-eixo. Fonte: Adaptado de (SCANIA(b), 2016). . . . .	86
Figura 34	Regras e condições de furação no chassi. Unidade:mm . Fonte: Adaptado de (SCANIA(b), 2016). . . . .	87
Figura 35	Padrão permitido de furação na longarina. Fonte: Adaptado de (SCANIA(c), 2016). . . . .	87
Figura 36	Estrutura de pesquisa. Fonte: Do autor. . . . .	88
Figura 37	Área de interesse da pesquisa. Fonte: Do autor . . . . .	89
Figura 38	Especificações do computador. Fonte: Do autor . . . . .	91
Figura 39	Dimensões da longarina do chassi estudada. Fonte: Do autor . . . . .	91
Figura 40	Características do perfil. Unidade (mm). Fonte: Do autor . . . . .	92
Figura 41	Longarina com EE 4.100 mm e 4.300 mm. Fonte: Do autor . . . . .	92
Figura 42	Longarina com EE 4.500 mm e 4.700 mm. Fonte: Do autor . . . . .	93
Figura 43	Modelo 2D do chassi comercial no AutoCad. Fonte: Do autor . . . . .	94
Figura 44	Modelo 3D do chassi comercial no SolidWorks. Fonte: Do Autor. . . . .	94
Figura 45	Diagrama do MOT utilizado para o SIMP. Fonte: Adaptado de Libera e Lima (2012). . . . .	96
Figura 46	Diagrama do MOT utilizado para o LS. Fonte: Adaptado de Wu e Wu (2016). . . . .	96
Figura 47	Exemplo para definir cargas distribuídas nos nós do domínio. Fonte: Do autor . . . . .	98
Figura 48	Exemplo de carga no elemento 6. Fonte: Do autor . . . . .	99
Figura 49	Alteração de código para longarina do chassi baseado no código SIMP. Fonte: Do autor . . . . .	99



Figura 50	Alteração de código para longarina do chassi baseado no código LS. Fonte: Do autor . . . . .	100
Figura 51	Novas implementações SIMP e LS com malha 60x30, fração de volume de 50%, $p=3$ , $r_{min}=1,5$ , $StepLength=3$ , $numReinit=2$ e $topWeight=2$ . Fonte: Do Autor. . . . .	100
Figura 52	Exemplo de veículo especial. Fonte: Cortesia Lavrita Engenharia (LA-VRITA, 2016). . . . .	101
Figura 53	Longarina Scania P440 CB 4x4. Fonte: Do autor . . . . .	102
Figura 54	Características LNE 380. Fonte: Site Benafer . . . . .	103
Figura 55	Cargas e apoios da estrutura. $L= 6.833$ mm. Carga = 100.000N. Não há simetria na estrutura. Fonte: Do autor . . . . .	103
Figura 56	a) Representação do perfil 3D b) e perfil 2D para as mesmas condições de contorno. Fonte: Do autor . . . . .	104
Figura 57	Linha Elástica do Perfil Tridimensional. Fonte: Do autor . . . . .	104
Figura 58	Linha Elástica do Perfil Bidimensional. Fonte: Do autor . . . . .	105
Figura 59	Dimensão do elemento da estrutura. Fonte: Do autor . . . . .	106
Figura 60	Posicionamento dos nós. Fonte: Do autor . . . . .	107
Figura 61	Posicionamento dos nós em EE 4.100 e 4.300mm. Fonte: Do autor . . . . .	107
Figura 62	Posicionamento dos nós em EE 4.500 e 4.700mm Fonte: Do autor . . . . .	108
Figura 63	Algoritmo SIMP sintetizado com $V=70\%$ . Fonte: Do Autor. . . . .	110
Figura 64	Aparecimento do formato <i>hinge</i> . Fonte: Do autor . . . . .	110
Figura 65	Chassi sintetizado para: a) EE 3.900mm com $N=503 \times 20$ , b) EE 4.100mm com $N=518 \times 20$ , c)EE 4.300mm com $N=533 \times 20$ , d)EE 4.500mm com $N=547 \times 20$ , e)EE 4.700mm com $N=562 \times 20$ , $V=70\%$ , $p=3$ , $r_{min}=1,5$ . Fonte: Do Autor. . . . .	111
Figura 66	Sintetização SIMP implementado com diversas frações de volume. Fonte: Do Autor. . . . .	113
Figura 67	Modelo padrão para relações de aspecto da longarina. Fonte: Do Autor. . . . .	114
Figura 68	Modelo padrão para relações de aspecto da longarina. Fonte: Do Autor. . . . .	115
Figura 69	Algoritmo LS sintetizado com $V=70\%$ . Fonte: Do Autor. . . . .	117
Figura 70	Chassi sintetizado para: a) EE 3.900mm com $N=503 \times 20$ , b) EE 4.100mm com $N=518 \times 20$ , c)EE 4.300mm com $N=533 \times 20$ , d)EE 4.500mm com $N=547 \times 20$ , e)EE 4.700mm com $N=562 \times 20$ , $V=70\%$ , $steplength=3$ , $numReinit=2$ e $topWeight=2$ . Fonte: Do Autor. . . . .	118
Figura 71	Convergência da Fração de Volume para o LS implementado. Fonte: Do Autor. . . . .	120
Figura 72	Sintetização LS implementado com diversas frações de volume. Fonte: Do Autor. . . . .	121

Figura 73	Otimizadores do Virtual.PYXIS. Fonte: Adaptado do manual de Virtual.PYXIS (2016). . . . .	122
Figura 74	Solvers do Virtual.PYXIS. Fonte: Adaptado do manual de Virtual.PYXIS (2016). . . . .	122
Figura 75	Modelo do material utilizada no Virtual.PYXIS. Fonte: Adaptado do manual de Virtual.PYXIS (2016) . . . . .	122
Figura 76	Filtros utilizados no Virtual.PYXIS. Fonte: Adaptado do manual de Virtual.PYXIS (2016) . . . . .	123
Figura 77	Painel de Interface de OT no Virtual.PYXIS. Fonte: Adaptado do manual de Virtual.PYXIS (2016). . . . .	123
Figura 78	Sintetização com Virtual.PYXIS (2D) com V=70%, EE 3.900mm. Fonte: Do autor . . . . .	124
Figura 79	Sintetização com Virtual.PYXIS (2D) com V=70% e corte de pseudo-densidade em 0,7, EE 3.900mm. Fonte: Do autor . . . . .	124
Figura 80	CAD da Sintetização com Virtual.PYXIS (2D) com V=70% e corte de pseudo-densidade em 0,3 com condições de contorno, EE 3.900mm. Fonte: Do autor . . . . .	124
Figura 81	Simulação de tensão pós-sintetização com ParetoWorks com V=70% com resultados do Virtual.PYXIS, EE 3.900mm. Fonte: Do autor . . . . .	125
Figura 82	Simulação de deslocamentos pós-sintetização com ParetoWorks com V=70% com resultados do Virtual.PYXIS, EE 3.900mm. Fonte: Do autor . . . . .	125
Figura 83	Quadro lateral de opções do ParetoWorks. Fonte: Do autor . . . . .	126
Figura 84	Caixa de unidades. Fonte: Do autor . . . . .	127
Figura 85	Propriedades do material para aço LNE380. Fonte: Do autor . . . . .	127
Figura 86	Definição das condições de contorno do material. Fonte: Do autor . . . . .	127
Figura 87	Condições da malha. Fonte: Do autor . . . . .	128
Figura 88	Condições de restrição. Fonte: Do autor . . . . .	128
Figura 89	Condições de otimização, V=70%. Fonte: Do autor . . . . .	128
Figura 90	Sintetização com ParetoWorks com V=70% e com condições de contorno, EE 3.900mm. Fonte: Do autor . . . . .	129
Figura 91	Sintetização com ParetoWorks com V=70% , EE 3.900mm. Fonte: Do autor . . . . .	129
Figura 92	Simulação de tensão pós-sintetização com ParetoWorks com V=70%, EE 3.900mm. Fonte: Do autor . . . . .	129
Figura 93	Simulação de deslocamentos pós-sintetização com ParetoWorks com V=70%, EE 3.900mm. Fonte: Do autor . . . . .	130
Figura 94	Máxima rigidez da estrutura com sintetização 2D com ParetoWorks, EE 3.900mm. Fonte: Do autor . . . . .	130
Figura 95	Sintetização em perspectiva em 3D, com V=70%. Fonte: Do autor . . . . .	131

Figura 96	Sintetização vista frontal em 3D com condições de contorno, com $V=70\%$ . Fonte: Do autor . . . . .	131
Figura 97	Sintetização vista frontal em 3D, com $V=70\%$ . Fonte: Do autor . . . . .	131
Figura 98	Simulação de tensão pós-sintetização em 3D com ParetoWorks com $V=70\%$ , EE 3.900mm. Fonte: Do autor . . . . .	132
Figura 99	Simulação de deslocamentos pós-sintetização em 3D com ParetoWorks com $V=70\%$ , EE 3.900mm. Fonte: Do autor . . . . .	132
Figura 100	Máxima rigidez da estrutura com sintetização 3D com ParetoWorks, EE 3.900mm. Fonte: Do autor . . . . .	133
Figura 101	Sintetização vista frontal em 3D, com coeficiente de segurança $s=2$ . Fonte: Do autor . . . . .	135
Figura 102	Simulação de tensão pós-sintetização em 3D com ParetoWorks com $V=60\%$ , EE 3.900mm. Fonte: Do autor . . . . .	135
Figura 103	Simulação de deslocamentos pós-sintetização em 3D com ParetoWorks com $V=60\%$ , EE 3.900mm. Fonte: Do autor . . . . .	135
Figura 104	Sintetizações de MOT encontradas. Fonte: Do Autor. . . . .	137
Figura 105	FEA da Longarina comercial existente contendo todos os furos. Fonte: Do autor . . . . .	138
Figura 106	Sintetização da longarina comercial em 3D (ParetoWorks) com restrição de $V=70\%$ . Fonte: Do autor . . . . .	138
Figura 107	Sintetização da longarina comercial em 3D (ParetoWorks) com maximi- zação de rigidez. Fonte: Do autor . . . . .	139
Figura 108	Otimização escolhida com projeção dos furos. (ParetoWorks 3D) Fonte: Do autor . . . . .	139
Figura 109	Utilização das máscara do perfil sintetizado pra produzir nova longarina Fonte: Do autor . . . . .	140
Figura 110	Furos existentes fora dos elementos sólidos. Fonte: Do autor . . . . .	140
Figura 111	Furos alocados para dentro dos elementos sólidos. Fonte: Do autor . . . . .	141
Figura 112	Longarina do chassi final, vista frontal. Fonte: Do autor . . . . .	141
Figura 113	Longarina do chassi final, vista em perspectiva. Fonte: Do autor . . . . .	141
Figura 114	Análise FEA da longarina final. Tensão máxima. Fonte: Do autor . . . . .	142
Figura 115	Análise FEA da longarina final. Deslocamento máximo. Fonte: Do autor . . . . .	142

# Lista de tabelas

Tabela 1	Cronologia da Otimização Estrutural. Fonte: Silva (2003a), Bendsøe e Sigmund (2003), Arora (2007). . . . .	29
Tabela 2	Classes Principais de otimização estrutural. Fonte: Bendsøe e Sigmund (2003), Christensen e Klarbring (2009). . . . .	30
Tabela 3	Tipos possíveis das Classes Principais de OT. Fonte: Bendsøe e Sigmund (2003), Christensen e Klarbring (2009) . . . . .	30
Tabela 4	Características de Derivações das Classes Principais de OT. Fonte: Bendsøe e Sigmund (2003), Christensen e Klarbring (2009). . . . .	32
Tabela 5	Novas classes de OT. Fonte: (JENKINS, 2015), (SIGMUND; MAUTE, 2013). . . . .	32
Tabela 6	Alguns autores que definem o histórico da OT. Fonte: Do Autor . . . .	35
Tabela 7	Diversas aplicações de OT. Fonte: Do Autor . . . . .	36
Tabela 8	Classificação das funções de otimização. Fonte: Criada pelo autor com base em Rao (2009) . . . . .	55
Tabela 9	Descrição de Parâmetros de Entrada . . . . .	78
Tabela 10	Classificação de tipo de veículo. Fonte: Do autor com base em e-mail fornecido pela empresa Scania . . . . .	102
Tabela 11	Classificação de tipo de veículo. Fonte: Do autor com base em e-mail fornecido pela empresa Scania . . . . .	102
Tabela 12	Valores de linhas elástica do perfil bidimensional para várias cargas. Fonte: Do autor. . . . .	105
Tabela 13	Valores de cargas para o caso 2D e o 3D. Fonte: Do autor. . . . .	105
Tabela 14	Lista de dados de entrada requeridos pelos algoritmos . . . . .	108
Tabela 15	Requisitos de entrada requeridos pelo programa . . . . .	109
Tabela 16	Tempo e Número de Iterações . . . . .	112
Tabela 17	Número de Iterações e Tempo . . . . .	119
Tabela 18	Resultados numéricos da análise FEA. . . . .	134
Tabela 19	Resultados de FEA e sintetização da longarina comercial que contém todos os furos . . . . .	139
Tabela 20	Resultados de FEA e sintetização da longarina comercial que contém todos os furos . . . . .	142

# Lista de abreviaturas e siglas

MEF	Método dos Elementos Finitos
CAE	Engenharia Auxiliada por computador
FEA	Análise por Elementos Finitos
PDE	Equações diferenciais parciais
MOT	Método de otimização Topológica
SIMP	Material Sólido isotrópico com penalidade
2D	Objeto Bidimensional
3D	Objeto Tridimensional
GSM	Método da malha densa de barras
CO	Critério de Otimalidade
OP	Otimização Paramétrica de tamanho, espessura, topográfica ou topométrica
OF	Otimização de forma
OT	Otimização Topológica
OMA	Otimização Multi-Objetiva ou de Pareto
OMD	Otimização Multidisciplinar
NVH	Ruídos, Vibração e Aspereza
MOTAL	Método de Otimização por alocação de lacunas
MOTCN	Método de Otimização por curvas de nível
MCF	Método de Otimização por campo de fase
MOTE	Método de Otimização Evolucionário
MOP	Método de Otimização Paramétrica
MOF	Método de Otimização de Forma

RAMP	Propriedades do Material de aproximação racional
MLS	Método <i>Level Set</i>
ESO	Otimização Estrutural Evolucionária
BESO	Otimização Estrutural Evolucionária Bidirecional
LS	Level Set
CAMD	Aproximação contínua de distribuição do material
PLS	Programação Linear Sequencial
MMA	Método de Movimentação Assintota
PL	Programação Linear
PNL	Programação Não-Linear
PQS	Programação Quadrática Sequencial
LINCO	Linearização Convexa
DOF	Graus de Liberdade
CAD	Desenho Assistido por Computador

# Lista de símbolos

$\mathbf{X}$	Domínio do vetor de variáveis de projeto
$f(\mathbf{X})$	Função custo ou função objetivo
$g_j(\mathbf{X})$	Restrição de Desigualdade
$l_j(\mathbf{X})$	Restrição de Igualdade
$\rho_e (i = 1, \dots, M)$	variáveis do problema e pseudo-densidade em cada ponto do domínio discretizado
$c(\mathbf{X})$	função objetivo que é a flexibilidade média (rigidez estrutural), conhecido como compliance
$\mathbf{U}$	Deslocamento global
$\mathbf{K}$	matriz de rigidez global da estrutura
$\mathbf{u}_e$	campo de deslocamento gerado pela força $\mathbf{f}$
$\mathbf{k}_e$	Vetor matriz de rigidez do elemento
$p$	valor da penalização (tipicamente $p=3$ )
$V(x)$	volume de material final
$V_0$	volume de material inicial
$f(volfrac)$	é a fração de volume
$\rho_{min}$	Vetor de pseudo-densidade relativa mínima
$\mathbf{F}$	Vetor força
$N$	Número totais de elementos para discretizar o domínio $N = (\mathbf{nex} \times \mathbf{nely})$
$E(\rho)$	módulo de Young do material
$E_0$	módulo de Young do material sólido
$E_{min}$	módulo de Young mínimo para evitar singularidade
$x_e$	Vetor variável do projeto

$V(X) = V_{req}$	Restrição do volume do material desejado ao final da otimização
$\psi(\mathbf{p}, t)$	função de Level Set
$s(\mathbf{p}, t)$	posição ou forma do LS
$t$	: tempo da função de LS
$v(\mathbf{p})$	: É o campo escalar sobre o domínio projetado, na função de velocidade.
$g(\mathbf{p})$	: É o campo escalar sobre o domínio projetado e força novos furos para serem criados em $\overline{D}$ .
$\omega$	: É o parâmetro positivo que determina a influência de $g$ .
$D$	: Sólido total do LS
$\partial D$	: Contorno do domínio de LS
$\overline{D}$	: Parcela do domínio escolhida do LS
$\hat{H}_{ef}$	: É o operador de convolução (fator de peso)
$dist(e, f)$	: Operador que define a distância entre o centro do elemento $e$ e o centro do elemento $f$ .



# Sumário

<b>1</b>	<b>Introdução</b>	<b>24</b>
	<b>Introdução</b>	<b>24</b>
1.1	Problemática	25
1.2	Hipótese	26
1.3	Objetivo Geral	26
1.3.1	Objetivos Específicos	26
1.4	Organização da dissertação	26
<b>2</b>	<b>Teoria de Otimização estrutural</b>	<b>28</b>
2.1	Classes de otimização estrutural	29
2.1.1	Otimização paramétrica ou dimensional	30
2.1.1.1	Otimização de tamanho ou de espessura.	31
2.1.1.2	Otimização topográfica ou topométrica ( <i>Topography or Topometry</i> )	31
2.1.2	Otimização de forma	31
2.1.3	Otimização topológica	32
2.1.4	Novas classes de otimização estrutural	32
2.1.4.1	Otimização Multi-objetiva (OMA) ou de Pareto	33
2.1.4.2	Otimização de Robustez e Confiabilidade	33
2.1.4.3	Otimização Multidisciplinar (OMD)	33
2.1.5	Fluxograma de otimização estrutural	33
<b>3</b>	<b>Teoria de Otimização Topológica</b>	<b>35</b>
3.1	Formulação do Problema de otimização básico	36
3.2	Abordagens do MOT	37
3.2.1	Campo Baseado no Material	38
3.2.1.1	Abordagem do método da homogeneização	38
3.2.1.2	Abordagem do método de densidade (modelo SIMP)	38
3.2.1.3	Formulação do problema de otimização topológica SIMP	40
3.2.1.4	Função Objetivo na formulação	41
3.2.1.5	Restrições na formulação	41
3.2.1.6	Variáveis de projeto na formulação	42
3.2.1.7	Características SIMP	42
3.2.2	Campo Baseado na Geometria	44
3.2.2.1	Abordagem de <i>Level Set</i>	44

3.2.2.2	Abordagem de Derivadas Topológicas . . . . .	47
3.2.2.3	Abordagem de <i>Phase-Field</i> . . . . .	47
3.2.2.4	Abordagem evolucionária . . . . .	48
3.2.2.5	Comparação dos métodos de otimização topológica . . . . .	48
3.3	Problemas de instabilidade numérica na OT . . . . .	49
3.3.1	Problema do tabuleiro de xadrez . . . . .	49
3.3.2	Problema de formação de ilhas . . . . .	50
3.3.3	Problema da dependência de malhas . . . . .	51
3.3.4	Problemas mínimo local . . . . .	52
3.3.4.1	Ótimo local vs Ótimo global . . . . .	52
<b>4</b>	<b>Implantação numérica . . . . .</b>	<b>54</b>
4.1	Introdução Teórica . . . . .	54
4.1.1	Otimizadores (Métodos de Otimização) . . . . .	54
4.1.1.1	Programação por Critério de Otimalidade (CO) . . . . .	55
4.1.1.2	Programação pelo Método das Assíntotas Móveis (MMA) . . . . .	56
4.1.1.3	Programação Linear Sequencial (PLS) . . . . .	56
4.1.1.4	Programação Quadrática Sequencial (PQS) . . . . .	57
4.1.1.5	Outros otimizadores . . . . .	57
4.2	Implementação do MEF na Otimização Topológica . . . . .	58
4.2.1	Implementação do MEF para elementos 2D genérico . . . . .	58
4.2.2	Implementação do MEF para elementos 2D simplificado . . . . .	64
4.2.3	Implementação do MEF para elementos 3D genéricos . . . . .	65
4.3	Algoritmos de Otimização no MATLAB® . . . . .	67
4.3.1	Rotinas de Otimização no MATLAB® . . . . .	67
4.3.2	História dos algoritmos escritos em MATLAB® e outros . . . . .	67
4.3.3	Algoritmos bidimensionais - 2D . . . . .	67
4.3.4	Algoritmos tridimensionais - 3D . . . . .	69
4.3.5	Outros algoritmos . . . . .	69
4.4	Alguns formatos de domínio para algoritmos usuais na literatura de OT . . . . .	69
4.5	Algoritmo de Otimização Topológica baseado em SIMP - 2D . . . . .	72
4.5.1	Programa principal (Linha 1 a 36) . . . . .	73
4.5.2	Atualização das densidades baseado no CO (Linhas 37 a 48) . . . . .	74
4.5.3	Filtro de independência de malha (Linhas 49 a 60) . . . . .	75
4.5.4	Programa do MEF (Linhas 65 a 99) . . . . .	76
4.6	Algoritmo de Otimização Topológica baseado no LS - 2D . . . . .	77
4.6.1	Programa Principal - Linhas 1-48 . . . . .	78
4.6.2	Reinicialização da função de LS - Linhas 49-53 . . . . .	79
4.6.3	Atualização do domínio de projeto de LS - Linhas 54-63 . . . . .	80
4.6.4	Evolução da função de LS - Linhas 64-86 . . . . .	80

4.6.5	Implementação da análise FEA - Linhas 87-106	80
4.6.6	Matriz de rigidez dos elementos - Linhas 107-129	81
4.7	Algoritmo de Otimização Topológica 3D	82
4.8	Descrição dos Programas Computacionais	82
4.8.1	Virtual.PYXIS - Baseado no SIMP	83
4.8.2	ParetoWorks - Baseado no LS	83
<b>5</b>	<b>Objeto de análise</b>	<b>84</b>
5.1	Algumas aplicações em chassi estrutural com OT	84
5.2	Crerios de Projeto e Fabricação de Chassis	85
<b>6</b>	<b>Metodologia</b>	<b>88</b>
<b>7</b>	<b>Procedimentos Metodológicos</b>	<b>90</b>
7.1	Modo de Coleta de Dados	90
7.2	Modo de Análise dos dados coletados	90
7.3	Materiais	90
7.4	Métodos	91
7.4.1	Estudo e levantamento das dimensões da longarina do chassi comercial	91
7.4.2	Criação do arquivo CAD da longarina do chassi para posterior tratamento	93
7.4.3	Implementação dos algoritmos	94
7.4.4	Definição das condições de contorno equivalentes ao algoritmo e aos programas computacionais	101
7.5	Análise de dados	109
7.5.1	Coletados dos algoritmos	109
7.5.2	Coletados do algoritmo SIMP implementado	109
7.5.3	Investigação da região de formação de <i>hinges</i> na estrutura	114
7.5.4	Coletados do algoritmo LS implementado	116
7.5.5	Coletados dos programas computacionais	122
7.5.5.1	Programa Computacional Virtual.PYXIS	122
7.5.5.2	Coleta de dados com Programa Computacional ParetoWorks 2D	125
7.5.5.3	Coleta de Dados com Programa Computacional ParetoWorks 3D	130
7.6	Análise Comparativa dos resultados encontrados	136
7.6.1	Resultados da análise MOT	136
7.7	Proposta de uma nova topologia da longarina	137
7.7.1	Análise de longarina comercial (todos os furos existentes)	137
7.7.2	Estratégia para otimização final da longarina	139

7.7.3 Nova longarina do Chassi . . . . .	140
<b>8 Conclusões e observações . . . . .</b>	<b>144</b>
<b>Referências . . . . .</b>	<b>147</b>
<b>APÊNDICE A Algoritmo 2D SIMP de Sigmund, 2001 . . . . .</b>	<b>157</b>
<b>APÊNDICE B Algoritmo 2D SIMP Implementado . . . . .</b>	<b>159</b>
<b>APÊNDICE C Algoritmo 2D LS de Challis, 2009 . . . . .</b>	<b>161</b>
<b>APÊNDICE D Algoritmo 2D LS Implementado . . . . .</b>	<b>163</b>
<b>APÊNDICE E Exemplo de Algoritmo 3D de Liu e Tovar, 2013 . . . . .</b>	<b>165</b>

# 1 Introdução

A maioria das empresas, fabricantes de chassi veiculares, tentam diminuir o peso de suas estruturas, porém os projetistas encontram dificuldades para chegar na melhor diminuição deste peso. A constante dificuldade dos engenheiros de projetos que dimensionam carrocerias em chassis e se deparam com o excesso de peso nos eixos dos veículos é considerável. A redução do peso gera como resultado final, diversos fatores positivos, como diminuição do consumo do combustível do veículo, possibilidade de transportar maiores volumes de carga para carrocerias, diminuição de peso de componentes dos eixos dianteiros e traseiros, possibilidade de diminuir a rodagem que são os índices de cargas dos pneus, possibilitando melhores custos de compra com a substituição destes. Como a longarina do chassi veicular é a ligação entre a cabine do veículo, motorização, eixos de transmissão e rodagem, este componente tem um grau de importância muito grande em um veículo automotor. A presente proposta de trabalho tem como problema, resolver a dúvida destes projetistas em determinar a melhor posição dos furos na longarina por meio de uma nova topologia, sem comprometer a funcionalidade da mesma. Assim esta melhoria na disposição dos furos e o novo formato da longarina devido a obtenção de resultados pela otimização facilitará a escolha e reduzirá custos de projeto, sejam eles com retrabalhos de engenharia, re-fabricação de novos moldes para as longarinas etc.

Existem técnicas da área de engenharia para auxiliar o projetista no desenvolvimento da longarina e outros problemas típicos de engenharia como a simulação baseada no método dos elementos finitos (MEF) denominado de Engenharia Auxiliada por Computador (CAE) que em inglês se escreve como *Computer-Aided Engineering*. O MEF que por muitas vezes é também referido como Análise de Elementos Finitos (FEA) ou em inglês *Finite Element Analysis* é uma técnica numérica para encontrar soluções aproximadas de equações diferenciais parciais (PDE) que dependem de sua derivada, bem como de equações integrais (ALVES FILHO, 2000), no entanto, um outro método, especificamente o Método de Otimização Topológica (MOT) que será discutido, é um assunto relativamente recente e moderno, havendo necessidade de pesquisas para o desenvolvimento e para a introdução deste método no departamento de engenharia das empresas fabricantes de veículo nacionais. O MOT é um método eficiente, mas tem problemas de instabilidade numérica, por meio de dependência de malha, problemas de mínimo local, e com surgimento do tabuleiro de xadrez (regiões no domínio da estrutura em que se tem elementos com presença ou ausência do material, distribuído de forma intercalada). Este método pode ser abordado de algumas formas como veremos nos capítulos posteriores, como com base no material

por meio do método de homogeneização ou pelo método SIMP, e com base na geometria por meio do método *Level Set* e outros ou abordagens heurísticas. O presente estudo visa demonstrar a aplicação de programas computacionais não somente de sistemas CAE, mas também de otimização topológica para a análise dos componentes e fabricação de chassi e também de algoritmos com base nos estudos de Sigmund (2001) e Challis e Guest (2009) que serão alterados para utilização na aplicação do chassi. Sendo assim, com bases em trabalhos recentes a serem discutidos nos capítulos posteriores, verifica-se que a otimização topológica é um dos métodos promissores para síntese de projetos, por isto utilizar-se-á este método para o estudo em questão. O estudo no presente trabalho de uma longarina de chassi comercial de um veículo automotor específico, será demonstrado visando a síntese da otimização topológica aplicado ao produto. Este método pode futuramente ser aplicado em outras partes do veículo como cabine, motorização e propulsão, carroceria etc.

Será discutido o método de otimização topológica para esta aplicação, mediante análise de resultados entre os programas computacionais ParetoWorks <sup>1</sup>, ao qual a eficiência do método com base na abordagem de *Level Set* se dá em Challis e Guest (2009) e em Suresh (2010) pela análise da teoria de Pareto com sensibilidade topológica que é aperfeiçoada para problemas multi-objetivos, e também será demonstrado o Virtual.PYXIS <sup>2</sup> que utiliza a abordagem SIMP com o critério de otimalidade conforme vemos em Sigmund (2001). Além disto, procuraremos verificar a base da criação de algoritmos de otimização topológica feito no MATLAB® <sup>3</sup> por meio de referências de outras aplicações como vemos em Sigmund (2001) e Challis e Guest (2009). Toda a avaliação será feita nestes módulos e algoritmos, e também será possível analisar os resultados do sistema CAE nos programas computacionais depois da otimização, sendo possível demonstrar a eficiência do método de otimização topológica para a aplicação de topologia de longarina de chassi veicular.

## 1.1 Problemática

Os projetistas de veículos podem desenvolver uma longarina de chassis estrutural com furos nas posições corretas, que dure o tempo de vida projetado para aquele produto nas diversas condições de cargas, sem que haja trincas provindas das furações ao longo deste chassi. As longarinas de chassi são sempre projetadas levando em consideração todos os esforços que serão submetidas por meio das diversas cargas que os veículos (carro, caminhão e ônibus) podem receber.

<sup>1</sup> ParetoWorks: Programa computacional criado e comercializado em 2015 pela empresa Sciart que usa a base na abordagem LS. Fonte: <https://www.sciartsoft.com/paretoworks/>

<sup>2</sup> Virtual.PYXIS: Programa computacional criado e comercializado em 2015 pela empresa nacional Virtual CAE que usa a base na abordagem SIMP com CO, MMA ou PLS. Fonte: <http://virtualpyxix.com.br/>

<sup>3</sup> MATLAB: Ferramenta para desenvolver algoritmos de linguagem matemática comercializado pela empresa MathWorks. Fonte: <http://www.mathworks.com/>

## 1.2 Hipótese

Será que é possível fazer uma nova topologia de uma longarina de um chassi veicular com o uso da técnica de otimização topológica?

## 1.3 Objetivo Geral

Desenvolver uma nova topologia para a longarina de veículo (caminhão) por meio da técnica de otimização topológica estrutural.

### 1.3.1 Objetivos Específicos

- Simular a técnica de otimização topológica 2D para identificar os recortes vazios e os elementos sólidos em que poderá ter furos na longarina por meio de dois programas computacionais: Virtual.PYXIS (Virtual CAE) e ParetoWorks (Sciart).
- Selecionar e propor alterações para implementação de um algoritmo 2D com base na abordagem SIMP que procure demonstrar uma longarina de chassi sintetizada.
- Selecionar e estudar um algoritmo 2D com base na abordagem *Level Set* procurando identificar as possíveis alterações e diferenças frente a abordagem SIMP.
- Sintetizar a longarina de chassi em 3D com um programa computacional para avaliar como a aba do perfil C influencia no resultado da sintetização de otimização topológica.
- Comparar as simulações nos programas computacionais e no algoritmo para validar o modelo de geração de uma nova topologia da longarina pela técnica de otimização topológica.
- Escolher a melhor sintetização dentre os algoritmos e os programas computacionais e simular por elementos finitos se os deslocamentos e as tensões máximas estão aceitos para a nova longarina de chassi encontrada.

## 1.4 Organização da dissertação

No capítulo 2 temos uma revisão da teoria da otimização estrutural, em que tem-se o contexto histórico de quase 1 século e meio de trabalhos com o tema. É descrito os estudos de otimização estrutural, e as características que cada estudo contemplou, além de demonstrar as referências principais de autores que escreveram sobre o tema. Neste capítulo é demonstrado as diferentes classes de otimização estrutural, especificando a definição das classes clássicas e das classes modernas. Por fim demonstra-se uma figura contemplando todas as classes de otimização estrutural para uma fácil interpretação do estado da arte de otimização estrutural.

No capítulo 3 é apresentado a teoria de otimização topológica, em que historicamente começa-se a descrever os primeiros estudos sobre o tema por volta de 3 séculos. Definem-se quais foram as principais aplicações de otimização topológica desenvolvidas. Demonstra-se a formulação do problema básico de otimização, com a função objetiva, as variáveis de projeto e suas restrições. Por fim demonstra-se os principais métodos de abordagem com esta temática até a atualidade como a abordagem da homogeneização, SIMP, *Level Set* e heurísticas. Por fim conclui-se o capítulo demonstrando os problemas de instabilidade numérica encontrados nas abordagens.

No capítulo 4 é demonstrado a implantação numérica para o problema de otimização topológica. Destacam-se os otimizadores, e como é a forma geral de implantação do MEF na otimização topológica. Além disto discute-se os principais algoritmos de otimização existentes, e os domínios encontrados na literatura. Apresenta-se a descrição das linhas de comando dos algoritmos com abordagem em SIMP e abordagem *Level Set* e é apresentado a base dos programas Virtual.PYXIS e ParetoWorks.

No capítulo 5 discute-se sobre o objeto de análise, que é uma longarina de um chassi veicular, demonstra-se algumas aplicações em chassi estrutural com otimização topológica e os critérios de projeto e fabricação de chassis.

No capítulo 6 apresenta-se a metodologia do trabalho e a proposta do trabalho em questão. Nesta metodologia discute-se qual a técnica de pesquisa utilizada e quais foram as áreas abrangentes para o estudo de aplicação da longarina do chassi veicular.

No capítulo 7 define-se os procedimentos metodológicos, como a coleta e o tratamento dos dados. Neste capítulo descreve-se como foram implementados os algoritmos, e as características da longarina do chassi comercial. Além disto descreve-se como determinar o número de elementos do objeto. Demonstra-se as análises com os algoritmos para os cinco casos de um chassi comercial contemplando diversas medidas de entre-eixos, sendo sintetizados com as abordagens SIMP e de *Level Set*, além dos programas computacionais. Destes verifica-se a longarina sintetizada mais próximo da realidade, e propõe-se um estudo para chegarmos em uma nova topologia de uma longarina de um chassi veicular com base na furação de uma longarina de um chassi comercial.

No capítulo 8 discute-se a conclusão dos resultados, e demonstra-se a eficiência dos algoritmos e dos programas computacionais para o estudo de aplicação da longarina do chassi veicular. Procura-se responder todos os objetivos específicos e tem-se a recomendação para trabalhos futuros, já que este é um tema atual e cabível de ser explorado por outros pesquisadores.

Na sequência apresenta-se as referências bibliográficas e os apêndices utilizados que no caso foram os algoritmos educacionais existentes e os algoritmos com implementações de condições de contorno etc.



## 2 Teoria de Otimização estrutural

[Silva \(2003a\)](#) informa que o primeiro estudo sobre otimização estrutural é mais antigo do que se pensa, e data de 1872 por Maxwell, com a aplicação em pontes para estruturas civis, em que procurou-se reduzir a massa para estruturas uniaxiais submetidas a carregamentos por meio de conceitos da teoria da elasticidade. No entanto [Arora \(2007\)](#), cita que os primeiros estudos referente a otimização originaram-se pelo artigo de [Michell \(1904\)](#) que estava preocupado com o projeto de estruturas "finas" e com limite de plasticidade. A preocupação consistia no dimensionamento estático de treliças com condições específicas de contorno e carregamento, assim baseou-se no método analítico. [Michell \(1904\)](#) deu continuidade aos trabalhos de Maxwell (1872) obtendo resultados analíticos para alguns problemas de otimização, segundo [Silva \(2003a\)](#), a otimização estrutural consistia inicialmente calcular o campo de tensões máximas principais, usando a teoria da elasticidade, com uma força aplicada num ponto de um domínio infinito que está sujeito as restrições de deslocamento em outros pontos conforme figura 1.

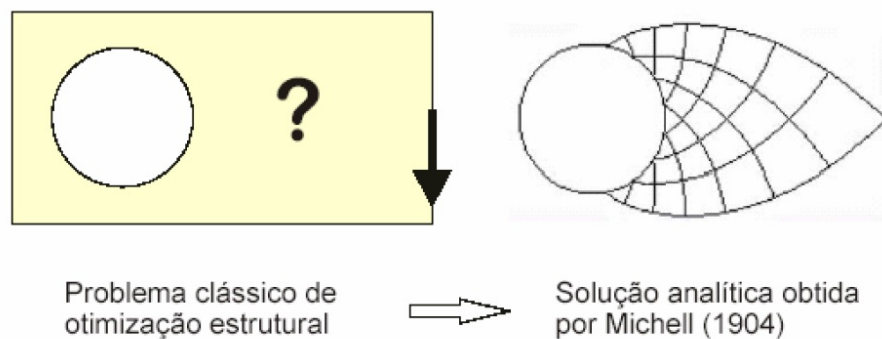


Figura 1 – Otimização estrutural obtida por [Michell \(1904\)](#). Fonte: ([RAMOS, 2007](#))

Até o início da década de 1960 não houve estudos práticos de otimização, eram apenas estudados problemas simples (vigas, treliças), mas a partir disto, e com o advento da computação e do MEF, técnicas foram empregadas para o projeto de redução de peso com restrição de tensão em estruturas de treliças de forma analítica. Verifica-se isto em [Victoria, Martí e Querin \(2009\)](#), em que a otimização estrutural partiu-se quando introduziu-se o título Método da malha densa de barras (GSM) do inglês *ground structure approach* desenvolvido por [Dorn \(1964\)](#) ao qual baseava-se na otimização topológica de estruturas e que tem-se uma matriz de nós que estão interligados com uma malha densa de membros lineares conhecida como estrutura de barras. [Hemp \(1973\)](#) e [Prager \(1974\)](#), ambos ainda para uma classe restrita de estruturas são vistos como uma extensão dos

trabalhos de [Michell \(1904\)](#) além de [Rozvany e Zhou \(1991\)](#) e [Bendsøe, Ben-Tal e Zowe \(1994\)](#) que recentemente continuaram o trabalho iniciado por [Michell \(1904\)](#). Assim na década de 1960 até 1990 tem-se os chamados técnicas de minimização com Métodos Diretos de Programação Matemática, após esta década é apresentado os métodos indiretos em que tem-se o Critério de otimalidade (CO) etc métodos.

Os livros publicados por [Haftka e Gürdal \(1992\)](#), [Bendsøe e Sigmund \(2003\)](#), [Rao \(2009\)](#) e [Christensen e Klarbring \(2009\)](#), nos deram um grande base para a teoria da otimização estrutural. Na tabela 1 podemos verificar um resumo de todos os autores que trabalharam com a teoria de otimização estrutural ao longo da história.

Tabela 1 – Cronologia da Otimização Estrutural. Fonte: [Silva \(2003a\)](#), [Bendsøe e Sigmund \(2003\)](#), [Arora \(2007\)](#).

Autor (Ano)	Característica	Método Otim.
Maxwell (1872)	Primeiro estudo de otimização	Analítico
<a href="#">Michell (1904)</a>	Topologia de treliças	Analítico
<a href="#">Dorn (1964)</a>	<i>Ground structure approach</i>	Prog. Matemática
<a href="#">Hemp (1973)</a> , <a href="#">Prager (1974)</a> , <a href="#">Rozvany e Zhou (1991)</a> , <a href="#">Bendsøe, Ben-Tal e Zowe (1994)</a>	Revisões de <a href="#">Michell (1904)</a>	Analítico

Ainda segundo [Silva \(2003a\)](#), na década de 1960 implementou-se a otimização paramétrica, utilizando a programação linear com base no método Simplex. Na década de 1970 outros algoritmos de problemas não-lineares são apresentados e são utilizados até hoje. Nesta década são implementados a otimização de forma e a abordagem probabilística com os algoritmos genéticos. Na década de 1980 aparecem os primeiros programas computacionais comerciais, e inicia-se o desenvolvimento da otimização topológica na área acadêmica. Sendo assim existem uma divisão de classes de otimização estrutural.

## 2.1 Classes de otimização estrutural

Segundo [Silva \(2003a\)](#), [Bendsøe e Sigmund \(2003\)](#) e [Christensen e Klarbring \(2009\)](#), a otimização estrutural é na forma clássica, classificada em 3 classes: otimização paramétrica (dimensional) (a), de forma (geométrica) (b), e topológica (c), conforme figura 2.

A tabela 2 apresenta as classes principais de otimização estrutural e a principal característica que a compõe, e também o tipo de modificação apresentado em sua estrutura. Observa-se que há uma ordem crescente de possibilidade de redução de volumes pelos ditos "novos buracos" em relação a estas 3 classes, assim a otimização topológica é atualmente o tipo que mais reduz material de uma estrutura, além de apresentar uma forma livre de obtenção dos "novos buracos" em função das condições de contorno (cargas e fixações) impostas ao domínio discretizado.

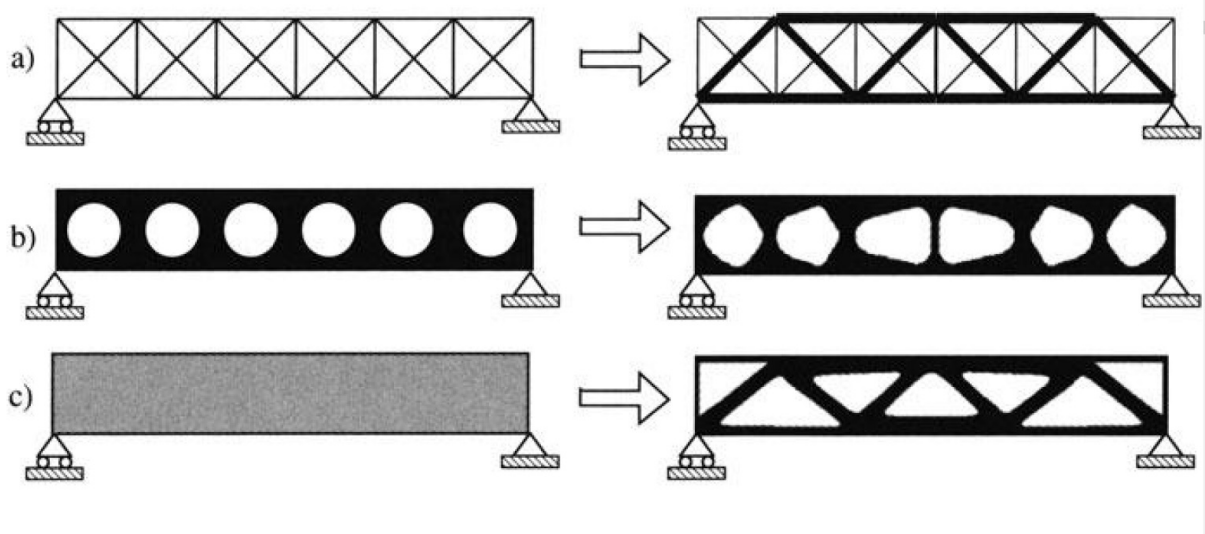


Figura 2 – Classes de otimização estrutural. Fonte: (BENDSØE; SIGMUND, 2003).

Tabela 2 – Classes Principais de otimização estrutural. Fonte: Bendsøe e Sigmund (2003), Christensen e Klarbring (2009).

Classes Principais	Característica	Modificação
Otimização Paramétrica (sizing)	Definição de L,b,h da viga (a)	Dimensão
Otimização de Forma (shape)	Trabalha nos "nós" ou "elementos" (b)	Geometria
Otimização Topológica (topology)	Forma livre (c)	Topologia

Cada uma dessas classes principais ao longo do histórico de otimização estrutural foram derivadas para outras sub-classes, desta forma as derivações das principais classes de otimização estrutural são apresentadas na tabela 3.

Tabela 3 – Tipos possíveis das Classes Principais de OT. Fonte: Bendsøe e Sigmund (2003), Christensen e Klarbring (2009)

Classes Principais	Tipo 1	Tipo 2
Otimização Paramétrica	De Tamanho (Size)	Espessura (Free-size)
Otimização Paramétrica	Topográfica (Topography)	Topométrica (Topometry)
Otimização Forma	Elemento (shape)	Nós (free-shape)
Otimização Topológica	Material (Homogeneização ou SIMP)	Geometria (LS etc.)

Por meio desta divisão, alguns autores como Haftka e Gürdal (1992), Vanderplaats (1999) e Christensen e Klarbring (2009) classificam todas as classes principais e tipos de acordo com os próximos tópicos.

### 2.1.1 Otimização paramétrica ou dimensional

Na otimização paramétrica ou dimensional (OP) do inglês *Sizing Optimization*, temos a definição prévia do formato da estrutura, e nas variáveis de projeto temos fixado

as dimensões (VANDERPLAATS, 1999). A OP é utilizada para resolver problemas em que a minimização de uma função objetiva é especificada por meio de um conjunto de dados definidos, por exemplo a espessura de uma placa ou as áreas das seções transversais das barras de uma treliça (CHRISTENSEN; KLARBRING, 2009). Outras formulação baseadas no critério de otimalidade são vistas em Rozvany (1997). Normalmente é umas das otimizações mais conservadoras, pois é predefinido os valores das dimensões do perfil e do comprimento, ou seja a uma forma inicial do objeto a ser sintetizado que não é modificada, e após a sintetização é alterado apenas a variável livre. Na figura 2(a), temos as variáveis representadas como as áreas transversais das barras da treliça.

#### 2.1.1.1 Otimização de tamanho ou de espessura.

Derivado da OP, a otimização de tamanho (*Size*) que também pode ser conhecida como otimização de espessura (*Free-size*) em alguns casos, é muito usada para materiais plásticos, ou compósitos, em que o controle da espessura do objeto é uma premissa importante. Segundo Jenkins (2015) a otimização de tamanho consiste em modificar propriedades relacionadas ao tamanho dos elementos como espessura de placas, seção transversal de uma viga, diâmetro de uma mola etc.

#### 2.1.1.2 Otimização topográfica ou topométrica (*Topography or Topometry*)

Recentemente, surgiu uma técnica de otimização conhecida como "Otimização Topográfica", que é uma classe especial da OP, e que é utilizada na indústria automobilística em componentes estampados, melhorando rigidez em determinados relevos da estrutura, focada em mudar a superfície do componente. Segundo Jenkins (2015) a otimização topografia e topométrica podem ser aplicadas para elementos 2D ou elementos de casca. A otimização topográfica é uma variação da otimização dimensional em que cada elemento da estrutura pode conter sua própria variável de projeto e é semelhante a otimização topométrica pois tem as variáveis restritas na espessura, a otimização topométrica procura encontrar um padrão de camadas ideal ao componente.

### 2.1.2 Otimização de forma

A otimização de forma (OF) que do inglês é *Shape Optimization* (HAFTKA; GÜRDAL, 1992), é aquela em que a forma é modificada com controles. Ela trabalha nos nós do componente da malha (*free-shape*) ou no próprio elemento (*shape*) contendo toda a região da malha, assim é possível ter um controle destas malhas em relação ao formato dos elementos e posicionamento dos nós facilitando a diminuição das tensões dos contornos. A *shape* tem mais controle de malha do que a *free-shape* (BENDSØE, 1995). No exemplo da figura 2(b), as variáveis são as coordenadas dos pontos de controle das *splines* que dão forma aos vazios da viga. A otimização de forma é bastante complicada no

caso de estrutura tridimensional, pois precisa atualizar a malha a cada iteração. [Jenkins \(2015\)](#) demonstra que a OF serve para ajustar as superfícies de uma placa 2D ou de um sólido 3D para minimizar o volume satisfazendo as restrições de tensão ou deslocamento impostas.

### 2.1.3 Otimização topológica

A otimização topológica (OT), do inglês *Topology optimization*, é um método computacional que permite projetar a topologia ótima retirando material do domínio da estrutura. Na figura 2(c) temos o método de otimização topológica trabalhando com variação da topologia da estrutura apresentando uma forma final livre sem que tenha-se que determinar as dimensões como vemos na otimização paramétrica. Por meio da teoria apresentada, na tabela 4 é possível verificar portanto, um resumo das características de cada uma dos tipos das classes principais de otimização estrutural.

Tabela 4 – Características de Derivações das Classes Principais de OT. Fonte: [Bendsøe e Sigmund \(2003\)](#), [Christensen e Klarbring \(2009\)](#).

Derivações	Característica
OP Tamanho (Size)	Usada para materiais plásticos para controlar espessura
OP Espessura (Free-Size)	Idem a OP Tamanho
OP Topográfica (Topography)	Utilizada em componentes estampados
OP Topométrica (Topometry)	Podem ser aplicadas para elementos de casca (2D)
OF (shape e free-shape)	Trabalha no elemento ou nos nós para modificar a malha
OT Baseado Material	Discretização do Domínio
OT Baseado Geometria	Discretização do Contorno do Domínio

### 2.1.4 Novas classes de otimização estrutural

Ao longo das duas últimas décadas desenvolveu-se novas classes, como por exemplo a otimização multi-objetiva ou de pareto, de robustez e confiabilidade e a multidisciplinar observados em [Jenkins \(2015\)](#) e [Sigmund e Maute \(2013\)](#). Assim, atualmente a otimização estrutural é dividida em mais classes que são exemplificadas de uma forma geral pelo autor com base nestes pesquisadores como na figura 2 e também apresentam características de acordo com a tabela 5.

Tabela 5 – Novas classes de OT. Fonte: ([JENKINS, 2015](#)), ([SIGMUND; MAUTE, 2013](#)).

Novas Classes	Característica
Otimização Multi-objetiva ou Pareto	Decisão por seres humanos
Otimização de Robustez e Confiabilidade	Métodos por controle estatístico
Otimização Multidisciplinar	Envolve o estudo de várias disciplinas

#### 2.1.4.1 Otimização Multi-objetiva (OMA) ou de Pareto

Segundo [Jenkins \(2015\)](#) nesta classe temos a premissa de que a otimização do mundo real é resolvida não somente com um objetivo, mas com muitos outros. Um exemplo é uma estrutura leve e rígida, um objetivo se sobressai em relação ao outro sempre trocando de posições, isto é conhecido como Pareto e este tipo de otimização numericamente define as fronteiras de Pareto com base em determinar soluções por fatores exógenos, com decisão por humanos e não por modelo de computação somente.

#### 2.1.4.2 Otimização de Robustez e Confiabilidade

São métodos para controle de medidas em termos de principal valor e variância. Segundo [Jenkins \(2015\)](#) é demonstrado que os projetos de produtos são nominais, no entanto as condições de fabricação e operação são reais, aparecendo as tolerâncias geométricas, as variações dos materiais, as incertezas, assim é possível utilizar esta classe de otimização frente a outros métodos de otimização para poder corrigir estes erros do meio ambiente.

#### 2.1.4.3 Otimização Multidisciplinar (OMD)

Segundo [Jenkins \(2015\)](#) a OMD envolve todas os campos de aplicação relevantes: Estrutural (linear ou não-linear, estático ou dinâmico, materiais particulados ou compósitos), fluidos, térmico, acústico, dinâmica de multi-corpos, crash-testes, NVH ( que do inglês significa *noise, vibration and harshness*), ou seja, ruídos, vibração e aspereza. Esta classe vem se tornando bem utilizada nos últimos anos, pois com o envolvimento de todos os campos há uma redução em custos de processos de otimização etc.

### 2.1.5 Fluxograma de otimização estrutural

Com base nos estudos das classes clássicas e das novas classes de otimização estrutural, é demonstrado o fluxograma da figura 3.

No fluxograma já destacamos as siglas mais utilizadas nas abordagens em relação aos modelos baseados na geometria e também heurísticos, conforme observado em [Amigo \(2013\)](#) aos quais temos a otimização por alocação de lacunas (MOTAL), por curvas de nível (MOTCN), por campo de fase (MCF) e evolucionária (MOTE). Os modelos de Otimização Topológica baseados no material e na geometria serão discutidos em detalhes no próximo capítulo, pois cabe o entendimento detalhado de todas as abordagens de OT para consequentemente trabalhar com os dados do problema proposto de avaliar uma topologia ótima para um chassi estrutural.



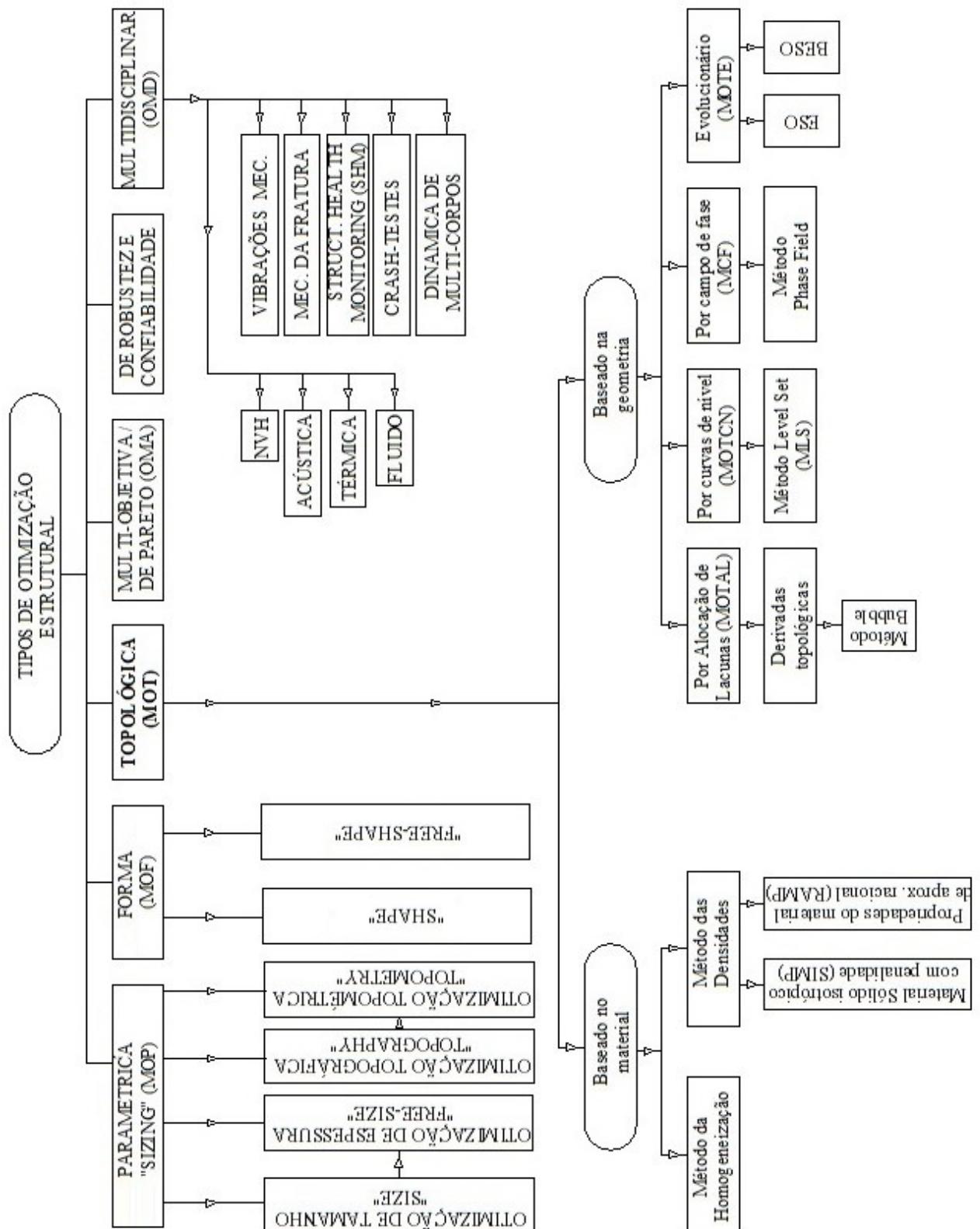


Figura 3 – Resumo dos tipos atuais de otimização estrutural. Fonte: Do autor com base em Jenkins (2015), Amigo (2013) Sigmund e Maute (2013).

### 3 Teoria de Otimização Topológica

[Bendsøe e Kikuchi \(1988\)](#) iniciou o conceito de otimização topológica (OT) baseado em estudos citados no capítulo 2 referente a otimização estrutural e também em publicações da década de 1980 em que começava-se a estudar problemas de otimização para a minimização do material (com restrição de volume) como vemos em [Rozvany et al. \(1982\)](#). O conceito de OT inicialmente baseou-se na abordagem de homogeneização. O começo da otimização topológica deu-se principalmente na área de aeronáutica, em que a redução do peso do material foi exaustivamente estudada para minimizar os custos de fabricação. Buscou-se desenvolver este tema, pois a OF era muito limitado e necessitava-se de um desenvolvimento para reduzir mais "buracos" em uma estrutura sem que a mesma fosse prejudicada em relação as suas condições físicas. Na tabela 6 podemos verificar alguns autores que definem o histórico da OT.

Tabela 6 – Alguns autores que definem o histórico da OT. Fonte: Do Autor

Autor (ano)	Autor (ano)
<a href="#">Sigmund (1994)</a>	<a href="#">Rao (2009)</a>
<a href="#">Bendsøe (1995)</a>	<a href="#">Sigmund e Maute (2013)</a>
<a href="#">Rozvany (1997)</a>	<a href="#">Liu e Tovar (2013)</a>
<a href="#">Bendsøe e Sigmund (2003)</a>	<a href="#">Christensen e Klarbring (2009)</a>

Segundo [Sigmund e Maute \(2013\)](#), a OT nos fornece a resposta para uma questão fundamental da engenharia: Como colocar ou retirar o material dentro de um domínio de projeto afim de obter a melhor performance estrutural? [Liu e Tovar \(2013\)](#) descrevem que a OT é um método computacional de distribuição de material para sintetização de estruturas sem formas pré-definidas, diferente ao que vemos na OF e na OP. Esta operação de sintetização de estrutura é denominada método de otimização topológica (MOT). O termo Otimização Topológica definido como MOT é descrito em alguns livros, dissertações e teses como em [Sigmund \(1994\)](#), [Bendsøe \(1995\)](#), [Rozvany \(1997\)](#), [Bendsøe e Sigmund \(2003\)](#), [Rao \(2009\)](#), [Christensen e Klarbring \(2009\)](#) conforme já vimos na tabela 6. As aplicações iniciais do MOT basearam-se apenas nos problemas de projetos mecânicos, mas atualmente outras disciplinas são estudadas com o conceito como fluidos, acústica, eletromagnetismo, óptica e a combinação entre estas, as quais são observadas na figura 3.

[Ahmed \(2013\)](#) explica que a otimização topológica é a junção de uma variedade de diferentes campos como matemática, mecânica, multi-física e ciência da computação, e que tem já importantes aplicações práticas na indústria de manufatura (automotiva, aeronáutica, aeroespacial, construção civil, têxtil, equipamentos e embalagem) e também



nas atuais micro e nano-tecnológicas. Assim, nas últimas 3 décadas com o advento da OT, diversos estudos envolvendo o MOT foram publicados, as aplicações baseiam-se em frequência de ressonância em estruturas contínuas (DíAAZ; KIKUCHI, 1992), problemas de maximização da condutividade térmica na transferência de calor (PARK, 1995), projetos de mecanismos flexíveis (SIGMUND, 1997) e (NISHIWAKI et al., 1998), obtenção de imagens de tomografia de órgãos do corpo humano (LIMA, 2006) etc. Assim, na tabela 7 podemos verificar alguns destes autores que aplicaram em diversos campos com o uso de OT.

Tabela 7 – Diversas aplicações de OT. Fonte: Do Autor

Autor (ano)	Aplicação
(DíAAZ; KIKUCHI, 1992)	Frequência de ressonância em estruturas contínuas
(PARK, 1995)	Maximização da condutividade térmica na transf. de calor
(SIGMUND, 1997), (NISHIWAKI et al., 1998)	Projetos de mecanismos flexíveis
(LIMA, 2006)	Obtenção de imagens de tomografia de órgãos

Estaremos a partir daqui definindo o conceito das abordagens e formulações matemáticas de otimização.

### 3.1 Formulação do Problema de otimização básico

Rao (2009) demonstra um estado básico de um problema de otimização ou de programação matemática que além da função objetiva da equação 3.1, envolve restrições de igualdade ou desigualdade conforme equação 3.2.

$$\text{Encontrar } \mathbf{X} = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix} \text{ que } \quad \text{Min}_{(\mathbf{X})} f(\mathbf{X}) \quad (\text{Função Objetivo}) \quad (3.1)$$

Sujeito as restrições:

$$\begin{aligned} g_j(\mathbf{X}) &\leq 0, & j &= 1, 2, \dots, m & (\text{Restrição de Desigualdade}) \\ l_j(\mathbf{X}) &= 0, & j &= 1, 2, \dots, p & (\text{Restrição de Igualdade}) \end{aligned} \quad (3.2)$$

Em que:

$\mathbf{X}$  : É o vetor das variáveis de projeto, ou seja o conjunto de variáveis  $\mathbf{x} = \{x_1, x_2, \dots, x_n\}$  que caracterizam a estrutura que pode ser largura, altura, comprimento

ou até mesmo a pseudo-densidade do material que veremos em otimização topológica baseada no SIMP.

$f(\mathbf{X})$  : É a função objetivo, ou o que se quer resolver, pode ser a minimização de uma flexibilidade média (conhecido como compliance) que maximiza a rigidez estrutural, minimização de massa, minimização de frequência natural etc.

$g_j(\mathbf{X}) \leq 0$ : São as restrições de desigualdade, em que pode ser considerado uma restrição de volume etc.

$l_j(\mathbf{X}) = 0$ : São as restrições de igualdade.

$m$  e  $p$ : São as quantidades de restrições de desigualdade e igualdade respectivamente.

## 3.2 Abordagens do MOT

No artigo de [Sigmund e Maute \(2013\)](#) temos uma eficiente revisão da evolução das abordagens de otimização topológica. Este autor demonstra toda a evolução das abordagens, começando com o problema da discretização de variáveis quantitativas do tipo contínuas <sup>1</sup>, por meio do método de homogeneização ([BENDSØE; KIKUCHI, 1988](#)) e após o modelo "Material Sólido Isotrópico com Penalização" ou (SIMP), que também é conhecido como o método das densidades criados por [Bendsøe \(1989\)](#) e [Zhou e Rozvany \(1991\)](#) que baseia-se em utilizar as pseudo-densidades como variáveis de projeto. Em estudos subsequentes descreve-se as variáveis discretas <sup>2</sup> como novas abordagens como por exemplo as derivadas topológicas ([SOKOŁOWSKI; ZOCHOWSKI, 1999](#)), abordagem de *Level set* ([ALLAIRE; JOUVE; TOADER, 2002](#)), abordagem de *phase field* ([BOURDIN; CHAMBOLLE, 2003](#)) e as evolucionárias ([XIE; STEVEN, 1993](#)).

Todo o desenvolvimento inicial com [Bendsøe e Kikuchi \(1988\)](#) baseou-se nas solução de variável contínua, verificou-se que após o estudo de várias aplicações que diferentemente das variáveis discretas, as contínuas são modificadas para convergir em direção a variáveis discretas, com domínio com o uso de filtros e de penalidades, e também é uma predestinadora a ter problemas de instabilidades numéricas denominados "tabuleiro do xadrez" ou "escala de cinza". Assim os desenvolvimentos atuais baseiam-se principalmente no estudo das abordagens com variáveis discretas em meios contínuos ou simplesmente discretas como as abordagens de geometria e heurísticas evolucionárias. Esta transformação por meio dos métodos faz com que tenha uma melhoria na eficiência dos algoritmos de otimização. Existe uma ordem em função do domínio a ser discretizado. Esta ordem é

<sup>1</sup> Variável contínua: Um número infinito de valores entre quaisquer dois valores. Uma variável contínua pode ser numérica ou data/hora. Por exemplo, o comprimento de uma peça ou a data e hora do recebimento de um pagamento.

<sup>2</sup> Variável discreta: Um número finito de valores entre quaisquer dois valores. Uma variável discreta é sempre numérica. Por exemplo, o número de reclamações de clientes ou o número de falhas ou defeitos.

dividida em função ou das propriedades do material ou das condições de contorno da geometria do domínio.

Assim [Amigo \(2013\)](#) define de uma forma objetiva as diversas abordagens em função de dois campos, o baseado no material e o baseado na geometria, ver figura 3. Estes campos são caracterizados de acordo com:

1. Baseada no material - Esta abordagem combina materiais na célula unitária do domínio (método de homogeneização de [Bendsøe e Kikuchi \(1988\)](#)), ou trabalha com a existência de materiais intermediários (método da densidade ([BENDSØE, 1989](#)) e ([ZHOU; ROZVANY, 1991](#))).
2. Baseada na geometria - Esta abordagem possui três métodos principais: O Método de Otimização Topológica por Alocação de Lacunas (MOTAL) em que concentra-se a abordagem das derivadas topológicas, o Método de Otimização Topológica por curvas de nível (MOTCN) em que tem-se a abordagem do método de *Level Set* (MLS) e o Método de Otimização Topológica Evolucionário (MOTE), em que tem-se a Otimização Estrutural Evolucionária, que do inglês é conhecida como *Evolutionary Structural Optimization* com a sigla (ESO).

### 3.2.1 Campo Baseado no Material

#### 3.2.1.1 Abordagem do método da homogeneização

Esta abordagem desenvolveu-se pelo trabalho de [Bendsøe e Kikuchi \(1988\)](#), em alternativa a OF que buscou apresentar pela primeira vez métodos numéricos para análise estrutural com o uso dos resultados da malha de FEA, deixando de ser apenas analítico de acordo com os trabalhos de [Michell \(1904\)](#), [Dorn \(1964\)](#), [Hemp \(1973\)](#) e [Prager \(1974\)](#), e que tem como princípio fixar o domínio inicial da estrutura sem eliminar os elementos do MEF utilizados, mas sim alterá-los em outros tamanhos. Em [Hassani e Hinton \(1999\)](#) temos uma versão mais recente dos desenvolvimentos iniciados em 1988 com a teoria da homogeneização. Alguns trabalhos posteriores foram desenvolvidos com base na abordagem da homogeneização, são eles: [Thomsen \(1992\)](#), [Díaz e Kikuchi \(1992\)](#), [Hassani e Hinton \(1998\)](#) e [Fujii, Chen e Kikuchi \(2001\)](#).

#### 3.2.1.2 Abordagem do método de densidade (modelo SIMP)

Esta abordagem desenvolveu-se com uma alteração frente a abordagem de homogeneização. Foram [Bendsøe \(1989\)](#) e [Zhou e Rozvany \(1991\)](#) que iniciaram a formulação deste método baseado nas densidades, aos quais tiveram contribuições como em [Mlejnek \(1992\)](#). O objetivo deste método é o de encontrar uma melhor distribuição do domínio de um dado material que minimizará a função objetivo, que pode estar sujeito a restri-

ção de volume ou outras restrições. Esta distribuição do material é fixada por meio da variável de pseudo-densidade  $\rho$  que pode ser 0 (vazio) ou 1 (sólido) para qualquer ponto do domínio. Neste método temos a discretização do domínio fixada, que é denominada estendida, que é toda a distribuição do material em determinada dimensão. Apresentou-se por estes autores o modelo de material chamado SIMP (Material Isotrópico Sólido com Penalização) ou também conhecido como abordagem da densidade. O termo "Material" que em alguns artigos aparece também como "Microestrutura" no título do SIMP deu-se porque no início desenvolveu-se materiais porosos, constituídos por vazios e que tinham várias distribuições. Esta abordagem tida como uma maneira fácil mas artificial de reduzir a complexabilidade da aproximação da homogeneização, desenvolveu-se para melhorar a convergência intermediária das soluções 0 – 1. Estas contribuições causaram um vasto desenvolvimento no campo da otimização topológica. No método há a possibilidade de encontrar estas posições denominadas vazias, claras (0) ou sólidas, escuras (1), e também as regiões intermediárias "escala de cinza". Segundo [Silva \(2003b\)](#), na figura 4 é ilustrado a distribuição do material por meio do MOT no interior do domínio fixo ao longo das iterações (mostrando a simetria do domínio para simplificar os cálculos), em que o problema consiste em obter a minimização do compliance da estrutura obtendo a máxima rigidez.

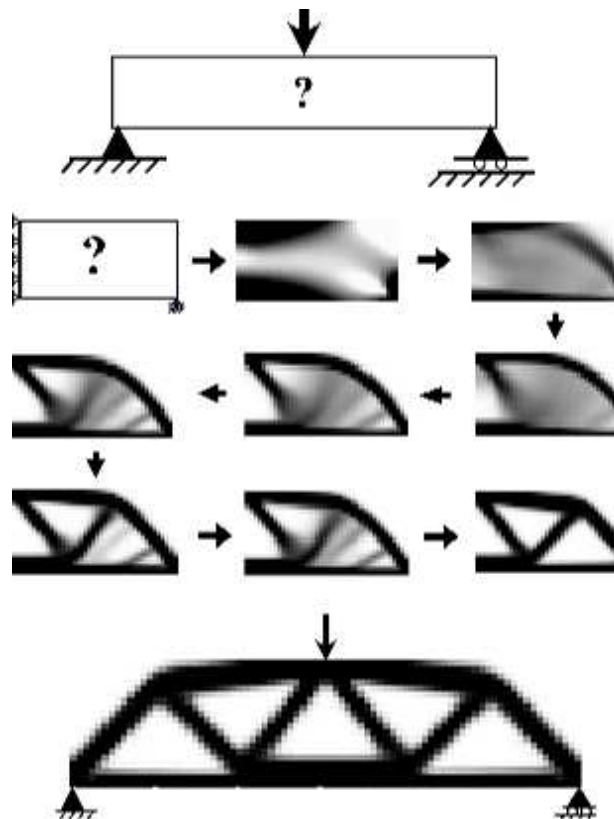


Figura 4 – Princípio de otimização topológica com região vazia, clara (0), região cheia, escura (1). Fonte: ([SILVA, 2003b](#))

No MOT devemos coletar os dados, objetivos, recursos etc, assim, é observado

quais são os dados do material, como propriedades mecânicas (por exemplo, o módulo de elasticidade), quais são os objetivos, por exemplo minimizar o material com as restrições como o peso, e os recursos como por exemplo a tensão normal e tensão de cisalhamento. Na formulação é também definido as variáveis do projeto que são as pseudo-densidades. A dimensão da viga ou outra forma são configuradas por meio do número de elementos da matriz de rigidez, estas podem ser chamadas como as dimensões do modelo. Segundo [Abbey \(2017a\)](#) o modelo de material SIMP é conhecido como um método "Soft Kill", que permite um range de minimização de valor baixo e muito alto da rigidez. Assim, temos a formulação do problema de otimização topológica com algumas variações em relação ao problema básico de otimização.

### 3.2.1.3 Formulação do problema de otimização topológica SIMP

Há várias demonstrações de equações para o MOT, isso em função da abordagem a ser utilizada e do objetivo a ser minimizado ou maximizado, no entanto a seguir demonstra-se a formulação mais utilizada do MOT incluindo os parâmetros função objetivo, a variável de projeto, a restrição do projeto e a condição de equilíbrio da estrutura conforme [Sigmund \(2001\)](#), em que o objetivo é minimizar a flexibilidade média (compliance). Verifica-se que nesta formulação o valor da variável de projeto  $\rho$  que são as pseudo-densidades podem assumir valores intermediários entre 0 e 1 que serve para relaxar o método conforme será discutido posteriormente, assim o problema em estudo é o de minimizar a função compliance (flexibilidade média) que garante a máxima rigidez estrutural e também inserida uma restrição de volume de material, sendo um problema não-linear, e que pode ser escrita como na equação 3.3. Sabe-se que para outras aplicações, como por exemplo para mecânicos flexíveis, em que o critério é maximizar a rigidez e também maximizar a flexibilidade, outra função objetiva deverá ser utilizada, assim como para aplicações de frequência de ressonância etc.

$$\begin{aligned}
 \text{Minimizar}_x & : c(X) = \mathbf{U}^T \mathbf{K} \mathbf{U} = \sum_{e=1}^N (\rho_e)^p \mathbf{u}_e^T \mathbf{k}_0 \mathbf{u}_e & (\text{função objetivo}) \\
 \text{Sujeito à} & : \frac{V(X)}{V_0} = f & (\text{fração volume}) \\
 & : 0 < \rho_{min} \leq \rho \leq 1 & (\text{variáveis projeto}) \\
 \text{Em condição de} & : \mathbf{K} \mathbf{U} = \mathbf{F} & (\text{equação equilíbrio})
 \end{aligned} \tag{3.3}$$

Em que:

$c(X)$  : função objetivo que é a flexibilidade média (rigidez estrutural)

$\rho$  : Vetor de variável do projeto como as pseudo-densidades dos elementos.

$N$  : Número totais de elementos para discretizar o domínio  $N = (\mathbf{nelx} \cdot \mathbf{nely})$

$p$  : valor da penalização (tipicamente  $p=3$ )

$V(X)$  : volume de material final

$V_0$  : volume de material inicial

$f(volfrac)$  : é a fração de volume

$\mathbf{F}$  : Vetor força

$\mathbf{U}$  : Deslocamento global

$\mathbf{K}$  : matriz de rigidez global

$\mathbf{u}_e$  : Vetor deslocamento do elemento

$\mathbf{k}_e$  : Vetor matriz de rigidez do elemento

$\rho_{min}$  : Vetor de pseudo-densidade relativa mínima para eliminar singularidade.

#### 3.2.1.4 Função Objetivo na formulação

Carvalho e Almeida (2013) descreve que o critério matemático que mede a qualidade de cada solução avaliada é chamada de função objetivo. Abbey (2017a) descreve que a função de minimização procura diminuir a energia de deformação, então a função objetivo pode ser escrita como minimizar o compliance, esta função é um problema de otimização estrutural clássico. O Compliance se dá pela força distribuída multiplicada pela somatória dos deslocamentos. Por este fato a minimização do compliance está diretamente relacionado com a máxima rigidez, assim qualquer ação que minimize a distribuição de deslocamento maximizará a rigidez. Então na função objetivo temos conceitos equivalentes como minimizar a energia de deformação, maximizar a rigidez ou minimizar o compliance. A função objetivo também pode ser chamada como flexibilidade média que está relacionada a rigidez estrutural. Na formulação temos a parcela da rigidez, assim como observado o inverso da flexibilidade da-se o aumento da rigidez.

#### 3.2.1.5 Restrições na formulação

No problema da equação 3.3 temos consideradamente duas restrições. A primeira é em relação a fração de volume e a segunda em relação a equação de equilíbrio. Normalmente a equação de equilíbrio não é vista como uma questão de rigidez, mas sim como uma condição de equilíbrio para que o MEF atue, assim, esta condição precisa existir para garantir o equilíbrio do sistema. Quando houver os elementos finitos, com a geração do modelo de malha e a definição das condições de contorno, a condição de equilíbrio é necessária, em que por exemplo, se não existir a posição dos nós fixos não será possível resolver a formulação. A rigidez está garantida na função objetivo.

### 3.2.1.6 Variáveis de projeto na formulação

Segundo [Ramos \(2007\)](#) as variáveis de projeto são os parâmetros do problema que podem ser alterados para otimizar o sistema, e que podem ser discretas ou contínuas.

As variáveis de projeto  $\rho$  podem assumir valores entre 0 e 1. Como o modelo SIMP é baseado em gradientes, diferente dos métodos heurísticos ou estocásticos, para se ter uma ótima distribuição do material no interior do domínio, o uso das derivadas é necessário. Os valores de  $\rho$  que deveremos encontrar são 0 ou 1, porém se houver algum valor intermediário gera escala de cinza. A estratégia consiste em tentar transformar os valores contínuos para valores discretos, isto é feito com a penalização dentro da função objetiva, em que basicamente o  $\rho$  é uma função exponencial que usa o penalizador.

Se tivermos por exemplo um valor de  $\rho$  próximo de 0,01 quando aplicamos o penalizador a variável de projeto tende ir para zero. Se tivermos um valor de  $\rho$  próximo de 0,99, quando aplicamos o penalizador a variável de projeto tende a ir para 1. Assim a transformação das variáveis de projeto intermediárias da função contínua para 0 ou 1 são chamadas de relaxação. Portanto a característica do modelo de material SIMP está extremamente ligada ao valor do penalizador na função.

### 3.2.1.7 Características SIMP

No SIMP a relação entre a variável de projeto  $\rho$  sendo uma função exponencial do penalizador  $p$ , e a propriedade do material para um material parcialmente isotrópico é dada pelo modelo ([BENDSØE; SIGMUND, 1999](#)) demonstradas na equação 3.4, conforme discutimos acima.

O modelo SIMP é uma equação matemática que define o material dentro do domínio. Este modelo é o que faz a relaxação do material, e que é resolvido uma função continua exponencial do penalizador.

$$E(\rho) = \rho^p E_0, \quad (3.4)$$

Em que:

$p$  = parâmetro de penalização

$E_0$  = módulo de Young do material sólido ( $N/m^2$ )

A penalização é introduzida para eliminar os efeitos do problema de "escala de cinza" que são valores alocados entre a variável 0 – 1 após a sintetização. Assim temos algumas diferenças para o valor das penalizações, para:

$p = 1$  : problema de otimização de camada de espessura variável, que atualmente para o objetivo de minimizar a flexibilidade (compliance) é denominado problema convexo



com única solução, (PETERSSON, 1999) e (BENDSØE; SIGMUND, 2003).

$p > 1$  : alguns objetos penalizam espessuras ou densidades intermediárias e portanto, favorecem as soluções 0-1. Com a escolha de  $p$  muito baixo ou muito alto, pode-se originar o efeito de escala de cinza ou uma convergência muito rápida para os locais.

$p = 3$  : É considerado um "número ideal" para boas convergências na maioria das soluções 0 – 1. Bendsøe e Sigmund (1999) confirmou este número para garantir realizações físicas de elementos com pseudo-densidades intermediárias.

Eschenauer e Olhoff (2001) nos dá uma visão da função de penalização de qualquer densidade que esteja na região cinza, a partir da figura 5 que mostra como a função de penalidade é usada, em que quanto maior o valor do expoente do penalizador maior é a curvatura baseada nos valores da pseudo-densidade com a rigidez relativa. A pseudo-densidade é ajustada a partir do seu valor nominal com uma polarização em direção aos extremos de 0,0 ou 1,0. A ordem da função de penalidade  $p$ , geralmente é fornecida como um controle de usuário. O ajuste  $p$  superior a 1 dará estruturas mais realistas com menos cinza. Em algum lugar entre 2 e 4 normalmente é usado, mas vale a pena experimentar para ver o efeito as diferentes condições.

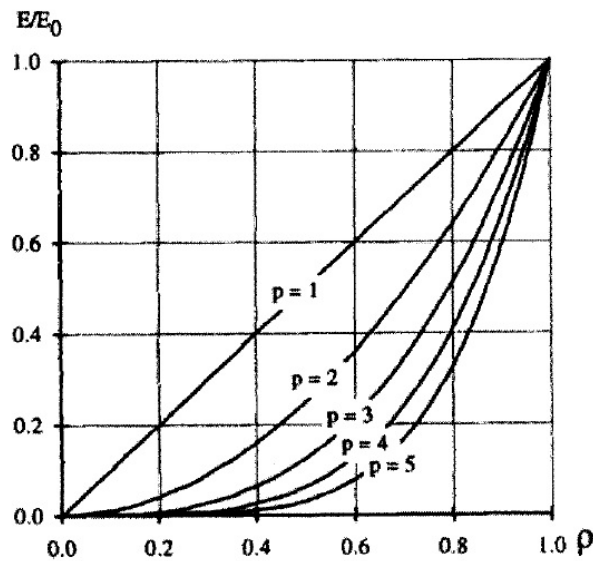


Figura 5 – Rigidez Relativa vs densidade de volume para o material SIMP para diferentes valores de penalização. Fonte: (ESCHENAUER; OLHOFF, 2001).

Na formulação SIMP deveremos garantir que a matriz de rigidez do elemento que no caso na equação 3.3 aparece em função de  $\rho$  se torne singular. Se verificarmos a equação de equilíbrio  $\mathbf{KU}=\mathbf{F}$ , temos ali a matriz de rigidez global, que é a matriz de somatória de cada rigidez do elemento e que garante a condição de equilíbrio, as variáveis de projeto  $\rho$  estão na função objetiva, assim os valores de  $\rho$  aparecem implicitamente na matriz de rigidez, ele não aparece no campo de deslocamento  $\mathbf{U}$ . Se o  $\rho$  é zero, se torna singular pois não consegue o equilíbrio  $\mathbf{KU}=\mathbf{F}$ , pois não há solução para o problema numérico.



De acordo com a informação acima e como observado em [Alphen \(2017\)](#), a equação 3.4 não é válida pois a rigidez do elemento se torna zero quando a densidade é zero, que faz com que os elementos finitos por meio da matriz de rigidez se torne singular. Assim, a 3.5 frente a 3.4, propõe um valor de  $E_{min}$ , auxiliando também na implementação simples dos filtros de sensibilidades etc. Existem duas alternativas para resolver o problema da singularidade, a primeira é não deixar que o valor de  $\rho$  seja 0, ou seja, devemos determinar um valor mínimo de  $\rho_{min}$ , a outra forma é verificada na nova equação para atender a abordagem SIMP em relação as propriedades do material são descritas na equação 3.5 em que inserimos no algoritmo um valor de propriedade do material considerado  $E_{min}$ .

$$E(\rho) = E_{min} + \rho^p (E_0 - E_{min}) \quad (3.5)$$

## 3.2.2 Campo Baseado na Geometria

### 3.2.2.1 Abordagem de *Level Set*

Segundo [Suresh \(2016\)](#) a abordagem de *Level Set* é uma outra estratégia para resolver problemas via conjunto de níveis em que o conceito se dá por meio de contornos ao invés das variáveis de projetos locais, ou seja, é baseado na geometria. Diferentemente do SIMP há uma vantagem em utilizar a estratégia *Level Set* que é a definição em todos os momentos do limite do domínio, como vemos em [Dijk et al. \(2013\)](#) que recentemente descreveu as diferenças da abordagem *Level Set*. Esta abordagem foi descrita por alguns autores como [Allaire, Jouve e Toader \(2002\)](#), [Allaire, Jouve e Toader \(2004\)](#) e [Wang, Wang e Guo \(2003\)](#), e foi primeiro desenvolvida por [Osher e Sethian \(1988\)](#), porém é uma abordagem que somente vem sendo difundida com a combinação de outras estratégias como SIMP e BESO.

Há várias utilizações com o conceito de *Level Set*, uma delas é a denominada *Pareto*. Segundo [Suresh \(2016\)](#), *Pareto* é um tipo particular de função LS em que o campo da sensibilidade topológica é aplicado. Segundo [Abbey \(2017b\)](#) a sensibilidade topológica é uma medida da mudança para qualquer resposta estrutural, com base na inserção de um pequeno orifício na estrutura. O pequeno buraco modifica a topologia. Uma alta sensibilidade significa que qualquer material removido, e que, portanto, altera a topologia, terá um grande efeito sobre a resposta estrutural. Por outro lado, a baixa sensibilidade significa que há pouco impacto da mudança na topologia. Se o compliance for utilizada como resposta estrutural, podem ser estabelecidas definições analíticas da sensibilidade da topologia. No LS é observado a eliminação de elementos como vemos em SIMP e no BESO devido a regiões com baixa sensibilidade, e através do valor denotado pelo usuário dos níveis da estrutura, é possível verificar a topologia da estrutura.

Conforme vemos em [Alphen \(2017\)](#), o LS não é um método baseado em gradiente,

mas é desenvolvido somente para ter dois valores das variáveis de densidades sendo discretos, 1 para sólidos e 0 para vazios, e com estes valores claramente será possível formar o contorno resultante do problema de OT de encontrar uma função objetivo com mínimo compliance da estrutura, sujeito a uma restrição de volume. Tal problema para esta abordagem é descrito de acordo com a equação 3.6.

$$\begin{aligned}
 \text{Minimizar}_X & : c(X) = \mathbf{U}^T \mathbf{K} \mathbf{U} = \sum_{e=1}^N \mathbf{u}_e^T \mathbf{k}_e \mathbf{u}_e = \sum_{e=1}^N x_e \mathbf{u}_e^T \mathbf{k}_1 \mathbf{u}_e \quad (\text{função objetivo}) \\
 \text{Sujeito à} & : V(X) = V_{req} \quad (\text{fração volume}) \\
 & : x_e = 0 \text{ ou } x_e = 1 \quad (\text{variáveis projeto}) \\
 \text{Em condição de} & : \mathbf{K} \mathbf{U} = \mathbf{F} \quad (\text{equação equilíbrio})
 \end{aligned} \tag{3.6}$$

Em que:

$c(X)$  : função objetivo que é a flexibilidade média (rigidez estrutural)

$N$  : Número totais de elementos para discretizar o domínio  $N = (\text{nelx} \cdot \text{nely})$

$x$  : Vetor variável do projeto com valores 0 ou 1.

$\mathbf{u}_e$  : Vetor deslocamento do elemento

$\mathbf{k}_e$  : Vetor matriz de rigidez para um elemento

$\mathbf{k}_1$  : Vetor matriz de rigidez para um elemento sólido

$V(X)$  : Número de elementos sólidos

$V_{req}$  : Número de elementos sólidos requeridos

$\mathbf{F}$  : Vetor força

$\mathbf{U}$  : Deslocamento global

$\mathbf{K}$  : matriz de rigidez global

Ainda segundo [Alphen \(2017\)](#) , para o processo do LS é observada a figura 6, em que de um domínio sólido total  $D$  tem as condições de contorno que são observadas pela  $\partial D$ . Para o LS, escolhe-se uma parcela menor de domínio denominada  $\overline{D}$ , que é uma garantia de que ali sempre contemplará a estrutura. Assim, o modelo de LS é dado por  $s$  que representa a dimensão em uma determinado nível da função que na figura é dada como  $\psi(\mathbf{p}, t)$  que é também conhecida como função de LS.

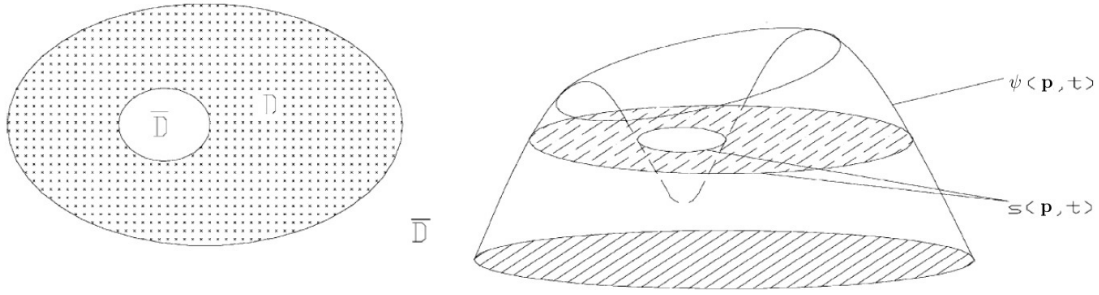


Figura 6 – Fatores utilizados no LS. Fonte: (ALPHEN, 2017)

Então a função de LS é dada pela equação 3.7.

$$\begin{cases} \psi(\mathbf{p}) < 0 \text{ Se } \mathbf{p} \in D \\ \psi(\mathbf{p}) = 0 \text{ Se } \mathbf{p} \in \partial D \\ \psi(\mathbf{p}) > 0 \text{ Se } \mathbf{p} \notin D \end{cases} \quad (\text{Condições Função LS}) \quad (3.7)$$

No caso o valor de  $\mathbf{p}$  é um ponto qualquer dado para o domínio do projeto e que sempre garante que a função de LS tenha que ser  $< 0$  para estar dentro do domínio  $D$ , também conhecido como nível zero da função. Para encontrar sempre este nível é feito a atualização da função do LS no algoritmo.

Com isto, a função de LS é atualizada pela generalização da equação de Hamilton-Jacobi demonstradas na equação 3.8.

$$\frac{\partial \psi}{\partial t} = -v \mid \nabla \psi \mid - \omega g \quad (\text{Atualização Função LS}) \quad (3.8)$$

Em que:

$t$ : É o tempo.

$v(\mathbf{p})$ : É o campo escalar sobre o domínio projetado, na função de velocidade.

$g(\mathbf{p})$ : É o campo escalar sobre o domínio projetado e força novos furos para serem criados em  $\bar{D}$ .

$\omega$ : É o parâmetro positivo que determina a influência de  $g$ .

O método de LS é configurado de forma a encontrar somente valores para um mínimo local, não se tem o encontro de valores para um mínimo global. Desta forma apesar de ser matematicamente mais rápido do que o modelo SIMP por exemplo, não se pode garantir soluções de mínimo global para a função objetiva.

O uso do método LS frente ao modelo SIMP é bem utilizado no sentido de que não forma as chamadas escalas de cinza, que são variáveis de projeto intermediários, no

entanto o SIMP é um método mais fácil de ser implementado, diferente do LS que utiliza funções numéricas mais complexas.

Challis e Guest (2009) definiu um algoritmo mono-objetivo utilizando o LS, Suresh (2010) definiu um algoritmo baseado no LS para funções multi-objetivos, que depois foi transformado no programa computacional ParetoWorks e que utilizou portanto, conceitos destes algoritmos. O ParetoWorks trabalha com base no MLS e o LS é calculado por meio de sensibilidade topológica. O ParetoWorks também utiliza o MMA como otimizador e não depende da penalização como por exemplo o modelo de material SIMP. A sensibilidade topológica do LS é utilizada como filtro.

### 3.2.2.2 Abordagem de Derivadas Topológicas

O uso de derivadas topológicas em OT e OF foi iniciado por Eschenauer, Kobelev e Schumacher (1994), e então foi denominado o método da bolha (bubble method). A ideia básica é prever a influência (derivada) de introduzir um buraco infinitesimal em qualquer ponto  $x$  no domínio de projeto  $\Omega$  e usar isso como o motor para a geração de novos buracos.

Amstutz (2011) argumenta que os gradientes topológicos para a minimização de compliance correspondem aos gradientes de densidade padrão, tal como utilizados no esquema SIMP, para as escolhas gerais das propriedades dos materiais (isto é, coeficiente de Poisson  $\mu = 1/3$ ). Em particular, observa que os dois esquemas dão resultados iguais para a escolha dos expoentes de penalização  $p = 2$  (para problemas térmicos 2D) e  $p = 3$  (para problemas elásticos 2D com coeficiente de Poisson  $1/3$ ). Como observado no artigo de Amstutz (2011), esta observação está em perfeita correspondência com as recomendações para os expoentes de penalidades dadas em Bendsøe e Sigmund (1999), o que não deve ser surpreendente. Os limites para a realização física da lei SIMP estavam em Bendsøe e Sigmund (1999) determinados pela rigidez do SIMP que tem ser menor ou igual ao limite superior de Hashin-Strikhman no limite da fração de alto volume (pequenos vazios). O limite superior de Hashin-Strikhman é, por sua vez, realizado por um composto periódico com pequenos furos circulares (VIGDERGAUZ, 1999) e, portanto, a sensibilidade de introduzir um pequeno orifício circular (isto é, a derivada topológica) pode ser diretamente encontrada no estudo do limite da propriedade de material ótima.

### 3.2.2.3 Abordagem de *Phase-Field*

(BOURDIN; CHAMBOLLE, 2003) descreve a abordagem *Phase-Field*, que é uma abordagem voltada a resolver problemas matemáticos interfaciais. Algumas aplicações são utilizadas como dinâmica da solidificação, fraturas dinâmicas etc. Assim normalmente vem sendo utilizada na aplicação de fluidos.

#### 3.2.2.4 Abordagem evolucionária

Xie e Steven (1993) descreve e compara a evolução da abordagem evolucionária, esta abordagem baseia-se totalmente em meios discretos, assim é considerada uma abordagem heurística, que são métodos que procuram resolver de uma forma mais simples os processos, em que a aceleração computacional se dá pela eliminação de elementos e, portanto, eliminação dos graus de liberdade, podendo ser muito significativos. Normalmente as sintetizações que são baseadas nesta abordagem são computacionalmente mais rápidas de serem resolvidos em comparação com a abordagem SIMP por exemplo.

Abbey (2017b) demonstra que houveram evoluções frente a esta abordagem. A abordagem evolucionária se iniciou com o método de otimização estrutural evolutiva (ESO) e consequentemente foi adaptada para o método bidirecional de otimização estrutural evolutiva (BESO). A diferença é que o primeiro exclui elementos a partir de uma determinada restrição e o segundo usa várias estratégias para voltar com os elementos excluídos, sendo assim, semelhantes às técnicas de suavização e filtragem do método SIMP. O método BESO também pode introduzir um elemento de "soft kill", na medida em que os elementos não são totalmente excluídos. Isso é feito para evitar sensibilidades singulares associadas aos elementos excluídos. Uma pequena rigidez de material maior do que zero é introduzida e as sensibilidades significativas podem ser calculadas. Um método de penalização, semelhante ao do SIMP, conduz o material às condições extremas. No entanto, a distribuição de material ainda é essencialmente binária, em oposição à variável contínua no SIMP. Por ser uma abordagem heurística é tida como difícil em encontrar a base matemática e também a avaliação das convergências, mas é dita uma abordagem popular em que os pesquisadores tentam reduzir tais problemas.

#### 3.2.2.5 Comparação dos métodos de otimização topológica

Como verificado em Alphen (2017), a escolha de uso de um método ou de outro se dá na aplicação dos domínios de projeto. Este autor cita por exemplo uma aplicação para construção de concretos baseado em manufatura aditiva em que define que o SIMP e o BESO são boas escolhas para estruturas tridimensionais. No entanto o autor informa que a comparação entre SIMP e LS requerem adicionais desenvolvimentos. O que se tem em relação a abordagem LS frente a SIMP é que o modelo de material SIMP por trabalhar com gradientes, na maioria das aplicações tem um tempo de cálculo maior do que as outras abordagens, e a LS apresenta vantagens por trabalhar mais facilmente com problemas multiobjetivos e que tem particularidades em ter os elementos com menos detalhamento pois não produz elementos com pseudo-densidades intermediárias, mas o fato de trabalharem com variáveis discretas 0 e 1 acelera o processo matemático. Discutiremos nos próximos capítulos os algoritmos educacionais desenvolvidos para estas abordagens, e os programas computacionais atuais, além de propor implementações para o objeto de

estudo (longarina do chassi veicular).

### 3.3 Problemas de instabilidade numérica na OT

Neste tópico são descritos os principais problemas encontrados na discretização de uma OT, que muitas vezes fazem com que o processo de iteração aumente, crescendo o tempo para se chegar a um ótimo local ou ótimo global. São apresentados os problemas do tabuleiro de xadrez, de soluções em ilhas, de dependência de malhas, e das soluções para problemas de mínimo local.

[Carvalho e Almeida \(2013\)](#) descreve que a maioria dos problemas são regulados por esquemas heurísticos que corrigem falhas do campo de densidade ou do campo de sensibilidades no cálculo numérico, isto é feito na maior parte das vezes por meio de uma média ponderada dos seus valores em uma vizinhança do elemento.

Na figura 7 observamos a forma como ocorrem estes problemas em que (a) cria padrões que alternam elementos vazios e sólidos, (b) isolam regiões de materiais em meio a regiões vazias sem conexão com as partes do material e (c e d) ao invés de apresentar uma melhor condição de contorno apresenta estruturas cada vez mais detalhadas.

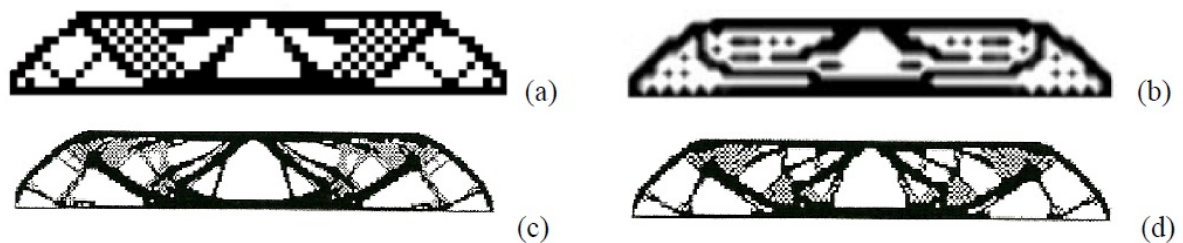


Figura 7 – (a) Tabuleiro de xadrez, (b) Solução em ilhas, (c) Dependência de malha (240x40 elementos), (d) Dependência de malha (180x30 elementos). Fonte: ([CARVALHO; ALMEIDA, 2013](#)) .

#### 3.3.1 Problema do tabuleiro de xadrez

É um problema encontrado no MEF referente ao processo de OT que é conhecido do termo em inglês como *checkerboard*, em que apresenta-se por meio de uma região em que há um quadrante de componentes vazios/sólidos sequencialmente na vizinhança. Esta região de elementos da malha representam uma sintetização inadequada para a OT, mas são favorecidas pelo processo de otimização, pois claramente verifica-se o elemento de cor branca (vazia) e cor sólida (preta), 0 ou 1 numericamente. A independência de malha descrita na sequência auxilia na minimização do tabuleiro de xadrez. Os *checkerboards* podem ser evitados para elementos de ordem superior e podem ser atendidos por métodos



de restrição de independência de malhas, (DIAZ; SIGMUND, 1995), (JOG; HABER, 1996), (SIGMUND; PETERSSON, 1998), ver figura 8.



Figura 8 – Problema do *checkerboard* (SILVA, 2003b) .

### 3.3.2 Problema de formação de ilhas

Como vemos na figura 7b, observa-se regiões muito isoladas de material, sem que haja a conexão entre elas por exemplo pela modelo de material SIMP que tem apenas uma variável por cada elemento. Isto ocorre devido a formulação denominada de aproximação contínua da distribuição do material (CAMD) (MATSUI; TERADA, 2004), é este processo que gera a formação de ilhas por meio do *checkerboard*. Esta formulação, diferentemente do SIMP em que é considerado uma pseudo-densidade no elemento, faz com que tenha-se variáveis de projeto nos nós do elemento ao invés de estar no próprio elemento, fazendo com que tenha-se uma análise do método para conhecermos as variáveis em qualquer ponto. Esta técnica consiste em distribuir o material no interior do domínio de uma maneira contínua, e consequentemente, permitir uma variação contínua da propriedade do material (LIMA, 2006) pois se tem o controle das variáveis nos nós. Implementou-se esta solução em várias aplicações, como no algoritmo para impedâncias elétricas desenvolvido em Lima (2006). O método foi apresentado para melhorar o problema de dependência de malha, isto foi a principal questão em relação a implantação do método, mas verificou-se depois que na parte do *checkerboard* obrinha-se esta formação de figuras em formato de ilhas. Assim a figura 7b nada mais é do que um *checkerboard* com o uso da técnica CAMD. Para resolver o problema de formação de ilhas, em que temos com o CAMD uma tentativa de melhorar o controle dos elementos de uma forma "grosseira" para uma forma mais "suave", em relação a discretização do domínio na sintetização, temos que utilizar as técnicas de filtragem para fazer uma normalização dos resultados da discretização.

### 3.3.3 Problema da dependência de malhas

[Abbey \(2017a\)](#) descreve o problema de dependência de malha como tendenciosa a ter um alto nível de porosidade, em que o tamanho do elemento da malha é diretamente relacionado com o problema da dependência de malha. Uma malha muito fina deixa a estrutura "fibrosa" semelhante a uma espuma, e uma malha muito grossa restringe a distribuição do material na otimização. Assim alguns recursos são utilizados para reduzir o problema de dependência de malhas, que são:

- Controle do Perímetro.
- Controle do gradiente de densidade.
- Filtros de sensibilidade e densidade.

O controle do perímetro é utilizado quando queremos por exemplo limitar o número de furos em uma OT. No entanto a limitação do número de furos não é intuitiva. Normalmente é analisado o elemento e o seu vizinho e medido o salto e o comprimento do elemento. Já no controle de gradiente de densidade procura limitar o gradiente de densidade e distribui-lo por meio de um comprimento significativo, e como a penalidade é aplicada nas densidades intermediárias durante as iterações a distribuição que teria um efeito de pirâmide fica mais homogênea tendendo a ficar quadrada. Os filtros de sensibilidade e densidade são utilizados para gerar dentro de um perímetro circular em volta de um elemento alvo a condição de definição do material sendo vazio ou sólido, assim todos os elementos dentro deste círculo de raio  $R$  normalmente definido pelo usuário passam a serem sensíveis e terem a mesma aparência, isto auxilia a não ter buracos demasiados em certas regiões. Assim, estes recursos são utilizados para deixar a OT independente de malhas.

Segundo [Ahmed \(2013\)](#), um destes recursos conforme exposto é o filtro de independência de malhas que também é conhecido como filtro de sensibilidade de Sigmund ([SIGMUND, 2001](#)), que no caso é baseado em gradientes (SIMP) e é usado para evitar a formação de buracos e do tabuleiro de xadrez, assim a formulação matemática trabalha modificando o gradiente do elemento  $\frac{\partial c}{\partial \rho_e}$ , pois ele corrige a sensibilidade da função objetivo em relação a densidade, utilizando densidade de elementos  $\rho_j$  localizados dentro de uma região circular (no caso 2D) e uma região esférica (no caso 3D) de acordo com a equação 3.9.

$$\frac{\widehat{\partial c}}{\partial \rho_e} = \frac{1}{\rho_e \sum_{f=1}^N \hat{H}_{ef}} \sum_{f=1}^N \hat{H}_{ef} \rho_f \frac{\partial c}{\partial \rho_f}, \quad (\text{Filtro de independência}) \quad (3.9)$$

Em que:



$\hat{H}_{ef}$  : É o operador de convolução (fator de peso), que é escrito de acordo com a equação 3.10.

$$\hat{H}_{ef} = r_{min} - dist(e, f), \quad \{f \in N \mid dist(e, f) \leq r_{min}\}, \quad e = 1, 2, \dots, N \quad (3.10)$$

Assim:

$N$  : Número total de elementos da malha

$dist(e, f)$  : Operador que define a distância entre o centro do elemento  $e$  e o centro do elemento  $f$ .

### 3.3.4 Problemas mínimo local

#### 3.3.4.1 Ótimo local vs Ótimo global

A maior parte das sintetizações de OT não convergem para um ótimo global, e sim para um ótimo local. Aquele ótimo local encontrado é uma solução ótimo porém não se pode garantir que é a melhor solução para aquele determinado problema, mas não deixa de ser ótimo. As abordagens de homogeneização e SIMP são consideradas "soft-kill" por serem contínuas na maioria das vezes e que podem apresentar soluções de ótimos globais, diferentemente das abordagens evolucionárias que por serem discretas trabalham com eliminação de elementos são e consideradas "hard-kill" por apresentarem dificuldades em encontrar um ótimo global, assim na maioria das vezes convergem para um ótimo local, que se for para trabalhar com uma função de minimização são consideradas mínimos locais. Rao (2009) demonstra que o resultado das funções podem ser apresentados de acordo com a figura 9.

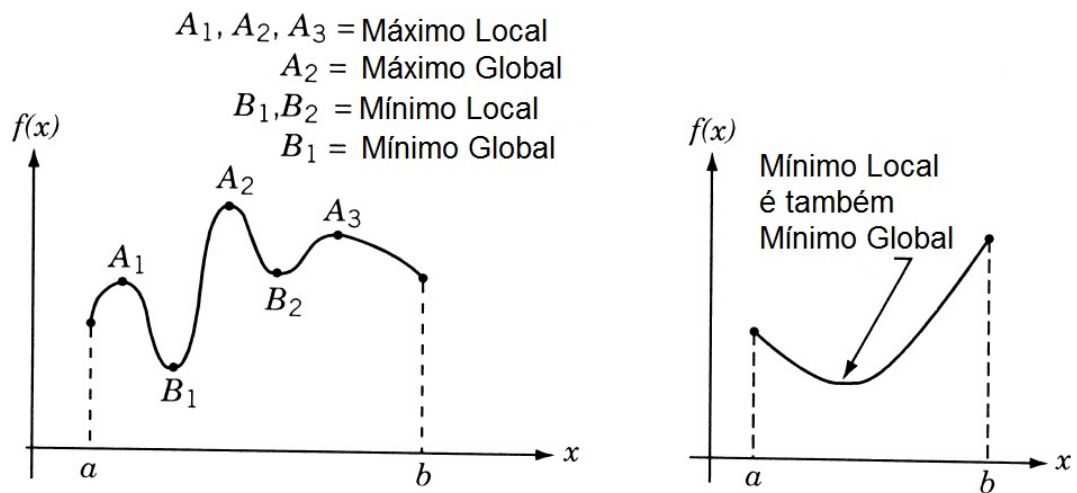


Figura 9 – Ótimo local ou global. Fonte: Adaptado de Rao (2009).

Como verificado o ponto B1 da função pode ser conhecido como ótimo global, ou mínimo global. Já o ponto B2 é conhecido como um ótimo local ou mínimo local. Assim como o ponto A1, A2 e A3 que podem ser conhecidos como máximos locais, e somente o A2 como máximo global. Também há funções que o mínimo local são também mínimos globais e portanto ótimos globais conforme observamos na função do lado direito da figura 9.

O conceito destas convergências nas funções tem que ser bastante estudado pelo usuário, e caso se deseje ter como resposta uma solução de ótimo global uma determinada abordagem deve ser direcionada. No SIMP há a possibilidade de diminuirmos o passo das iterações, fazendo com que tenha-se mais ótimos locais, tendendo a convergência para um ótimo global, porém por mais que diminuamos o passo não temos garantia da solução ótimo global e na maioria das vezes a sintetização de OT começa a ficar lenta necessitando muitos recursos computacionais para poder convergir.

Assim o critério de solução adotado é um parâmetro que deve ser observado para uma melhor solução, pois pode se tornar um grande problema.

## 4 Implantação numérica

### 4.1 Introdução Teórica

[Lima \(2002\)](#) descreve que a solução para o problema de OT da eq. 3.3, é encontrada por meio de um algoritmo computacional que combina o método de otimização linear sequencial com o método numérico de análise por elementos finitos. Porém o PLS é só um método dentro outros possíveis que veremos a seguir. Assim [Lima \(2002\)](#) divide a classificação destes métodos em específico e genéricos que são baseados na teoria da programação matemática, em que os específicos são o critério de otimalidade (CO) e o Método das Assintotas Móveis (MMA) de [Svanberg \(1987\)](#), e os genéricos são classificados em programação linear (PL), programação não-linear (PNL), programação linear sequencial (PLS) e programação quadrática sequencial (PQS). Estes métodos de otimização também são conhecidos simplesmente como otimizadores.

#### 4.1.1 Otimizadores (Métodos de Otimização)

Na figura 10, baseado em [Griva, Nash e Sofer \(2009\)](#), temos estes principais otimizadores para o MOT baseado em gradientes para otimização estrutural. Estes otimizadores procuram simplificar as funções objetivo antes do processo de otimização para trazer para uma realidade em que a computação matemática consegue sintetizar e aproximar os cálculos, sendo assim, também são conhecidos como métodos de aproximação explícitas.

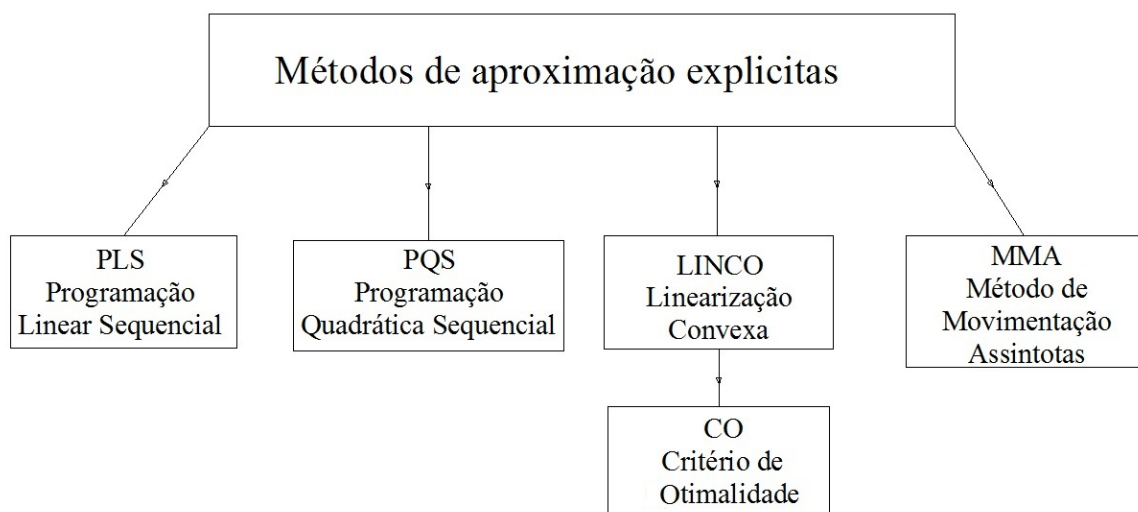


Figura 10 – Principais otimizadores baseados em gradientes para otimização estrutural. Fonte: ([GRIVA; NASH; SOFER, 2009](#)).

De acordo com [Rao \(2009\)](#) diversas são as técnicas para otimização que foram desenvolvidas na história. As técnicas são providas para encontrar o mínimo de uma função de uma ou várias variáveis ou de problemas complexos sob um conjunto prescrito de restrições, que podem ser divididas pela característica de suas funções, com variáveis unidimensionais (função com uma variável) ou multidimensional (função com mais de uma variável). Outra base de referência em relação a estas técnicas é verificada em [Martinez e Santos \(1998\)](#), ao qual é afirmado que a problemática de adequar um modelo matemático a uma situação real também pode ser formulada como um problema matemático, quase sempre de otimização. Em [Ravindran, Reklaitis e Ragsdell \(2006\)](#) também temos a classificação das funções de otimização numérica, que são classificadas pelo número de variáveis de acordo com a tabela 8.

Tabela 8 – Classificação das funções de otimização. Fonte: Criada pelo autor com base em [Rao \(2009\)](#)

Tipo da função	Característica
Unidimensional	Uma variável
Multidimensional com ou sem restrições	Várias variáveis
Problemas de programação convexas	Várias variáveis

#### 4.1.1.1 Programação por Critério de Otimalidade (CO)

[Abbey \(2017a\)](#) descreve que a OT age de acordo com a discretização de um domínio sobre as suas condições de contorno, e que este domínio dá uma resposta do FEA em função da tensão e deslocamento obtidas, e com base nestas respostas FEA a OT busca remover material nos níveis baixos de tensão, então esta ação é definida como CO, e que baseia-se em uma ideia heurística.

[Lima \(2002\)](#) cita que no CO é desenvolvida uma formulação empírica específica para cada problema que é elaborado, tendo uma aplicação restrita para o critério de OC. É um método computacional eficiente e tem sido aplicado em vários problemas de otimização topológica de minimização da flexibilidade média (compliance).

Segundo [Ahmed \(2013\)](#) a formulação matemática do OC é feita com o uso de um algoritmo iterativo. Há uma atualização do esquema para as variáveis de projeto utilizado por [Bendsøe \(1989\)](#). Assim a formulação matemática se dá de acordo com a equação 4.1.

$$\rho_e^{new} = \begin{cases} \max(\rho_{min}, \rho_e - m) & \text{Se } \rho_e B_e^\eta \leq \max(\rho_{min}, \rho_e - m), \\ \rho_e B_e^\eta & \text{Se } \max(\rho_{min}, \rho_e - m) < \rho_e B_e^\eta < \min(1, \rho_e + m), \\ \min(1, \rho_e + m) & \text{Se } \min(1, \rho_e + m) \leq \rho_e B_e^\eta. \end{cases} \quad (4.1)$$

Em que:

$B_e = \frac{-\frac{\partial c}{\partial \rho_e}}{\lambda \frac{\partial V}{\partial \rho_e}}$  : Condição de Otimalidade.

$m$  : Limite de movimento, com valor de 0,2, encontrado de pesquisas experimentais.

$\eta (= 1/2)$  : Coeficiente de amortecimento numérico para estabilizar a iteração, encontrado de pesquisas experimentais.

$\lambda$  : Multiplicador de Lagrangeano.

E o gradiente da função objetiva  $c(x)$  é transformado para a equação 4.2.

$$\frac{\partial c}{\partial \rho_e} = -p(\rho_e)^{p-1} \mathbf{u}_e^T \mathbf{k}_0 \mathbf{u}_e \quad (\text{função objetivo com uso de OC}) \quad (4.2)$$

Desta forma e conforme vemos em [Alphen \(2017\)](#), o método CO é bastante importante dentro do esquema de otimização estrutural, pois ele é essencialmente o loop do processo de otimização. E com estas formulações, as densidades de todos os elementos são atualizadas para uma solução mais otimizada, até o critério de parada.

#### 4.1.1.2 Programação pelo Método das Assíntotas Móveis (MMA)

O Método das Assíntotas Móveis (MMA) foi criado por [Svanberg \(1987\)](#). A diferença entre o CO e o MMA é que enquanto o OC é um otimizador para restrições únicas e está baseado na atualização de um ponto fixo heurístico, o MMA consegue resolver mais problemas complexos.

Segundo [Ahmed \(2013\)](#) é um método para otimização estrutural baseado em problemas de um tipo especiais de aproximação convexa, em que cada etapa do processo iterativo de um sub-problema de aproximação convexa é gerado e resolvido.

#### 4.1.1.3 Programação Linear Sequencial (PLS)

Segundo [Lima \(2002\)](#), este é um método de otimização que vem sendo aplicado com bastante sucesso na otimização topológica, como no projeto de mecanismos flexíveis. O objetivo do PLS é resolver sequencialmente vários sub-problemas lineares, em busca de uma solução não-linear de funções objetivas muito complexas.

Assim o PLS é um método de programação não linear com restrição considerado como método direto. Além deste método para o uso de funções específicas lineares, temos dois métodos, o Simplex ([DANTZIG, 1963](#)), e o Kamarkar ([HAFTKA; GÜRDAL, 1992](#)). O método Simplex foi desenvolvido permitindo assim a solução de problemas de programação linear com maior eficiência, este método foi muito utilizado e atualmente várias pesquisas ainda continuam a utilizá-lo, no entanto com o advento dos estudos de otimização e a computação, várias outras funções foram sendo desenvolvidas e implementadas em paralelo a esta função.

#### 4.1.1.4 Programação Quadrática Sequencial (PQS)

A Programação Quadrática sequencial (PQS), consiste na aproximação sequencial de subproblemas lineares de programação não-linear como um problema de aproximação quadrática. O método PQS é bastante difundido em programas computacionais comerciais. A ideia básica da programação Quadrática Sequencial é a modelagem de um sistema exponencial por meio de um sub-problema de programação quadrática, e em seguida usar o resultado para construir uma melhor aproximação da função objetivo para uma forma quadrática e as restrições para uma forma linear (PQS).

#### 4.1.1.5 Outros otimizadores

Além dos otimizadores já descritos, [Rao \(2009\)](#) define outros tipos de métodos de otimização como as técnicas de processo estocástico, os métodos estatísticos e as modernas técnicas de otimização, que estão descritos na figura 11.

Programação matemática ou técnicas de otimização	Técnicas de Processo Estocástico	Métodos Estatísticos
Métodos de Cálculo	Teoria de decisão estatística	Análise de Regressão
Cálculo de Variações	Processo Markov	Análise de Agrupamento, reconhecimento de padrão
Programação Não Linear	Teoria de Filas	Design of Experiments
Programação Geométrica	Teoria Renewal	Análise de discretização (Análise fatorial)
Programação Quadrática	Métodos de Simulação	
Programação Linear	Teoria de Confiabilidade	
Programação Dinâmica		
Programação Inteira		
Programação Estocástica		
Programação Separável		
Programação multiobjetivo		
Métodos de rede: PERT e CPM		
Teoria dos Games		
<i>Técnicas de Otimização Modernas ou não tradicionais</i>		
Algoritmos Genéticos		
Arrefecimento Simulado		
Otimização de Colônia de Formigas		
Otimização por Enxame de Partículas		
Redes Neurais		
Otimização Fuzzy		

Figura 11 – Principais métodos de otimização. Fonte: Adaptado de [Rao \(2009\)](#).

Como descrito na introdução teórica da implantação numérica, os métodos de otimização são combinados com os métodos numéricos de análise por elementos finitos. A partir daqui estaremos definindo como é calculado, a forma dos elementos da malha e como o MEF interage com estes otimizadores.

## 4.2 Implementação do MEF na Otimização Topológica

Suresh (2016) nos dá uma boa visão dos métodos dos elementos finitos aplicado na otimização topológica, em que nos mostra como chegar na matriz de rigidez para elementos 1D, 2D e 3D por meio do MEF e com o uso da teoria de elasticidade linear para minimizar a energia potencial. No caso 2D por exemplo, demonstra-se que caso o comprimento e a largura de um objeto 3D seja muito maior do que a sua espessura, ele pode ser tratado com um problema de tensão planar 2D para efeito de implementação numérica.

Lima (2002) descreve como também já vimos no capítulo de história da otimização estrutural que o MEF surgiu na década de 1950, e com o advento da computação na década de 1970, passou a ser utilizado em grande escala. O MEF vem sendo utilizado na OT por ser um método robusto, capaz de lidar com estruturas de formas complexas. Assim segundo Lima (2002), todos os trabalhos consultados na literatura com respeito a OT tem o uso do MEF. O processo consiste em gerar uma malha de elementos em todo o seu domínio, e representá-las por meio de cálculo numérico em matriz numéricas.

### 4.2.1 Implementação do MEF para elementos 2D genérico

De uma forma geral, é necessário obter a matriz de rigidez global do domínio do projeto, e se as malhas tiverem um elemento quadrilátero mas que não tenham lados unitários há uma formulação para tratar que é a formulação genérica para chegar-se nesta matriz de rigidez global, ao qual cada elemento terá uma matriz de rigidez distinta. Por outro lado há alguns estudos que simplificam a solução desta matriz de rigidez definindo que os elementos são totalmente quadrados com lados unitários guais. Veremos portanto a implementação genérica e depois discutiremos a implementação simplificada.

Suresh (2016) demonstra para elementos bidimensionais conforme figura 12, 2 tipos de elementos de malha, um com elementos triangulares (lado esquerdo) e outro com elementos quadriláteros (lado direito).

Normalmente, o uso de elementos quadriláteros facilita o cálculo computacional. Assim recomenda-se sempre que possível trabalhar com elementos quadriláteros para os casos bidimensionais. No nosso trabalho propusemos trabalhar com algoritmos e programas computacionais bidimensionais e também tridimensionais. Para os elementos bidimensionais, a preparação de uma matriz de rigidez consiste em definir uma malha, considerando a forma dos elementos, o tamanho e o domínio resultando em uma quantidade total de elementos discretizados pelo MEF, e a cada iteração ou com as condições de contorno adicionadas no domínio como as restrições e as forças nodais, poder ter como resultado os deslocamentos nodais encontrados neste domínio. Sendo assim para definirmos a forma dos elementos com a forma quadrilátero temos que ter um elemento isoparamétrico de 4 nós que abrangerá as coordenadas locais de cada um dos nós.



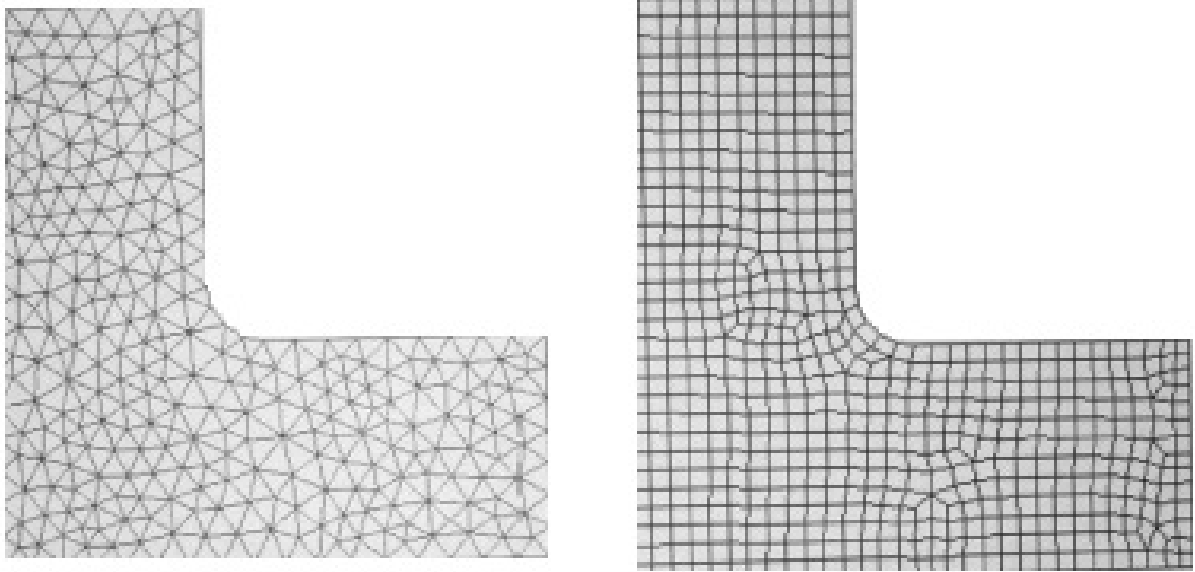
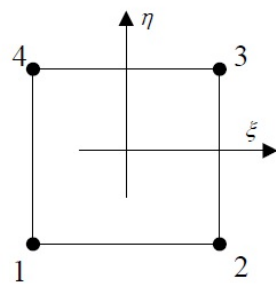
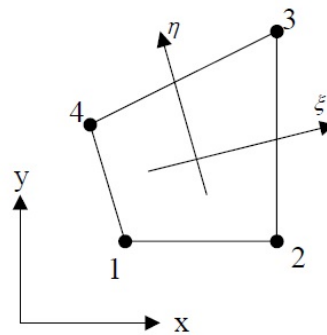


Figura 12 – Tipos de malhas bidimensionais. Fonte ([SURESH, 2016](#)) .

Os elementos de 4 nós para o caso bidimensional podem ser retangulares, ou isoparamétricos em função das coordenadas originais de deslocamento nodais do elemento, estes possuem dois graus de liberdade para o nó, que são os deslocamentos na direção horizontal (eixo  $x$ ) e na direção vertical (eixo  $y$ ), conforme figura 13.



a) elemento retangular



b) elemento isoparamétrico

Figura 13 – Formato dos elementos bidimensionais. Fonte: ([LIMA, 2002](#)) .

[Ahmed \(2013\)](#) desenvolveu um trabalho em que utilizou outras formas de elementos além dos quadriláteros como o hexagonal (seis nós) e para nove nós, e comparou-os resultando numa melhor eficiência para o hexagonal (seis nós), no entanto por uma questão de simplificação continuamos a utilizar elementos quadriláteros para o caso bidimensional. Ele também determinou uma ordem em que o MEF deve ser preparado para ser utilizado na sintetização de OT, conforme segue:



- Construção das funções de forma da matriz  $\mathbf{N}$  que satisfaça a função de propriedade Delta e a unidade de Partição.
- Formulação da matriz de deformação de energia  $\mathbf{B}$ .
- Cálculo da matriz de rigidez  $k_e$  usando a função de forma  $\mathbf{N}$  e a matriz de energia  $\mathbf{B}$ .

Lima (2002) descreve de uma maneira objetiva a formulação matemática para o MEF ser discretizado para a OT. Sendo assim começa a definir que a associação dos graus de liberdade, do inglês *degree of freedom* (DOF) de um nó relacionam com os componentes nodais de deslocamento para o MEF de acordo com a equação 4.3.

$$\mathbf{u} = \mathbf{N} \mathbf{v} \quad (\text{Relação DOFs vs Componentes Nodais de Deslocamentos}) \quad (4.3)$$

Em que:

$\mathbf{u}$  : Vetor de componentes dos deslocamentos em cada ponto do elemento

$\mathbf{v}$  : Vetor das componentes dos deslocamentos nodais

$\mathbf{N}$  : Matriz das funções de forma

Também demonstra-se o sistema de coordenadas natural local  $(\xi, \eta)$  que é definido com a localização da origem no centro do elemento conforme demonstrado na figura 14.

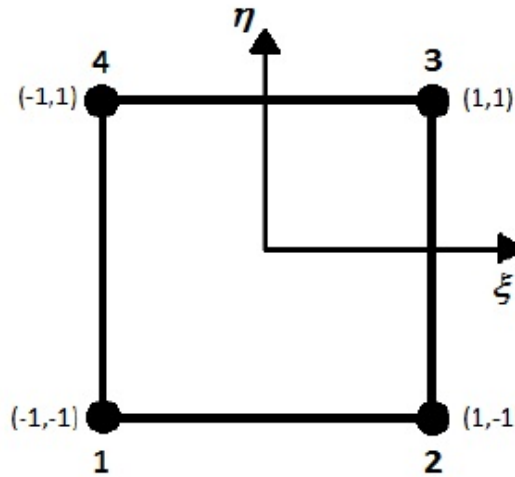


Figura 14 – Elementos finitos quadriláteros nas coordenadas locais  $(\xi, \eta)$ . Fonte: (AHMED, 2013).

A matriz das funções para os elementos quadriláteros é apresentada para os 4 nós como nas equações 4.4, 4.5, 4.6, 4.7.

$$N_1 = \frac{1}{4}(1 - \xi)(1 - \eta) \quad (\text{matriz função nó 1}) \quad (4.4)$$

$$N_2 = \frac{1}{4}(1 + \xi)(1 - \eta) \quad (\text{matriz função nó 2}) \quad (4.5)$$

$$N_3 = \frac{1}{4}(1 + \xi)(1 + \eta) \quad (\text{matriz função nó 3}) \quad (4.6)$$

$$N_4 = \frac{1}{4}(1 - \xi)(1 + \eta) \quad (\text{matriz função nó 4}) \quad (4.7)$$

Em que:

$\xi$  : Coordenada local do elemento na direção horizontal

$\eta$  : Coordenada local do elemento na direção vertical

Assim, de acordo com Lima (2002), as coordenadas dos nós (x,y) da figura 13 b, são obtidas em função das coordenadas nodais do elemento, usando os polinômios lineares das equações 4.4, 4.5, 4.6, 4.7. Precisamos descobrir as aproximações para as componentes horizontais e verticais do deslocamento de cada ponto do domínio do elemento, conforme verificamos nas equações 4.8 e 4.9.

$$x = N_1x_1 + N_2x_2 + N_3x_3 + N_4x_4 \quad (\text{coordenadas de } x) \quad (4.8)$$

$$y = N_1y_1 + N_2y_2 + N_3y_3 + N_4y_4 \quad (\text{coordenadas de } y) \quad (4.9)$$

Para determinar a matriz de função do elemento sólido, devemos entender a energia elástica de deformação ( $U_e$ ), que é descrito na equação 4.10.

$$U_e = \frac{1}{2} \int_{V_e} \epsilon^T \sigma dV_e \quad (\text{Energia Elástica de Deformação}) \quad (4.10)$$

Em que:

$\epsilon^T$  : Vetor da energia de deformação transposto

$\sigma$  : Vetor das tensões

$V_e$  : Volume do elemento

Se o volume é igual a sua área pela altura, e na equação 4.10 o  $V_e$  é integrável, então teremos  $dV_e = h_e dA_e$ , e sabendo que a relação entre a tensão e as deformações com a forma será de acordo com a equação 4.11.

$$\sigma = C \epsilon \quad (\text{relação Tensão e Energia de Deformação}) \quad (4.11)$$

Portanto, de acordo com as equações 4.10 e 4.11, podemos reescrever a energia elástica de deformação em função de uma integral de área de acordo com a equação 4.12.

$$Ue = \frac{1}{2} \int_{A_e} \epsilon^T \mathbf{C} \epsilon h_e dA_e \quad (\text{Energia Elástica em função de área}) \quad (4.12)$$

Em que:

$\mathbf{C}$  : Matriz contendo os parâmetros de elasticidade do material

E de acordo com o estado plano de tensão para elementos bidimensionais a matriz que contém os parâmetros de elasticidade do material do elemento está de acordo com a equação 4.13.

$$\mathbf{C} = \frac{E}{1-v^2} \begin{bmatrix} 1 & v & 0 \\ v & 1 & 0 \\ 0 & 0 & \frac{1-v}{2} \end{bmatrix} \quad (\text{Matriz elast. constitutiva 2D}) \quad (4.13)$$

Em que:

$E$  : módulo de elasticidade.

$v$  : coeficiente de Poisson.

Assim, de acordo com a lei de Hooke e a equação 4.11 e 4.13, podemos escrever para um plano de tensão, assumindo o material ser isotrópico, a relação de acordo com a equação 4.14.

$$\sigma = \begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \end{pmatrix} = C \epsilon = \frac{E}{1-v^2} \begin{bmatrix} 1 & v & 0 \\ v & 1 & 0 \\ 0 & 0 & \frac{1-v}{2} \end{bmatrix} \begin{pmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{xy} \end{pmatrix} \quad (\text{Matriz 2D}) \quad (4.14)$$

Em que:

$\sigma^T = \sigma_{xx}, \sigma_{yy}, \sigma_{zz}$  : São as três componentes do vetor de tensão.

$\epsilon^T = (\epsilon_{xx}, \epsilon_{yy}, \epsilon_{xy})$  : São as três componentes do vetor de deformação e  $\epsilon_{xy}$  é a energia de cisalhamento.

Então as deformações em função dos componentes de deslocamentos nodais do elemento e da matriz de energia tem a forma de acordo com a equação 4.15.

$$\epsilon = \mathbf{B} \mathbf{v} \quad (\text{deformações das componentes dos elementos}) \quad (4.15)$$

Em que:

$\mathbf{B}$  : matriz de operadores diferenciais dos polinômios bilineares, que para o elemento de quatro nós será descrita de acordo com a equação 4.16.

$$\mathbf{B} = \begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial x} & 0 & \frac{\partial N_3}{\partial x} & 0 & \frac{\partial N_4}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial y} & 0 & \frac{\partial N_3}{\partial y} & 0 & \frac{\partial N_4}{\partial y} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial y} & \frac{\partial N_3}{\partial x} & \frac{\partial N_4}{\partial y} & \frac{\partial N_4}{\partial x} \end{bmatrix} \quad (\text{Matriz dos elementos}) \quad (4.16)$$

E da equação 4.15 na equação 4.12, temos a equação 4.17.

$$\begin{aligned} Ue &= \frac{1}{2} \int_{A_e} (\mathbf{B}\mathbf{v})^T \mathbf{C} (\mathbf{B}\mathbf{v}) h_e dA_e \\ Ue &= \frac{1}{2} \mathbf{v}^T \int_{A_e} \mathbf{B}^T \mathbf{C} \mathbf{B} h_e dA_e \mathbf{v} \quad (\text{Energia Elástica função matriz elemento}) \\ Ue &= \frac{1}{2} \mathbf{v}^T \mathbf{k}_e \mathbf{v} \end{aligned} \quad (4.17)$$

Em que:

$\mathbf{k}_e$  : matriz de rigidez do elemento, então o valor da matriz é dimensionado de acordo com a equação 4.18.

$$\mathbf{k}_e = \int_{A_e} \mathbf{B}^T \mathbf{C} \mathbf{B} h_e dA_e \quad (\text{Matriz de Rigidez do Elemento}) \quad (4.18)$$

Será necessário a partir daqui utilizar o Jacobiano como um operador para transformar as coordenadas locais  $(\xi, \eta)$  em coordenadas globais  $(x, y)$ , assim a matriz de rigidez do elemento apresenta a forma de acordo com a equação 4.19.

$$\mathbf{k}_e = \int_{-1}^1 \int_{-1}^1 \mathbf{B}^T \mathbf{C} \mathbf{B} h_e \det \mathbf{J} d\xi d\eta \quad (\text{Matriz de Rigidez El. com Jacobiano}) \quad (4.19)$$

Em que:

$\det \mathbf{J}$  : Determinante da matriz Jacobiana que é descrita de acordo com a equação 4.20.

$$\mathbf{J} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \quad (\text{Matriz Jacobiana}) \quad (4.20)$$

Lima (2002) ainda informa que para elementos que apresentam outras formas que não o quadrado perfeito, não se tem cálculos analíticos obtidos facilmente, portanto para uma forma de quadrilátero qualquer, utiliza-se ainda alguns métodos de integração

numérica, como a Quadratura de Gauss-Legendre. Para os elementos quadriláteros perfeitos o Jacobiano é simplificado, e a integral da equação 4.19 também, assim para estes casos não é necessário utilizar-se os métodos de integração numéricos.

Então, com a matriz de rigidez  $\mathbf{k}_e$  do elemento, é determinada a matriz de rigidez global  $\mathbf{K}$ , que engloba toda a estrutura. Utiliza-se uma matriz de conectividade, denotando um número a cada elemento e também o número de nós da ligação, gerando uma matriz com coordenadas para cada nó, conseguindo desta forma montar a matriz de rigidez global  $\mathbf{K}$ . Na equação 4.21 nós visualizamos portanto, a matriz de rigidez global para domínios bidimensionais.

$$\mathbf{K} = \Sigma \mathbf{k}_e \quad (\text{Matriz de Rigidez Global}) \quad (4.21)$$

A matriz de rigidez global  $\mathbf{K}$  está associada ao vetor de forças  $\mathbf{f}$  conhecidos e aplicado nos nós, assim obtêm-se o vetor de deslocamentos nodais  $\mathbf{v}$  a partir da relação de equilíbrio da equação 4.22.

$$\mathbf{K} \mathbf{v} = \mathbf{f} \quad (\text{Relação de Equilíbrio da Matriz de Rigidez Global}) \quad (4.22)$$

Com esta relação de equilíbrio, que pode ser resolvida com sistema de equações lineares, é aplicado métodos numéricos para a resolução de equações como Gauss-Jordan, Gradientes Conjugados etc., para assim poder encontrar os vetores de deslocamentos nodais  $\mathbf{v}$  da estrutura.

#### 4.2.2 Implementação do MEF para elementos 2D simplificado

No algoritmo de Sigmund (2001) há uma simplificação da matriz de rigidez global pelo fato do autor considerar domínios simples para discretização e que trabalham somente com elementos quadrados com lado unitários iguais, desta forma a matriz de rigidez de cada elemento é igual para todos os elementos. A equação que constitui a matriz de rigidez do elemento conforme descrito na equação 4.19 é simplificada, portanto a matriz de rigidez de todos os elementos assume o valor de acordo com a equação 4.23. O detalhamento da formulação da matriz de rigidez neste caso é específico para este algoritmo ser montado no MATLAB, pois o autor monta a matriz com elementos totalmente quadrados, com lados unitários, portanto iguais, e coloca os elementos todos iguais no domínio. Portanto a formulação da equação da matriz de rigidez do elemento, que é uma integral dupla da matriz constitutiva com a matriz dos elementos bilineares com as posições das coordenadas locais em que visualizamos na equação 4.19, é simplificada para este caso, de forma que o cálculo simplificado seja possível de se resolver por meio de uma expressão analítica

para cada componente implementado dentro da matriz. Neste caso, não é necessário complementar o algoritmo com nenhuma integração numérica, pois ele é resolvido de forma analítica.

$$k_e = \frac{E}{1 - \nu^2} \begin{bmatrix} k(1) & k(2) & k(3) & k(4) & k(5) & k(6) & k(7) & k(8) \\ k(2) & k(1) & k(8) & k(7) & k(6) & k(5) & k(4) & k(3) \\ k(3) & k(8) & k(1) & k(6) & k(7) & k(4) & k(5) & k(2) \\ k(4) & k(7) & k(6) & k(1) & k(8) & k(3) & k(2) & k(5) \\ k(5) & k(6) & k(7) & k(8) & k(1) & k(2) & k(3) & k(4) \\ k(6) & k(5) & k(4) & k(3) & k(2) & k(1) & k(8) & k(7) \\ k(7) & k(4) & k(5) & k(2) & k(3) & k(8) & k(1) & k(6) \\ k(8) & k(3) & k(2) & k(5) & k(4) & k(7) & k(6) & k(1) \end{bmatrix} \quad (\text{Matriz}_e \text{ Sigmund}) \quad (4.23)$$

As componentes  $k(1)$ ,  $k(2)$ ,  $k(3)$ , respectivamente até o  $k(8)$  estão distribuídas dentro da matriz de rigidez da estrutura conforme verifica-se na equação 4.23, e que de acordo com Sigmund (2001) foram distribuídas desta forma, sendo consideradas para uma matriz de rigidez 8 x 8 bi-linear quadrada para 4 nós por elemento, rearranjadas por meio de um *software* de manipulação simbólica.

Além de determinar a matriz de rigidez dos elementos de forma simplificada, Sigmund (2001) determina a montagem no algoritmo da matriz uma vez só por meio da função  $[KE] = lk$  que é chamada em uma linha anterior a função objetivo de compliance, e que posteriormente a iteração volta a ser chamada mantendo todos os elementos. Na implementação do MEF 2D genérico teria que rodar uma vez para todos os elementos, ou seja, cada elemento deve ter a sua matriz, tendo que resolver a integral para cada elemento.

#### 4.2.3 Implementação do MEF para elementos 3D genéricos

Como vemos em Carvalho e Almeida (2013), para estruturas 3D, que são representadas pelo sistema de coordenadas cartesiano xyz, tem-se os vetores das tensões e os vetores das deformações em sua forma transporta de acordo com as equações 4.24 e 4.25.

$$\sigma^T = (\sigma_x, \sigma_y, \sigma_z, \tau_{xy}, \tau_{yz}, \tau_{zx}) \quad (\text{Vetor de tensão 3D}) \quad (4.24)$$

$$\epsilon^T = (\epsilon_x, \epsilon_y, \epsilon_z, \gamma_{xy}, \gamma_{yz}, \gamma_{zx}) \quad (\text{Vetor de deformação 3D}) \quad (4.25)$$

Os valores das tensões normais, em relação a matriz elástica constitutiva 3D  $\mathbf{C}$  e as deformações são apresentados na equação 4.26.

$$\sigma = \begin{pmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{zx} \end{pmatrix} = C \epsilon = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix} \begin{pmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{pmatrix} \quad (4.26)$$

Em que:

$\sigma$  : Tensões normais

$\tau$  : Tensões cisalhantes

$\epsilon$  : Deformações normais

$\gamma$  : Deformações transversais

$\nu$  : coeficiente de Poisson

$E$  : Módulo de Elasticidade

Assim, para encontrarmos a matriz dos elementos associada a função objetivo de OT, e consequentemente a matriz de rigidez global e a equação de equilíbrio, como no elemento bidimensional teremos que encontrar a equação das deformações das componentes dos elementos  $\epsilon = \mathbf{B} \mathbf{v}$ , que constitui estabelecer uma nova matriz de energia, conforme já foi vista na equação 4.15 para elementos bidimensionais, encontrando então a nova matriz dos polinômios bilineares  $\mathbf{B}$ , que para o caso 3D tem a forma de acordo com a equação 4.27.

$$\mathbf{B} = \begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & 0 & \frac{\partial N_2}{\partial y} & 0 & \frac{\partial N_3}{\partial z} \\ 0 & \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial z} & 0 \\ 0 & 0 & \frac{\partial N_1}{\partial z} & 0 & \frac{\partial N_2}{\partial y} & \frac{\partial N_3}{\partial x} \end{bmatrix} \quad (\text{Matriz polinomios bilineares 3D}) \quad (4.27)$$

E substituindo a equação 3.4 da teoria de SIMP na matriz elástica constitutiva 4.26, podemos reescrever a matriz de rigidez do elemento em função de um elemento sólido  $\mathbf{k0}_e$  demonstrados na equação equação 4.28

$$\mathbf{k}_e = \rho_e^P \int_V \mathbf{B}^T \mathbf{E}_0 \mathbf{B} dV = \rho_e^P \mathbf{k0}_e \quad (\text{Matriz de Rig. do El. em função do el. sólido}) \quad (4.28)$$

Sendo assim é possível encontrar para o sistema de coordenadas 3D a equação 4.21 ( $\mathbf{K} = \Sigma \mathbf{k}_e$ ) e a equação 4.22 ( $\mathbf{K} \mathbf{v} = \mathbf{f}$ ) para o caso 3D.

### 4.3 Algoritmos de Otimização no MATLAB®

Para transformar o processo de otimização para uma linguagem computacional afim de ser resolvido com eficiência, tem-se o uso do no programa computacional MATLAB® para o MOT, em que existem funções ou sub-rotinas necessárias para resolver problemas de otimização linear ou não lineares com base nos métodos de aproximação explícita como já vimos na figura 10.

Lima (2002) descreve que um algoritmo numérico de otimização é usado para encontrar de forma iterativa a distribuição ótima de material. Existem vários tipos de algoritmos de otimização na literatura, sejam eles adotando métodos de otimização pelo critério de um método específico, ou por métodos genéricos como já vimos.

Messac (2015) define que o MATLAB® (abreviação do termo em inglês *Matrix Laboratory*) é um programa ou ferramenta computacional para resolver problemas de otimização numéricos, tais como problemas com restrição e sem restrição, e linear e não lineares, por meio de sua biblioteca com todas as funções específicas. Uma das funções lineares por exemplo é a chamada Lingprog, esta função baseia-se no método Simplex para a resolução de problemas de pequeno porte já que posteriormente foram desenvolvidas outras funções de grande porte e que diminuem o tempo computacional da sintetização de OT. Esta rotina por exemplo, foi desenvolvida para auxiliar os programadores no cálculo de programação linear, pois é um programa robusto e amplamente utilizado na resolução de problemas de engenharia, conforme descrito por Martins (2011).

#### 4.3.1 Rotinas de Otimização no MATLAB®

Além da função Lingprog, o programa MATLAB® tem uma biblioteca com diversas funções de otimização, conforme figura 15.

#### 4.3.2 História dos algoritmos escritos em MATLAB® e outros

Segundo Liu e Tovar (2013), com o advento da OT alguns códigos foram escritos com o uso do software MATLAB® ou outros. Estes códigos são descritos pela quantidade de linhas e pela composição, pela característica unidimensional (1D), bidimensional (2D) ou tridimensional (3D).

#### 4.3.3 Algoritmos bidimensionais - 2D

Em Sigmund (2001) introduziu-se o código para otimização topológica em duas dimensões (2D) para MATLAB®. Este programa usa montagem de matriz de rigidez e filtragem por meio de loops aninhados, o que torna o código legível e bem organizado, mas também torna lento na resolução de problemas maiores, neste programa foi apresentado um



Comando	Descrição
<code>bintprog</code>	- Problemas de programação inteira de Binário
<code>fgoalattain</code>	- Problemas de realização de metas multiobjetivos
<code>fminbnd</code>	- Mínimo de função variável única em intervalo fixo
<code>fmincon</code>	- Mínimo de função multivariável não linear restrita
<code>fminimax</code>	- Problemas de restrição Minimax
<code>fminsearch</code>	- Mínimo de função multivariável sem restrição utilizando método da livre derivada
<code>fminunc</code>	- Mínimo de função multivariável sem restrição
<code>fseminf</code>	- Mínimo de função não-linear multivariável restrita semi-infinita
<code>ktrlink</code>	- Mínimo de função multivariável com e sem restrição não linear utilizando KNITRO
<code>linprog</code>	- Problemas de Programação Linear
<code>quadprog</code>	- Problemas de Programação Quadrática

Figura 15 – Funções de otimização no MATLAB® Fonte: Adaptado de [Messac \(2015\)](#).

código com 99 linhas. Este é um código tido como básico e simples de ser implementado, mas primordial para o meio acadêmico e que vêm incentivando vários desenvolvimentos e aplicações a partir dele.

Em [Wang, Chen e Xia \(2004\)](#) criou-se o programa *TOPLSM* com 199 linhas de *MATLAB*® para a otimização da conformidade de elementos 2D com o uso do LS. Esta otimização inclui esquema Upwind para resolver a PDE de Hamilton-Jacobi, reinicialização baseada em PDE, cálculo de quantidades geométricas e análise de elementos finitos em 2D.

Em [Challis e Guest \(2009\)](#) usou-se o LS com variáveis discretas para 129 linhas, este foi um dos principais algoritmos desenvolvidos para a abordagem LS para um objetivo único.

Em [Suresh \(2010\)](#) utilizou-se o programa *ParetoOptimalTracing* que rastreia as fronteiras de Pareto para diferentes frações de volume usando sensibilidade topológica por meio do LS. Este desenvolvimento baseou-se em utilizar mais de uma função de objetivo, ou seja, trabalhou com funções multi-objetivos, por exemplo o objetivo de minimizar o compliance e o objetivo de maximizar o deslocamento.

Em [Andreassen et al. \(2011\)](#), utilizou-se um código para *MATLAB*®, com base em [Sigmund \(2001\)](#), que reduzia as 99 linhas anteriormente para 88 linhas, enquanto o código de 99 linhas apresentou apenas o filtro de sensibilidade, o código de 88 linhas

apresentou o filtro de densidade, proporcionando uma maior suavidade nos limites das variáveis de pseudo-densidades do domínio do projeto. [Liu e Tovar \(2013\)](#) descreve que foi comparado os dois códigos de 88 linhas e de 99 linhas com uma estrutura de 7500 elementos e a versão atualizada de 88 linhas é 2X mais rápida em relação a pioneira

#### 4.3.4 Algoritmos tridimensionais - 3D

[Zhou e Wang \(2005\)](#), propuseram a criação de um algoritmo para estruturas em 3D feitas no MATLAB® que continha 177 linhas, baseado no algoritmo 2D de [Sigmund \(2001\)](#).

Em [Liu e Tovar \(2013\)](#), descreve-se um código para implementação de otimização topológica para três dimensões (3D) que possui 169 linhas. Nestas linhas estão inclusos a análise por elementos finitos (MEF), a análise sensibilidade, o filtro de densidade e o critério de otimalidade, e também é demonstrado os resultados. O código é descrito para resolver o problema de minimização da flexibilidade média da estrutura.

#### 4.3.5 Outros algoritmos

Em [Liu, Korvink e Huang \(2005\)](#) para *FemLab*.

Em [Allaire e Pantz \(2006\)](#) para o *FreeFem++* com a otimização de forma utilizando a abordagem da homogeneização e o método da variação do contorno baseado em derivada.

Em [Hunter \(2009\)](#) para o *ToPy* utilizando linguagem *Python* em código aberto.

Em [Sokół \(2011\)](#) para estruturas de treliças do tipo Michael utilizando o *Mathematica*.

Em [Aage et al. \(2013\)](#), foi introduzido um aplicativo denominado *TopOpt* para dispositivos portáteis.

### 4.4 Alguns formatos de domínio para algoritmos usuais na literatura de OT

Desde o desenvolvimento do início da OT com [Bendsøe e Kikuchi \(1988\)](#) e conforme vemos em [Ahmed \(2013\)](#), algumas formas de domínio são estudadas em OT para dar ao leitor uma equivalência de dados e interpretação de resultados de forma simples e objetiva, os seguintes exemplos são encontrados na literatura, e que aparecem nos diversos artigos de OT encontrados atualmente:

- Viga MBB simétrica, do inglês *MBB (Messerschmitt-Bolkow-Blohm) beam*.
- Viga curta em balanço, do inglês *Short Cantilever Beam*.

- Viga em balanço com furo fixado, do inglês *cantilever beam with a fixed hole*.
- Pontes, do inglês *bridge*.
- Etc.

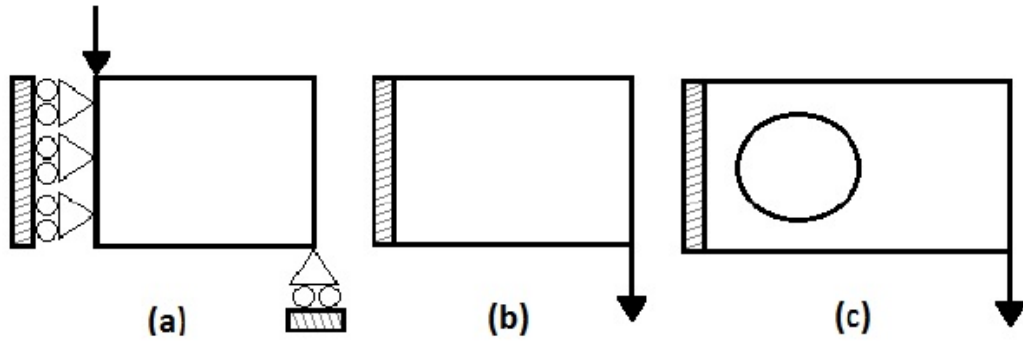


Figura 16 – Sintetizações com CO. (a)Viga MBB Simétrica, (b)Viga curta em balanço, (c)Viga em balanço com furo fixado. Fonte: (AHMED, 2013).

Em cada um destes itens é possível avaliar a OT por critérios de otimização de CO ou de MMA. Por exemplo, na figura 16 tem-se a diferença da aplicação das cargas e dos pontos fixos para alguns destes casos, e na figura 17 tem-se os objetos sintetizados por OT segundo estes casos.

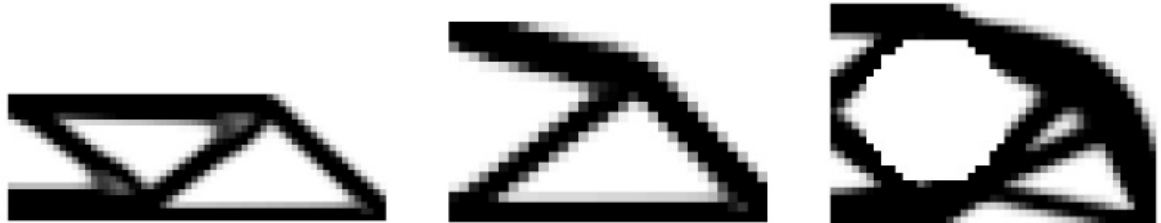


Figura 17 – (a)Viga MBB Simétrica, (b)Viga curta em balanço, (c)Viga em balanço com furo fixado. Fonte: (AHMED, 2013).

São estes portanto, alguns dos domínios de projetos encontrados nos algoritmos de otimização desenvolvidos como por exemplo em Sigmund (2001) etc. No trabalho proposto, uma das dificuldades encontradas foi definir um domínio e condições de contorno que se aproximassem da estrutura de uma longarina de chassi veicular. Um dos problemas encontrados é a relação de aspecto do inglês *aspect ratio*, porque, quanto maior é a relação, mais problemas podem ser encontrados na discretização do domínio da estrutura. Na literatura alguns trabalhos tem sido desenvolvidos atualmente discutindo a relação de aspecto de algumas aplicações como observamos em Seow (2016), Diogo (2016) e Zhao (2016), no entanto já é algo que vinha sendo discutido durante os desenvolvimentos nas ultimas décadas de otimização como verificamos em Bendsøe e Sigmund (2003).

Seow (2016) discute a alta relação de aspecto das asas de aeronaves e utiliza a abordagem de LS para a proposta. Naquele trabalho ele propõe relações de aspecto de 18,20

e 24:1, com minimização de compliance e restrição de volume e concluí que o aspecto de 24:1 foi o melhor para o estudo. Já Zhao (2016) define uma nova subestrutura baseada no método de otimização topológica para o projeto da estrutura de asas em que ele compara a topologia tradicional e a formulação proposta, sendo que na formulação proposta ele corrige a discretização do domínio para que as regiões sólidas sejam mais contínuas, de acordo com que vemos na figura 18. Naquele problema utiliza-se uma viga I de relação de aspecto 20:1.

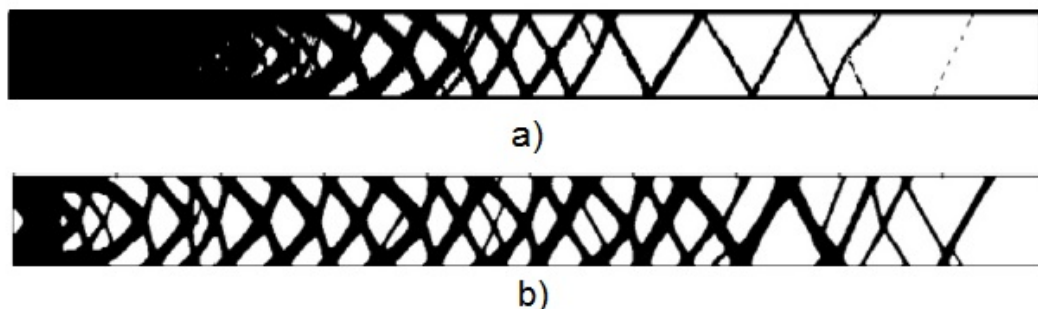


Figura 18 – (a)topologia tradicional, (b)formulação proposta. Fonte: Adaptado de Zhao (2016).

Já Diogo (2016), observa que cada vez mais é importante ter altas relações de aspecto para asas de aeronaves, pois facilita a diminuição do peso das aeronaves, resultando em menos consumo de combustível, porém informa que com grandes relações de aspectos há problemas estruturais que podem surgir e que é necessário considerar uma *wing box* suficientemente rígida, mais tem-se problemas pois aumentará o peso da asa e inviabilizará o processo de otimização para esta aplicação com esta alta relação de aspecto. Assim ele propõe utilizar um algoritmo para minimizar a menor wing box com a maior relação de aspecto tornando-o um problema de otimização. Diogo (2016) ainda demonstra algumas relações de aspecto para asas de aeronaves existentes que variam de 2,4 até 12,8:1.

O estudo de Diogo (2016), baseia-se em asa de aeronaves, enquanto o estudo do objeto deste trabalho é uma longarina utilizada na indústria automotiva, são aplicações distintas, porém há uma questão que é paralela e que poderá ser igual para os dois casos, que seria em qual relação de aspecto, o projeto de asas ou o projeto de uma longarina é mais eficiente. Há uma razão portanto de ser verificado para qual relação de aspecto o domínio do estudo deste projeto pode ser confiável.

Na proposta deste trabalho temos uma longarina de chassi veicular com a relação de aspecto de aproximadamente 26:1. Portanto observa-se que é uma relação alta e que pode ser comparadas a aplicação de asas de uma aeronave. No entanto na literatura ainda não há desenvolvimentos que comparem diversas longarinas de chassi veicular do mercado e que demonstrem se as relações de aspectos são adequadas. Assim um estudo de otimização paramétrica poderia ser valido para um outro trabalho para avaliar as condições de relação de aspecto do objeto proposto (longarina do chassi veicular).

Para o trabalho, o autor implementará um algoritmo e testará algumas condições de relação de aspecto, em que observará se o aumento desta relação influenciará em distorções ou descontinuidades na estrutura com base na sintetização da OT, e estas relações serão discutidas nos próximos capítulos.

## 4.5 Algoritmo de Otimização Topológica baseado em SIMP - 2D

Foi utilizado como base para o desenvolvimento do novo código proposto, o algoritmo baseado em [Sigmund \(2001\)](#) que possui 99 linhas e que foi um dos primeiros códigos desenvolvidos para o MOT para sintetizar domínios bidimensionais (2D). Este algoritmo não é um método de fácil e de rápida convergência, mas é simples de implementação, e no uso acadêmico mais de 80% dos trabalhos publicados desde as últimas 3 décadas usam o SIMP. Por este fato adotou-se como o primeiro algoritmo para a tratativa do objeto de estudo, apesar de saber que atualmente existem outros algoritmos mais eficientes para projetos de grande porte. Sendo assim, verifica-se que para a aplicação do algoritmo proposto, o algoritmo baseado em [Sigmund \(2001\)](#) é muito limitado, serve apenas para conceitual, no entanto para os primeiros resultados para solucionar o objetivo deste trabalho que é o de encontrar nova topologia para o chassi, os conceitos serão base para outras implementações, assim é factível a importância da implementação deste algoritmo. Outras implementações como o uso do método de otimização MMA podem ser testadas para melhorar a convergência do algoritmo. Este algoritmo trabalha com uma malha de EF de elementos quadrados simplificados, porém outras formas dos elementos podem ser testadas modificando a matriz de chamada dos EF.

Segundo [Sigmund \(2001\)](#) as linhas de comando do código são subdivididas por grupos de processos para facilitar o entendimento. Nas primeiras 36 linhas temos o programa principal, em que tem-se o processo de otimização, em outras 12 linhas há o método de otimização baseado na função de atualização da abordagem SIMP, outras 16 linhas para o filtro, fazendo com que o código se torne independente de malha, e as últimas 35 linhas com a definição dos elementos finitos. O código de [Sigmund \(2001\)](#) definiu a função principal para ser chamada no *prompt* de comando do MATLAB®, de acordo com [4.29](#).

$$\text{topopt}(\text{nelx}, \text{nely}, \text{volfrac}, \text{penal}, \text{rmin}) \quad (4.29)$$

Em que:

- nelx e nely é o número de elementos na horizontal e na vertical respectivamente.
- volfrac é a fração de volume desejada na sintetização.
- penal é a penalização p, que pode variar de 1 a 3 conforme vimos anteriormente.

-  $r_{min}$  é raio mínimo do filtro (que é dividido pelo tamanho do elemento).

Como condições de contorno devem ser reduzidas para melhorar o algoritmo, Sigmund (2001) propôs a divisão de uma viga simplesmente apoiada, conforme figura 19.

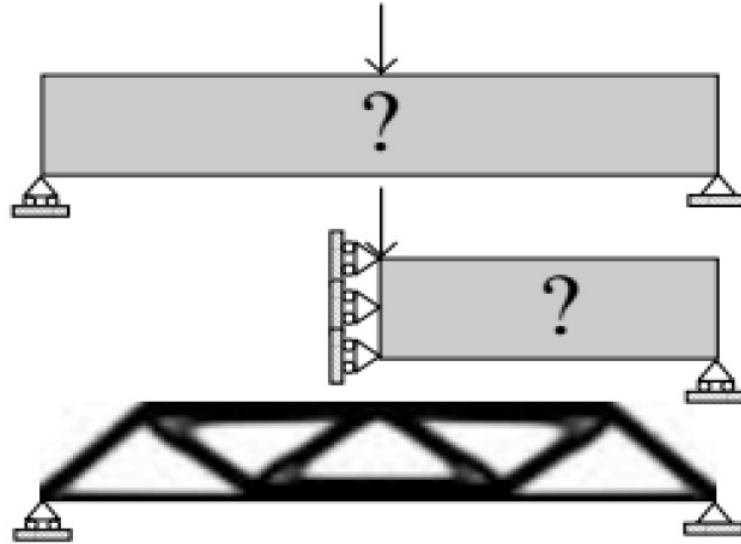


Figura 19 – Tipos de domínio do projeto para a viga MBB. Fonte: (BENDSØE; SIGMUND, 2003).

#### 4.5.1 Programa principal (Linha 1 a 36)

No código de Sigmund (2001), tem-se a (linha 1) que tem as informações sobre a origem do código, já na (linha 2) tem-se a descrição da função principal do código conforme a equação 4.29, na (linha 4) é declarado a posição em que as pseudo-densidades  $x$  podem atuar ao longo do domínio, que deverá ser igual ao volume fracionado, nas próximas linhas começa-se os comandos de inicialização que vão até a (linha 10), na (linha 12) temos uma sub-rotina que chama o elementos finitos para retornar os valores de deslocamentos de  $\mathbf{U}$ . Na (linha 14) temos uma sub-rotina que chama a matriz de rigidez do elemento, e na (linha 15) temos a inicialização da variável ( $c = 0$ ), que é a função objetivo e que atualiza a flexibilidade (compliance). Nas (linhas 16 a 24), temos a varredura dos elementos e o cálculo da função objetivo e dos gradientes, e nas (linhas 18 a 19) tem-se as variáveis  $n_1$  e  $n_2$ , que caracterizam os números de nós superiores a esquerda e a direita do elemento das coordenadas globais, e que são variáveis que mapeiam o deslocamento  $\mathbf{ue}$  do sistema de coordenadas globais para o sistema de coordenadas locais. Na (linha 28) temos a função de atualização de densidades CO, nas (linhas 30 a 33) temos a demonstração dos resultados da função objetivo, e outros parâmetros anteriormente calculados. Na (linha 35) temos o resultado da distribuição das densidades. Como critério de parada, adota-se que o ciclo principal será finalizado, quando a mudanças nas variáveis for menor do que 1%, caso seja



maior volta na (linha 30) para repetir as etapas. Todos estes passos são possíveis de ser visualizados na figura 20.

```

1  %%%% CÓDIGO 99 LINHAS BASEADO EM SIGMUND, 2001. %%%%
2  function toptopt(nelx,nely,volfrac,penal,rmin)
3  % INICIALIZAÇÃO
4  x(1:nely,1:nelx) = volfrac;
5  loop = 0;
6  change = 1.;
7  % INÍCIO DAS ITERAÇÕES
8  while change > 0.01
9      loop = loop + 1;
10     xold = x;
11     % ANÁLISE FEA
12     [U]=FE(nelx,nely,x,penal);
13     % FUNÇÃO OBJETIVA E ANÁLISE DE SENSIBILIDADE
14     [KE] = lk;
15     c = 0.;
16     for ely = 1:nely
17         for elx = 1:nelx
18             n1 = (nely+1)*(elx-1)+ely;
19             n2 = (nely+1)* elx +ely;
20             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);
21             c = c + x(ely,elx)^penal*Ue'*KE*Ue;
22             dc(ely,elx) = -penal*x(ely,elx)^(penal-1)*Ue'*KE*Ue;
23         end
24     end
25     % FILTRO DE SENSIBILIDADE
26     [dc] = check(nelx,nely,rmin,x,dc);
27     % ATUALIZAÇÃO DAS DENSIDADES
28     [x] = OC(nelx,nely,x,volfrac,dc);
29     % MOSTRA OS RESULTADOS
30     change = max(max(abs(x-xold)));
31     disp([' It.: ' sprintf('%4i',loop) ' Obj.: ' sprintf('%10.4f',c) ...
32           ' Vol.: ' sprintf('%6.3f',sum(sum(x))/(nelx*nely)) ...
33           ' ch.: ' sprintf('%6.3f',change )])
34     % PLOTA A FIGURA EM ESCALA DE CINZA
35     colormap(gray); imagesc(-x); axis equal; axis tight; axis off; pause(1e-6)
36 end

```

Figura 20 – Programa principal. Fonte: (SIGMUND, 2001).

#### 4.5.2 Atualização das densidades baseado no CO (Linhas 37 a 48)

Neste próximo bloco, o código é definido para atualizar o valor das pseudo-densidades  $x$ , baseados no critério de otimalidade. Aqui há uma restrição lateral que é a restrição de volume, mas de modo a simplificar o critério o método de biseção é adotado independente do número de restrições (linhas 40 a 48). No método de biseção há a inicialização dos multiplicadores de Lagrange de  $l_1$  pequeno e de um valor de  $l_2$  alto (linha 39), neste método temos um intervalo definido para o multiplicador de Lagrange, que é repetidamente reduzido à metade, até que o tamanho estabeleça o critério de convergência estabelecido na (linha 40), figura 21.

```

37      %***** ATUALIZAÇÃO CRITÉRIO DE OPTIMALIDADE %*****
38      function [xnew]=OC(nelx,nely,x,volfrac,dc)
39      l1 = 0; l2 = 100000; move = 0.2;
40      while (l2-l1 > 1e-4)
41          lmid = 0.5*(l2+l1);
42          xnew = max(0.001,max(x-move,min(1.,min(x+move,x.*sqrt(-dc./lmid)))));
43          if sum(sum(xnew)) - volfrac*nelx*nely > 0;
44              l1 = lmid;
45          else
46              l2 = lmid;
47          end
48      end

```

Figura 21 – Rotina CO. Fonte: (SIGMUND, 2001).

#### 4.5.3 Filtro de independência de malha (Linhas 49 a 60)

Conforme já vimos no capítulo anterior 3.3, existem problemas de discretização para serem resolvidos nos algoritmos, um destes problemas é a dependência de malhas, assim Sigmund (2001) colocou no algoritmo um filtro de independência de malha, em que troca o gradiente que guia a otimização por uma média ponderada dos gradientes dos elementos finitos vizinhos dentro de um raio chamado de  $r_{min}$ , assim o peso do elemento depende da distância entre dois elementos, quanto maior a distância, menor o peso. O filtro implementado tem a equação de acordo com o que foi descrito no capítulo 3.3 e nas equações 3.9 e 3.10.

Nas linhas 25 e 26 da figura 19, tem-se a função *check* para recalculer o gradiente da função objetivo em relação as densidades, aplicando o filtro no gradiente, e a implementação das equações 3.9 e 3.10 na forma da função *check*, que está demonstrado nas (linhas 49 a 64), conforme figura 22.

```

49      %***** FILTRO DE INDEPENDÊNCIA DE MALHA %*****
50      function [dcn]=check(nelx,nely,rmin,x,dc)
51      dcn=zeros(nely,nelx);
52      for i = 1:nelx
53          for j = 1:nely
54              sum=0.0;
55              for k = max(i-floor(rmin),1):min(i+floor(rmin),nelx)
56                  for l = max(j-floor(rmin),1):min(j+floor(rmin),nely)
57                      fac = rmin-sqrt((i-k)^2+(j-l)^2);
58                      sum = sum+max(0,fac);
59                      dcn(j,i) = dcn(j,i) + max(0,fac)*x(l,k)*dc(l,k);
60                  end
61              end
62              dcn(j,i) = dcn(j,i)/(x(j,i)*sum);
63          end
64      end

```

Figura 22 – Programa do Filtro de independência de malha. Fonte: (SIGMUND, 2001).

O elemento quadrilátero tem que ter o lado de comprimento duas vezes maior que o  $r_{min}$ , quando seleciona-se  $r_{min}$  menor do que 1 na função *check*, a filtragem não ocorre.



#### 4.5.4 Programa do MEF (Linhas 65 a 99)

Nas (linhas 65 a 99) temos todo o código dos elementos finitos, que é um tema a parte da otimização topológica, temos neste bloco, a definição das condições de contorno do domínio (linha 65 a 85) e a montagem da matriz de rigidez global (linha 86 a 99), encontramos aqui uma função do MATLAB® chamada *sparse*, que inicia a matriz de rigidez global  $K$  e o vetor das forças externas  $F$ . Nas (linhas 70 a 77), temos um laço duplo que varre os elementos da matriz de rigidez global. Nas (linhas 78 a 82) temos as definições de carregamento por meio do vetor  $F$ , que é definido por uma coordenada global (2,1) da viga, e tem-se as restrições de deslocamento, representando os apoios da viga. A carga do vetor força -1 significa que a força exerce um sentido vertical com direção para baixo. Os nós e elementos são numerados da coluna da esquerda para a direita e de cima para baixo, e cada nó tem dois graus de liberdade (horizontal e vertical). Na figura 23 é possível ver a distribuição dos elementos em uma matriz em que o elemento 1 por exemplo tem um nó superior esquerdo correspondente ao número 1 para o DOF na horizontal e número 2 para o DOF na vertical.

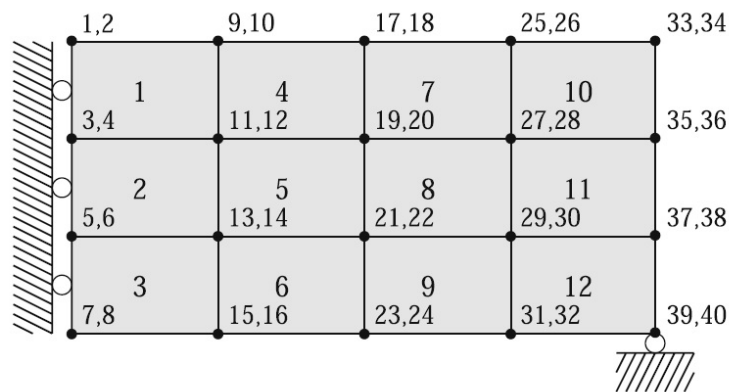


Figura 23 – Distribuição dos elementos. Fonte: (ANDREASSEN et al., 2011).

Assim o comando  $F(2,1) = -1$  na linha 79, aplica uma força unitária vertical no canto superior esquerdo, posição 2 na linha 1. Para os apoios temos a eliminação das coordenadas fixas para poder resolver um sistema linear que são descritos nas linhas (83 a 85), conforme figura 24, em que:

- *freedofs*: São as coordenadas livres
- *fixeddofs*: São as coordenadas fixas

Nas (linhas 86 a 99) para finalizar o algoritmo tem-se a matriz de rigidez do elemento, descritas nas coordenadas locais, transcrevendo os coeficientes do elemento quadrilátero, assim tem-se a matriz de rigidez  $8 \times 8$  do elemento quadrilátero de 4 nós, também é colocada o módulo de Young e o coeficiente de Poisson nas (linhas 88 e 89) respectivamente para definir as propriedades do material, conforme figura 24.

```

65  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% ANÁLISE DO MEF %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
66  function [U]=FE(nelx,nely,x,penal)
67  [KE] = lk;
68  K = sparse(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1));
69  F = sparse(2*(nely+1)*(nelx+1),1); U = zeros(2*(nely+1)*(nelx+1),1);
70  for elx = 1:nelx
71      for ely = 1:nely
72          n1 = (nely+1)*(elx-1)+ely;
73          n2 = (nely+1)* elx +ely;
74          edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1; 2*n2+2; 2*n1+1; 2*n1+2];
75          K(edof,edof) = K(edof,edof) + x(ely,elx)^penal*KE;
76      end
77  end
78  % DEFINIÇÃO DE CARGAS E SUPORTES (CASO DA VIGA MBB-BEAM DIVIDIDA)
79  F(2,1) = -1;
80  fixeddofs = union([1:2*2*(nely+1)], [2*(nelx+1)*(nely+1)]);
81  alldofs = [1:2*(nely+1)*(nelx+1)];
82  freedofs = setdiff(alldofs,fixeddofs);
83  % CONDIÇÃO DOS PONTOS DE FIXAÇÃO
84  U(freedofs,:) = K(freedofs,freedofs) \ F(freedofs,:);
85  U(fixeddofs,:) = 0;
86  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% MATRIZ DE RIGIDEZ DO ELEMENTO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
87  function [KE]=lk
88  E = 1.;
89  nu = 0.3;
90  k=[ 1/2-nu/6    1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 ...
91      -1/4+nu/12 -1/8-nu/8  nu/6    1/8-3*nu/8];
92  KE = E/(1-nu^2)*[ k(1) k(2) k(3) k(4) k(5) k(6) k(7) k(8)
93                  k(2) k(1) k(8) k(7) k(6) k(5) k(4) k(3)
94                  k(3) k(8) k(1) k(6) k(7) k(4) k(5) k(2)
95                  k(4) k(7) k(6) k(1) k(8) k(3) k(2) k(5)
96                  k(5) k(6) k(7) k(8) k(1) k(2) k(3) k(4)
97                  k(6) k(5) k(4) k(3) k(2) k(1) k(8) k(7)
98                  k(7) k(4) k(5) k(2) k(3) k(8) k(1) k(6)
99                  k(8) k(3) k(2) k(5) k(4) k(7) k(6) k(1)];

```

Figura 24 – Código para o MEF. Fonte: (SIGMUND, 2001).

Sigmund (2001) propôs inicialmente o estudo de caso para uma viga MBB que por uma simplificação do código foi apresentada de forma simétrica, em que houve a necessidade de cálculo de apenas um dos lados, ver figura 19. No algoritmo proposto discutido nos próximos capítulos a característica da longarina do chassi automotiva não é simétrica, então algumas alterações deverão ser feitas no código original, em que devemos reposicionar os pontos de fixação, e adotar força distribuída ao longo da viga para termos uma similaridade com as características de uma longarina de chassi de um caminhão comercial.

## 4.6 Algoritmo de Otimização Topológica baseado no LS - 2D

O código apresentado por Challis e Guest (2009), diferentemente de Sigmund (2001), baseia-se no LS com variáveis discretas, e adéqua a função de otimização por meio da sensibilidade topológica e com a função de atualização de LS. Como um dos programas computacionais utilizados neste trabalho que é o ParetoWorks que provém de estudos baseando em Challis e Guest (2009) e Suresh (2010), descreveremos a partir daqui o algoritmo bidimensional baseado em LS.

O código de [Challis e Guest \(2009\)](#) é dividido por grupos, totalizando 199 linhas. Na tabela 9 por meio de [Challis e Guest \(2009\)](#), é verificado os dados de entrada do código e posteriormente podemos descrever o código em cada linha ou grupo.

Nesta tabela 9, podemos observar que diferentemente do SIMP, existem 3 parâmetros adicionais que são utilizados que é o *StepLength*, o *numReinit* e o *topWeight*.

Tabela 9 – Descrição de Parâmetros de Entrada

Variáveis	Descrição	Entrada
nelx (adm.)	Número de elementos na direção horizontal	60
nely (adm.)	Número de elementos na direção vertical	30
volReq	É a fração sólida do volume requerido	0,3
StepLength	Especifica o intervalo de tempo sobre a função de LS	3
numReinit	Frequência em que a função LS é reinicializada	2
topWeight	O parâmetro de peso	2

#### 4.6.1 Programa Principal - Linhas 1-48

Nestas linhas do programa principal são incluídos a inicialização, o loop de iteração para convergir a otimização e a checagem da convergência, conforme figura 25. A inicialização começa com a definição da fase de tamanho de *nelx* e *nely*, de valores de  $x_e$  correntes da estrutura para que sejam sólidas (linha 4). Na linha 5 temos a reinicialização da função LS (lsf) por meio da função *reinit*. Na linha 6 temos os parâmetros de forma (*shaleSens*) e de sensibilidade topológica (*topSens*) inicializados. Na linha 7 temos a matriz de rigidez e outros parâmetros de materiais identificados como o chamado *materialInfo*. Na linha 9 são definidos o número máximo de iterações, que no caso proposto por [Challis e Guest \(2009\)](#) é de 200 mas que pode ser alterado pelo usuário. Na linha 11 temos a chamada da rotina de FEA dentro do loop de iteração, e critérios que garantem o cálculo da forma e da sensibilidade topológica da estrutura (linhas 12 à 21). A linha 17 calcula a sensibilidade de forma do compliance e a linha 18 e 19 calcula a sensibilidade topológica do compliance conforme suas equações detalhadas em [Challis e Guest \(2009\)](#). O objetivo de compliance é calculado da sensibilidade de forma e guardado na linha 23, a fração de volume sólido é calculada na linha 24. Nas linhas 25 à 27 temos a informação do objetivo e do volume extraídas para uma figura. Nas linhas 28 a 32 temos a checagem de convergência, em que é verificado o número de iterações, e é garantido depois de 5 iterações se o volume está há 0,005 do valor de volReq, estando dentro de uma tolerância de 1% do valor de objetivo. Assim, se o algoritmo não terminar, uma atualização é feita de acordo com o Lagrangiano Aumentado (linhas 33 à 38). As sensibilidade de termos adicionais do lagrangiano são adicionadas para a sensibilidade da função objetiva (linhas 39 à 41), e na linha 43 temos a função de projeto de atualização chamada *updateStep*. Então as linhas 44 à 47 trabalham com a reinicialização da função de LS, repetindo assim todos os resultados.

```

1  %% TOPOLOGY OPTIMIZATION USING THE LEVEL-SET METHOD, VIVIEN J. CHALLIS 2009
2  function [struc] = top_levelset(nelx,nely,volReq,stepLength,numReinit,topWeight)
3  % Initialization
4  struc = ones(nely,nelx);
5  [lsf] = reinit(struc);
6  shapeSens = zeros(nely,nelx); topSens = zeros(nely,nelx);
7  [KE,KTr,lambda,mu] = materialInfo();
8  % Main loop:
9  for iterNum = 1:200
10 % FE-analysis, calculate sensitivities
11 [U] = FE(struc,KE);
12 for ely = 1:nely
13 for elx = 1:nelx
14 n1 = (nely+1)*(elx-1)+ely;
15 n2 = (nely+1)* elx +ely;
16 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);
17 shapeSens(ely,elx) = -max(struc(ely,elx),0.0001)*Ue'*KE*Ue;
18 topSens(ely,elx) = struc(ely,elx)*pi/2*(lambda+2*mu)/mu/(lambda+mu)* ...
19 (4*mu*Ue'*KE*Ue+(lambda-mu)*Ue'*KTr*Ue);
20 end
21 end
22 % Store data, print & plot information
23 objective(iterNum) = -sum(shapeSens(:));
24 volCurr = sum(struc(:))/(nelx*nely);
25 disp([' It.: ' num2str(iterNum) ' Compl.: ' sprintf('%10.4f',objective(iterNum)) ...
26 ' Vol.: ' sprintf('%6.3f',volCurr)])
27 colormap(gray); imagesc(-struc,[-1,0]); axis equal; axis tight; axis off; drawnow;
28 % Check for convergence
29 if iterNum > 5 && ( abs(volCurr-volReq) < 0.005 ) && ...
30 all( abs(objective(end)-objective(end-5:end-1)) < 0.01*abs(objective(end)) )
31 return;
32 end
33 % Set augmented Lagrangian parameters
34 if iterNum == 1
35 la = -0.01; La = 1000; alpha = 0.9;
36 else
37 la = la - 1/La * (volCurr - volReq); La = alpha * La;
38 end
39 % Include volume sensitivities
40 shapeSens = shapeSens - la + 1/La*(volCurr-volReq);
41 topSens = topSens + pi*(la - 1/La*(volCurr-volReq));
42 % Design update
43 [struc,lsf] = updateStep(lsf,shapeSens,topSens,stepLength,topWeight);
44 % Reinitialize level-set function
45 if ~mod(iterNum,numReinit)
46 [lsf] = reinit(struc);
47 end
48 end

```

Figura 25 – Programa Principal para o LS. Fonte: (CHALLIS; GUEST, 2009).

#### 4.6.2 Reinicialização da função de LS - Linhas 49-53

Segundo Challis e Guest (2009) etc., a reinicialização da função de LS é importante para garantir a resolução da equação de evolução, e a função de nucleação que determina a inserção de furos na estrutura que foi dada na equação 3.8, verifica-se que não necessariamente seja reinicializado frequentemente a função de LS. Desta forma existe um controle para isto denominado no algoritmo como *numReinit* denominado na linha 50 com a função *Reinit*, conforme vemos na figura 26.

```

49 %%---- REINITIALIZATION OF LEVEL-SET FUNCTION ----
50 function [lsf] = reinit(struc)
51 strucFull = zeros(size(struc)+2); strucFull(2:end-1,2:end-1) = struc;
52 % Use "bwdist" (Image Processing Toolbox)
53 lsf = (~strucFull).*(bwdist(strucFull)-0.5) - strucFull.*(bwdist(strucFull)-0.5);

```

Figura 26 – Reinicialização da função LS. Fonte: (CHALLIS; GUEST, 2009).



### 4.6.3 Atualização do domínio de projeto de LS - Linhas 54-63

Neste grupo de linhas temos a função *updateStep* já vistas na linha 43, porém aqui aparece como uma atualização de domínio de projeto por meio de 3 etapas. A primeira pela suavidade da sensibilidade que evita mudança bruscas entre os elementos vizinhos (linhas 56 à 58), a segunda consiste em condicionar a sensibilidade para zero (linhas 59 à 61) e a terceira consiste em gerenciar a função de LS para encontrar novas estruturas utilizando a função *evolve* (linhas 62 à 63), conforme vemos na figura 27.

```

54      %%----- DESIGN UPDATE -----
55      function [struc,lsf] = updateStep(lsf,shapeSens,topSens,stepLength,topWeight)
56      % Smooth the sensitivities
57      [shapeSens] = conv2(padarray(shapeSens,[1,1],'replicate'),1/6*[0 1 0; 1 2 1; 0 1 0],'valid');
58      [topSens] = conv2(padarray(topSens,[1,1],'replicate'),1/6*[0 1 0; 1 2 1; 0 1 0],'valid');
59      % Load bearing pixels must remain solid - Bridge:
60      shapeSens(end,[1,round(end/2):round(end/2+1),end]) = 0;
61      topSens(end,[1,round(end/2):round(end/2+1),end]) = 0;
62      % Design update via evolution
63      [struc,lsf] = evolve(-shapeSens,topSens.*(lsf(2:end-1,2:end-1)<0),lsf,stepLength,topWeight);

```

Figura 27 – Atualização do domínio de projeto de LS. Fonte: (CHALLIS; GUEST, 2009).

### 4.6.4 Evolução da função de LS - Linhas 64-86

Neste grupo são definidos os parâmetros de evolução da função de LS que resolve a equação 3.8, conforme vemos na figura 28.

```

64      %%----- EVOLUTION OF LEVEL-SET FUNCTION -----
65      function [struc,lsf] = evolve(v,g,lsf,stepLength,w)
66      % Extend sensitivities using a zero border
67      vFull = zeros(size(v)+2); vFull(2:end-1,2:end-1) = v;
68      gFull = zeros(size(g)+2); gFull(2:end-1,2:end-1) = g;
69      % Choose time step for evolution based on CFL value
70      dt = 0.1/max(abs(v(:)));
71      % Evolve for total time stepLength * CFL value:
72      for i = 1:(10*stepLength)
73      % Calculate derivatives on the grid
74      dpdx = circshift(lsf,[0,-1])-lsf;
75      dmdx = lsf - circshift(lsf,[0,1]);
76      dpdy = circshift(lsf,[-1,0]) - lsf;
77      dmdy = lsf - circshift(lsf,[1,0]);
78      % Update level set function using an upwind scheme
79      lsf = lsf - dt * min(vFull,0) .* ...
80      sqrt( min(dmdx,0).^2+max(dpdx,0).^2+min(dmdy,0).^2+max(dpdy,0).^2 ) ...
81      - dt * max(vFull,0) .* ...
82      sqrt( max(dmdx,0).^2+min(dpdx,0).^2+max(dmdy,0).^2+min(dpdy,0).^2 ) ...
83      - w*dt*gFull;
84      end
85      % New structure obtained from lsf
86      strucFull = (lsf<0); struc = strucFull(2:end-1,2:end-1);

```

Figura 28 – Evolução da função de LS. Fonte: (CHALLIS; GUEST, 2009).

### 4.6.5 Implementação da análise FEA - Linhas 87-106

Assim como foi feito no algoritmo com abordagem SIMP, é necessário a determinação dos parâmetros da matriz de rigidez para trabalhar com a função de chamada da linha 11. Aqui podemos observar nas linhas 100 à 102, as condições de contorno ao qual o problema de otimização é aplicado, que no caso foi especificado para uma ponte (Bridge). O trabalho

posterior de implementação se baseará em modificar tais condições de contorno para que tenhamos um domínio de projeto de uma longarina de um chassi veicular. Conforme observado na linha 97, como é um método que não precisa de penalização pois usa variáveis discretas, é necessário apenas adotar um valor de  $(elx, ely)$ , 0,0001 para garantir uma não singularidade para resolver o problema de otimização, conforme figura 29.

```

87 %----- FINITE ELEMENT ANALYSIS -----
88 function [U] = FE(struc,KE)
89 [nely,nelx] = size(struc);
90 K = sparse(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1));
91 F = sparse(2*(nely+1)*(nelx+1),1); U = zeros(2*(nely+1)*(nelx+1),1);
92 for elx = 1:nelx
93     for ely = 1:nely
94         n1 = (nely+1)*(elx-1)+ely;
95         n2 = (nely+1)* elx +ely;
96         edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1; 2*n2+2; 2*n1+1; 2*n1+2];
97         K(edof,edof) = K(edof,edof) + max(struc(ely,elx),0.0001)*KE;
98     end
99 end
100 % Define loads and supports - Bridge:
101 F(2*(round(nelx/2)+1)*(nely+1),1) = 1;
102 fixeddofs = [2*(nely+1)-1:2*(nely+1),2*(nelx+1)*(nely+1)-1:2*(nelx+1)*(nely+1)];
103 % Solving
104 alldofs = [1:2*(nely+1)*(nelx+1)];
105 freedofs = setdiff(alldofs,fixeddofs);
106 U(freedofs,:) = K(freedofs,freedofs) \ F(freedofs,:);

```

Figura 29 – Implementação da análise FEA. Fonte: (CHALLIS; GUEST, 2009).

#### 4.6.6 Matriz de rigidez dos elementos - Linhas 107-129

Neste último grupo temos a definição da matriz de rigidez dos elementos, por meio da função *materialInfo* que retorna os elementos da matriz de rigidez  $KE(= \mathbf{ke})$  conforme figura 30.

```

107 %----- MATERIAL INFORMATION -----
108 function [KE,KTr,lambda,mu] = materialInfo()
109 % Set material parameters, find Lamé values
110 E = 1.; nu = 0.3;
111 lambda = E*nu/((1+nu)*(1-nu)); mu = E/(2*(1+nu));
112 % Find stiffness matrix "KE"
113 k = [ 1/2-nu/6 1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 ...
114       -1/4+nu/12 -1/8-nu/8 nu/6 1/8-3*nu/8];
115 KE = E/(1-nu^2)*stiffnessMatrix(k);
116 % Find "trace" matrix "KTr"
117 k = [1/3 1/4 -1/3 1/4 -1/6 -1/4 1/6 -1/4];
118 KTr = E/(1-nu)*stiffnessMatrix(k);
119 %----- ELEMENT STIFFNESS MATRIX -----
120 function [K] = stiffnessMatrix(k)
121 % Forms stiffness matrix from first row
122 K=[k(1) k(2) k(3) k(4) k(5) k(6) k(7) k(8)
123    k(2) k(1) k(8) k(7) k(6) k(5) k(4) k(3)
124    k(3) k(8) k(1) k(6) k(7) k(4) k(5) k(2)
125    k(4) k(7) k(6) k(1) k(8) k(3) k(2) k(5)
126    k(5) k(6) k(7) k(8) k(1) k(2) k(3) k(4)
127    k(6) k(5) k(4) k(3) k(2) k(1) k(8) k(7)
128    k(7) k(4) k(5) k(2) k(3) k(8) k(1) k(6)
129    k(8) k(3) k(2) k(5) k(4) k(7) k(6) k(1)];

```

Figura 30 – Matriz de rigidez dos elementos. Fonte: (CHALLIS; GUEST, 2009).

Sendo assim, a partir da descrição dos grupos, podemos verificar a implementação do algoritmo com a abordagem LS de variáveis discretas. Será demonstrado em capítulos

posteriores as alterações frente as condições de contorno deste algoritmo e demais outras linhas para que seja utilizado o objeto de estudo proposto que é a topologia da longarina do chassi veicular. A descrição do algoritmo completo bem como a implementação do algoritmo com as novas condições de contorno estão descritas nos apêndices "B" e "D" deste trabalho.

## 4.7 Algoritmo de Otimização Topológica 3D

Os algoritmos descritos acima que são o SIMP baseado em [Sigmund \(2001\)](#) e o baseado em LS [Challis e Guest \(2009\)](#), são algoritmos configurados para domínios bidimensionais - 2D. Como o objeto de estudo que é a longarina do chassi veicular tem a característica de ser tridimensional - 3D, apresentamos aqui um algoritmo da literatura desenvolvido com a proposta 3D. No entanto, para este trabalho utilizaremos para a obtenção dos dados coletados do programa computacional ParetoWorks para demonstrar a sintetização 3D do objeto de estudo e comparar com os algoritmos 2D mencionados.

Portanto, um dos algoritmos 3D desenvolvidos foi o de [Liu e Tovar \(2013\)](#). Este é um eficiente algoritmo escrito em MATLAB®, que tem 169 linhas, e utiliza análise de elementos finitos, filtro de densidade e otimizador de critério de otimalidade para resolver problemas de minimização de compliance, utiliza estratégias para programação linear do tipo MMA, além do CO. É um ótimo artigo e vários exemplos de implementação podem ser feitos. Como sugestão a futuros trabalhos, é possível desenvolver um algoritmo para o objeto de estudo deste trabalho, reescrevendo a forma do domínio da matriz de rigidez para o perfil da longarina que tem o formato "C". No apêndice "E" encontramos este algoritmo 3D para conhecimento e estudos posteriores de implementação de um possível algoritmo 3D para o domínio apresentado neste trabalho.

## 4.8 Descrição dos Programas Computacionais

Os programas computacionais utilizados para obter a topologia da longarina do chassi comercial foram o Virtual.PYXIS (Virtual CAE) e o ParetoWorks (Sciart). A escolha destes programas comerciais deveu-se devido a disponibilidade dos mesmos para sintetizar OT com abordagens distintas (o Virtual.PYXIS é baseado no SIMP e o ParetoWorks é baseado no LS). Assim com a distinção destas abordagens é possível avaliar que o projetista na indústria poderá ter com vários recursos computacionais para utilizar o método para encontrar a topologia da longarina do chassi ou de outros componentes do chassi. É factível que pelo fato das abordagens serem diferentes, espera-se resultados diferentes, no entanto procura-se mostrar a possibilidade de utilizar um ou outro programa computacional para que os projetistas façam a sintetização de OT, pois espera-se como resultado um efeito qualitativo para a topologia da estrutura do objeto de estudo.

#### 4.8.1 Virtual.PYXIS - Baseado no SIMP

A Virtual CAE é uma empresa localizada em São Caetano do Sul - SP e que iniciou suas atividades como representante comercial, e evoluiu para o lançamento do programa computacional próprio Virtual.PYXIS no ano de 2015, que otimiza o design de componentes. O add-on Virtual.PYXIS é um pré-processador que roda em conjunto com o solver do ANSYS Mechanical distribuídos no Brasil pela empresa ESSS. O Virtual.PYXIS é o primeiro programa computacional brasileiro de otimização de componentes estruturais, que possibilita a redução dos custos de produção e desenvolvimento em até 20% segundo o desenvolvedor, pela obtenção de componentes até 30% mais leves, mais confiáveis, eficientes e inovadores. É um programa computacional gratuito para a academia.

#### 4.8.2 ParetoWorks - Baseado no LS

Crociata (2016) utilizou o programa computacional ParetoWorks 2016 para trabalhar com recursos de manufatura aditiva, segundo ele o programa computacional é um add-on (complemento) desenvolvido por Sciart, LLC, e que trabalha dentro do ambiente do Solidworks e não oferece a possibilidade de criar peças e sim de trabalhar com FEA e OT, sendo que as peças são criadas exclusivamente no próprio programa SolidWorks, e após acessar o comando para iniciar o ParetoWorks, é possível executar o FEA deste componente (definindo o número de elementos que o componente será dividido), sequencialmente é definido as condições de contorno e por fim é gerado a sintetização de otimização topológica. Ainda segundo Crociata (2016), o ParetoWorks foi liberado para uso inicialmente em 2013 e foi criado pela Universidade de Winsconsin (Madison). É um programa computacional baseado na abordagem *Level Set*, por isto ele é diferente da maioria dos programas comerciais que tem em sua essência o uso do SIMP.



## 5 Objeto de análise

### 5.1 Algumas aplicações em chassi estrutural com OT

No ramo automotivo já houveram vários estudos sobre chassi de veículos e caminhões. Alguns destes estudos são vistos em [Chen \(2000\)](#), [Silva \(2001\)](#), [Oliveira \(2007\)](#), [Prete, Mazzotta e Anglani \(2010\)](#), [Cavazzuti et al. \(2011\)](#), [Furtado \(2013\)](#), [Alba \(2015\)](#) e [Zhong et al. \(2016\)](#).

Conforme observamos em [Alba \(2015\)](#), [Silva \(2001\)](#) e [Oliveira \(2007\)](#) desenvolveram-se estudos estruturais de chassi automotivo tipo escada que contém a viga lateral e as travessas. [Furtado \(2013\)](#) trabalhou com conceitos de desenvolvimento de chassi veicular do tipo escada e depois foi utilizado o método de otimização paramétrica em [Alba \(2015\)](#) em complemento ao estudo anterior.

Em [Prete, Mazzotta e Anglani \(2010\)](#), foi desenvolvido a base do banco de uma aeronave que também pode ser aplicada na indústria automotiva da mesma forma, naquele estudo o autor trabalhou com as abordagens de homogeneização e SIMP, com o auxílio do programa computacional da Altair (Optistruct) para sintetizar a otimização. Foi possível ver as normas associadas, os testes executados e a fabricação do protótipo desenvolvido.

[Cavazzuti et al. \(2011\)](#) desenvolveu um artigo em que buscava reduzir peso do chassi de um Ferrari F458 Italia, em que procurou obter redução de peso na estrutura tubular do chassi e na distribuição da espessura do invólucro do carro, por meio das configurações de contorno para diversas situações no carro como força vertical, torção, crash-test etc. Utilizou-se os programas computacionais da Altair (Hyperworks 9 Optistruct 9) para resolver estes processos de otimização.

[Zhong et al. \(2016\)](#) desenvolveu um estudo de otimização para um ônibus, por meio de funções multi-objetivas baseados em algoritmo genético e também por otimização de forma, que utilizou o programa computacional MSC Patran para construir o modelo de EF, e que posteriormente foi fabricado o modelo.

Então diversos estudos estão sendo feitos com base em componentes ou até chassi completos de veículos automotivos. O nosso objeto de análise busca analisar um dos componentes principais de um chassi automotivo que é a longarina do caminhão. Especificamente na longarina não foi possível encontrar estudos com otimização topológica, assim verifica-se por meio da OT uma nova aplicação para este componente.

## 5.2 Critérios de Projeto e Fabricação de Chassis

Os engenheiros do produto no Brasil utilizam atualmente os sistemas CAE para analisar estruturas contínuas utilizando programas computacionais comerciais como ANSYS Mechanical, Abaqus etc. O chassi é especificado exclusivamente para dar ao veículo algumas características com relação aos aspectos de resistência, conforto para o motorista dentro da cabine, características de condução e processamento de tipos de carga nas mais diversas aplicações, como cargas contendo reservatório de água, cargas com rejeitos sólidos, cargas de transporte de animais etc. O chassi é projetado para absorver impactos com o sobre-chassi de carrocerias de diversas aplicações, como carro coletor de resíduos, carro de carga basculante, carro de transporte de carga rodoviário etc, conforme figura 31.



Figura 31 – Tipos de carrocerias aplicados no chassi. Fonte: Adaptado de (SCANIA(a), 2016).

As configurações do chassi dependem do tipo ou modelo do equipamento, variando de 4x2, ou seja com 4 rodas de contato com o pavimento, porém com tração em 2 rodas, passando por 4x4 conforme figura 32, além de outras configurações como 6x2, 6x6, 8x2 ou 8x4. Algumas aplicações para implementos de carroceria sobre o chassi podem ser projetadas para equipamentos especiais, ver figura 31. Nesta figura temos da esquerda para a direita, um caminhão coletor de resíduos (lixo), um caminhão basculante para transporte de areia, terra ou pedras, e um caminhão para transporte de mercadorias diversas, no entanto há uma enorme gama de possibilidades de aplicação para este chassi.

4x4

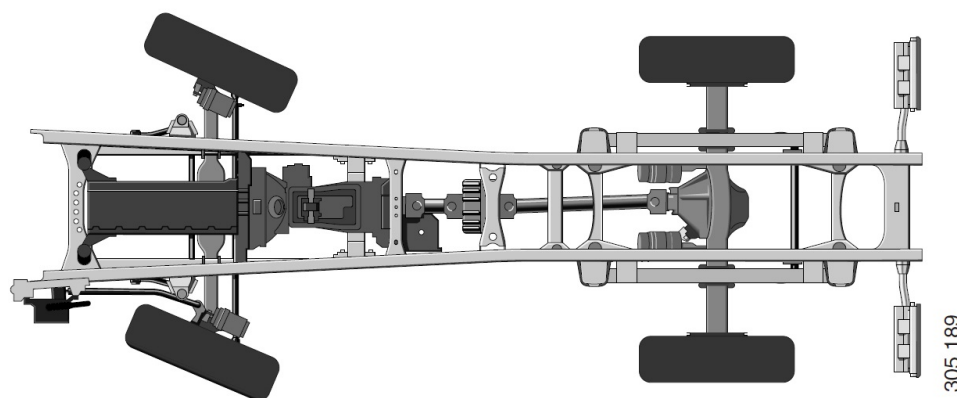


Figura 32 – Chassi 4x4. Fonte: Adaptado de (SCANIA(a), 2016).

A maioria dos implementos rodoviários ou *off-road*, como por exemplo os demonstrados na figura 31, que são fabricados pelos encarroçadores, exige dimensões diversificadas de entre-eixo (EE). O entre-eixo é a distância do eixo frontal até o eixo traseiro. Isto se deve pelo fato de que as indústrias fabricantes de veículos possuem em seu catálogo de venda apenas algumas medidas padronizadas de entre eixos devido a minimização de custo do seu processo de fabricação. Essa exigência de dimensão diversificada faz com que haja alterações no posicionamento do eixo traseiro alterando furações na longarina do chassi quando estas não são definidas pelo fabricante, conforme Figura 33. Também por este fato é importante que os engenheiros conheçam detalhadamente a longarina do chassi aplicada ao veículo.

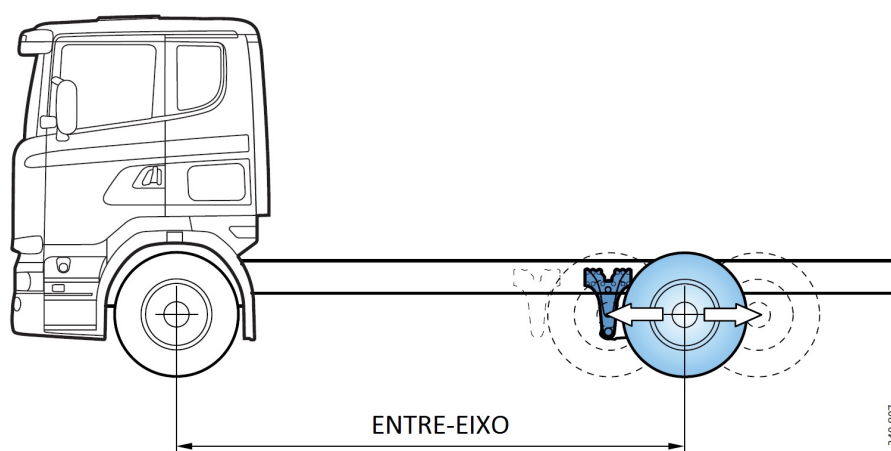


Figura 33 – Encurtamento ou prolongamento de entre-eixo. Fonte: Adaptado de (SCANIA(b), 2016).

O encurtamento da longarina do chassi é muito prejudicial ao veículo devido ao corte promovido no quadro do chassi. Em caminhões tratores e basculantes sem sobre-chassi não é recomendado pelo fabricante tal corte (SCANIA(b), 2016), pois todos os esforços serão aplicados diretamente no chassi. A furação ou retirada de material na longarina do chassi é um dos elementos mais importantes em relação a eficácia da aplicação de encarroçamento. Um fato também não recomendado pelos fabricantes de veículos é a perfuração na aba das longarinas quando estas forem do tipo perfil C (SCANIA(b), 2016), pois é nestas abas que são apoiados os quadros superiores de carrocerias e também como as abas estão localizadas nas extremidades do perfil "C", é mais propensa a trincar por meio da propagação dos furos na aba, apenas na extremidade traseira do chassi em exatos 100 mm recomenda-se travar ou furar a aba para fixação de travessa por exemplo, conforme observamos na figura 34. Deve-se portanto analisar tecnicamente a furação, sempre que possível utilizando a alma (espessura do lado central) do perfil C para a retirada dos materiais.

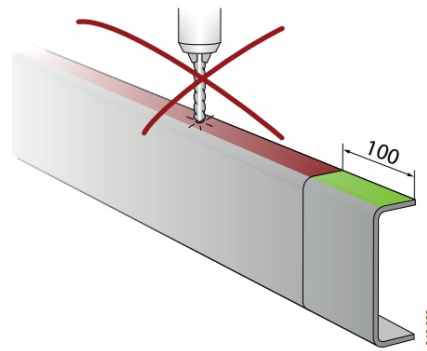


Figura 34 – Regras e condições de furação no chassi. Unidade:mm . Fonte: Adaptado de (SCANIA(b), 2016).

O fabricante do chassi prevendo o problema que pode ser originado em relação a perfuração de uma longarina de chassi, recomenda que sempre que possível possa ser utilizado o padrão de orifícios pré-furados nas longarinas, no entanto caso haja a necessidade de fazer novas furações pede-se que adotem a configuração da Figura 35 para a furação, de acordo com a norma (2007/46/EC, 2007), em que a letra "A" mostra a distância mínima entre o orifício e a aba, delimitando que esta dimensão deve ser 3 vezes o diâmetro do orifício acima de 40mm, já a dimensão "B" deve ser também 3 vezes o diâmetro maior que 50mm. O fabricante define que o tamanho dos furos localizados entre o eixo dianteiro e o eixo traseiro não pode ser maior do que 30mm.

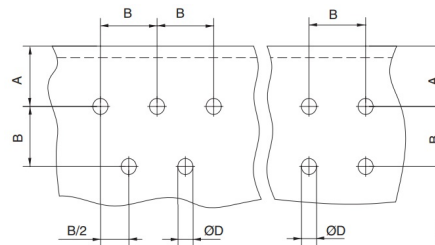


Figura 35 – Padrão permitido de furação na longarina. Fonte: Adaptado de (SCANIA(c), 2016).

Por mais que este padrão de furação da longarina seja adotado, existe sempre uma dúvida se aquele ponto da longarina pode ser furado ou não por parte dos encarroçadores, sem que isto afete criticamente a estrutura da longarina. Desta forma, verifica-se a necessidade de ferramentas para que possam auxiliar o projetista para definir com segurança as furações necessárias.

## 6 Metodologia

As empresas de engenharia de projetos apresentam desenvolvimentos em diversos segmentos, como aeronáutico, automobilístico, siderúrgico, farmacêutico, hidráulico, eletrônico etc. O MOT pode ser aplicado em todos estes segmentos. No Brasil ainda há poucas empresas que buscam suprir as necessidades dos clientes para o desenvolvimento de produtos por meio do MOT. Os conceitos trabalhados neste tema procuram entender a estrutura de tarefas para a síntese de um produto por meio do MOT, e com este procurar soluções para que haja uma desmitificação da dificuldade do uso dos sistemas CAE pelos projetistas de empresas de engenharia. Para atingir os objetivos deverá ser realizado um estudo de uma longarina de chassi veicular a partir dos programas computacionais ParetoWorks, Virtual.PYXIS e algoritmos de MOT, e propor uma nova topologia da longarina do chassi. O intuito desta metodologia é o de ajudar-nos a compreender o não compreensível, nos mais amplos termos, não os produtos somente da pesquisa científica, mas o próprio processo, desta forma, qualquer pesquisa científica bem sucedida implica em certa consciência metodológica, (KAPLAN; HEGENBERG; MOTA, 1972). A metodologia de pesquisa adotada dar-se-á com bases na execução de estudos bibliográficos das referências que existem com relação aos casos de análise aplicando-se MOT, e a partir destes estudos, mensurar, ou seja, quantificar os estudos referente a aplicação do MOT nas empresas de engenharia. Esta execução buscará primeiro definir a fundamentação do que é, e como atuam as empresas de engenharia nas diversas organizações deste meio no Brasil, com relação ao desenvolvimento de produtos por meio do MOT, a partir desta definição será explorado os tipos de tarefas que já existem por meio do estudo de MOT, e após isto deverá ser relacionado com o estudo de desenvolvimento de uma longarina de um chassi veicular. Na elaboração do trabalho, será adotado uma metodologia estruturada pela sequência de passos representados graficamente na Figura 36.

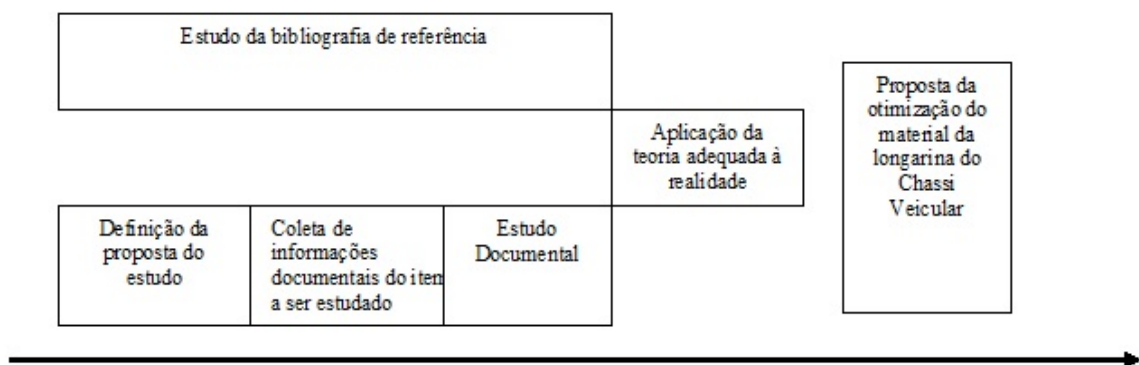


Figura 36 – Estrutura de pesquisa. Fonte: Do autor.

Por meio da técnica de coleta de dados proposta inicialmente, será realizada uma pesquisa bibliográfica de referência com enfoque nas áreas de Otimização Topológica, Mecânica de Estruturas e Processos de Fabricação, representadas na figura 37 sendo que, os conceitos abordados nas três áreas contemplam o interesse da pesquisa.

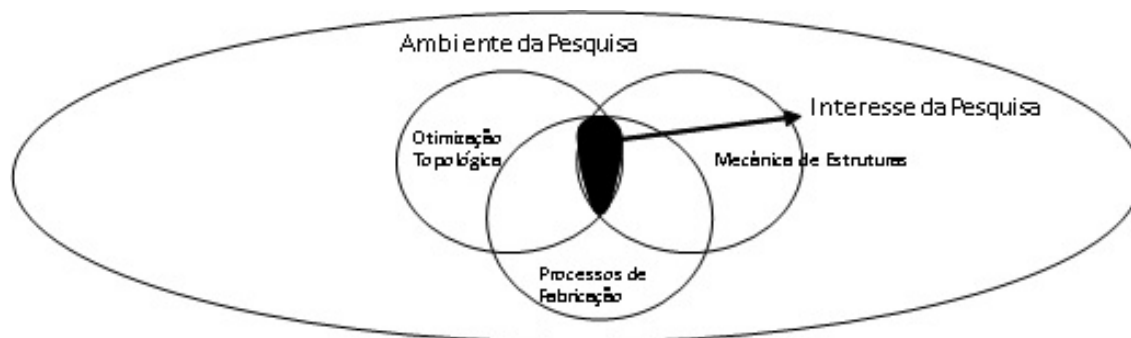


Figura 37 – Área de interesse da pesquisa. Fonte: Do autor

É observado que o maior ambiente de pesquisa relacionado ao trabalho será focado na área de otimização topológica, as outras áreas mecânica das estruturas e processos de fabricação são complementares ao trabalho, mas são necessários para o desenvolvimento do objeto da proposta que é o desenvolvimento de uma topologia para a longarina do chassi veicular.

Após o levantamento bibliográfico do objeto de estudo, realiza-se uma classificação em relação aos métodos de otimização estrutural a fim de delinear as fronteiras que definem o escopo da presente pesquisa. Como principal técnica para atingir os objetivos deste trabalho, utiliza-se a pesquisa documental que, segundo Godoy (1995), é o exame de materiais de natureza diversa, que ainda não receberam um tratamento analítico, ou que podem ser reexaminados, buscando-se interpretações novas ou complementares. Esta parte documental será gerado por meio dos resultados dos programas computacionais ParetoWorks e Virtual.PYXIS e de algoritmos de MOT. A possibilidade de observar a sistemática dos algoritmos e a operacionalidade dos programas computacionais, gera uma natureza de dar critérios ao leitor para poder avaliar se o uso do MOT é eficiente para a aplicação de encontrar novas topologias de uma longarina de um chassi veicular.

Como procedimentos metodológicos serão coletados todos os dados dos programas computacionais e dos algoritmos e serão analisados e discutidos. Sabe-se que todo procedimento deve ser seguido por etapas para que outros pesquisadores possam testar as mesmas condições encontradas neste trabalho. Sendo assim, no próximo capítulo discutiremos tais procedimentos metodológicos.

## 7 Procedimentos Metodológicos

Nesta seção, é definido quais foram os itens gerais contemplados para modo de coleta de dados e o modo de análise dos dados, que são a base para as conclusões do trabalho.

### 7.1 Modo de Coleta de Dados

- Estudo e levantamento das dimensões da longarina de um chassi comercial.
- Criação do arquivo CAD para esta longarina.
- Implementação dos algoritmos.
- Definição dos elementos finitos do objeto de estudo, incluindo as novas condições de contorno (domínio do projeto, fixação e cargas) dos algoritmos e dos programas computacionais.

### 7.2 Modo de Análise dos dados coletados

- Analisar os dados coletados pelos algoritmos.
- Analisar os dados coletados pelos programas computacionais.
- Comparar os resultados dos programas computacionais e do algoritmo.
- Propor uma nova topologia para a longarina.

### 7.3 Materiais

Nesta seção está contemplado todos os recursos utilizados para o desenvolvimento dos algoritmos e as sintetizações e simulações com os programas computacionais comerciais. Para o desenvolvimento dos algoritmos, as sintetizações e as simulações, adotou-se como recurso principal um notebook Dell Inspire com processador Intel(R) Core (TM) i7-3630QM com CPU de 2.4 GHz, com memória RAM instalada de 12 GB, com o sistema operacional Windows 10, versão de 64 bits, conforme figura 38, neste equipamento estavam configurados os seguintes programas:

- 1- MATLAB® - Versão Student R2017a, modelo 64-bit (win64)
- 2- AutoCAD - Versão 2007, modelo 64-bit (win64)



- 3- SolidWorks - Versão Student 2015, modelo 64-bit (win64)
- 4- ParetoWorks - Versão 2017, modelo 64-bit (win64)
- 5- ANSYS Workbench - Versão Student 17.2, modelo 64-bit (win64)
- 6- Virtual.PYXIS - Versão 171.1, modelo 64-bit (win64)



Figura 38 – Especificações do computador. Fonte: Do autor

## 7.4 Métodos

Definiu-se o método utilizado para a coleta dos dados que resultará num possível posicionamento de furos no chassi de um veículo comercial. Para a obtenção dos dados, seguiu-se um método, que tanto para a sintetização dos algoritmos, quanto para a sintetização com os programas computacionais comerciais, pode gerar um esquema sintetizado para posterior tratamento com o programa computacional AutoCAD por meio da colocação de figuras para resultar em uma nova estrutura. Sendo assim, as etapas para a obtenção do produto sintetizado pelos algoritmos e pelos programas computacionais comerciais estão descritas nos tópicos a seguir.

### 7.4.1 Estudo e levantamento das dimensões da longarina do chassi comercial

A dimensão principal do chassi comercial com base no veículo P440 CB 4x4 Scania à ser definido no capítulo 7.4.4 foi levantada e tem as dimensões conforme figura 39.



Figura 39 – Dimensões da longarina do chassi estudada. Fonte: Do autor



Foi analisado também as dimensões do perfil desta longarina conforme figura 40. Estes foram os dados de dimensões colhidos pelo autor e utilizados como premissa para obtermos os domínios de projeto para esta aplicação.

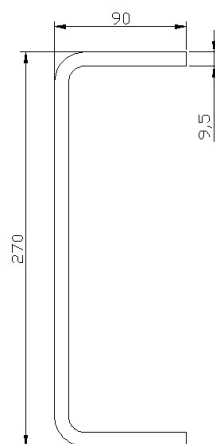


Figura 40 – Características do perfil. Unidade (mm). Fonte: Do autor

Verificou-se que os dois pontos de apoio móvel, que são os os eixos principais da estrutura (eixo dianteiro e eixo traseiro) tem a dimensão de 3.900mm. Foi observado que na linha de portfólio de produtos da Scania, este mesmo veículo P440 CB 4x4 pode ter outras dimensões de chassis como 4.100 mm, 4.300 mm, 4.500 mm, 4.700 mm., conforme figuras 41 e 42.

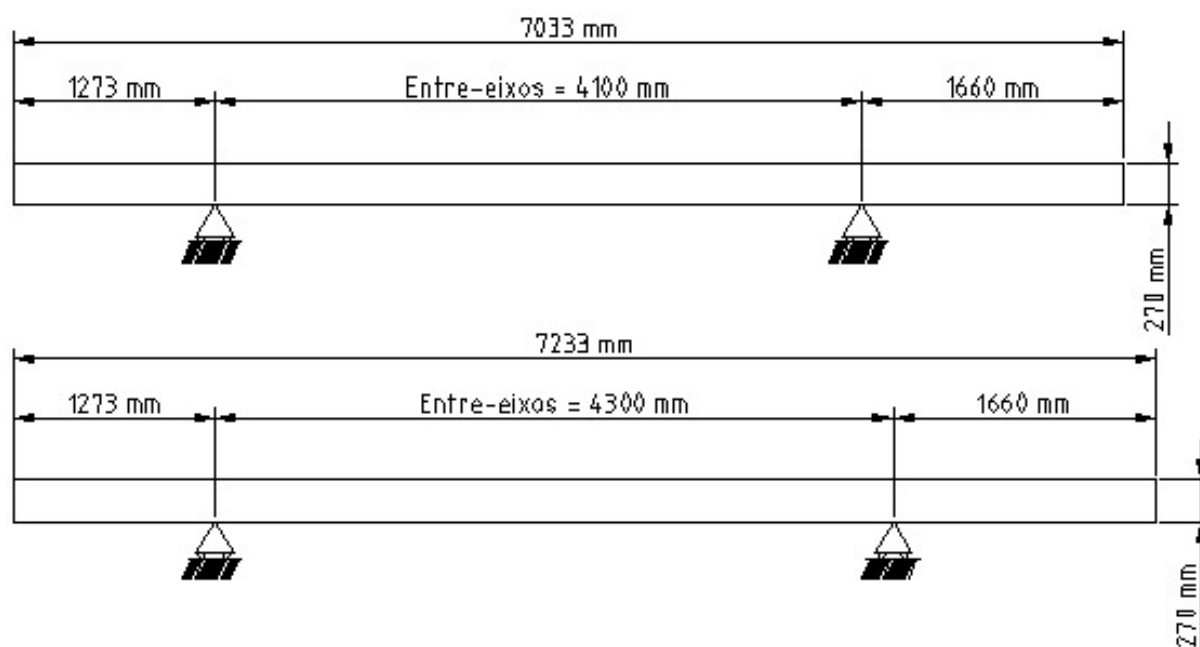


Figura 41 – Longarina com EE 4.100 mm e 4.300 mm. Fonte: Do autor

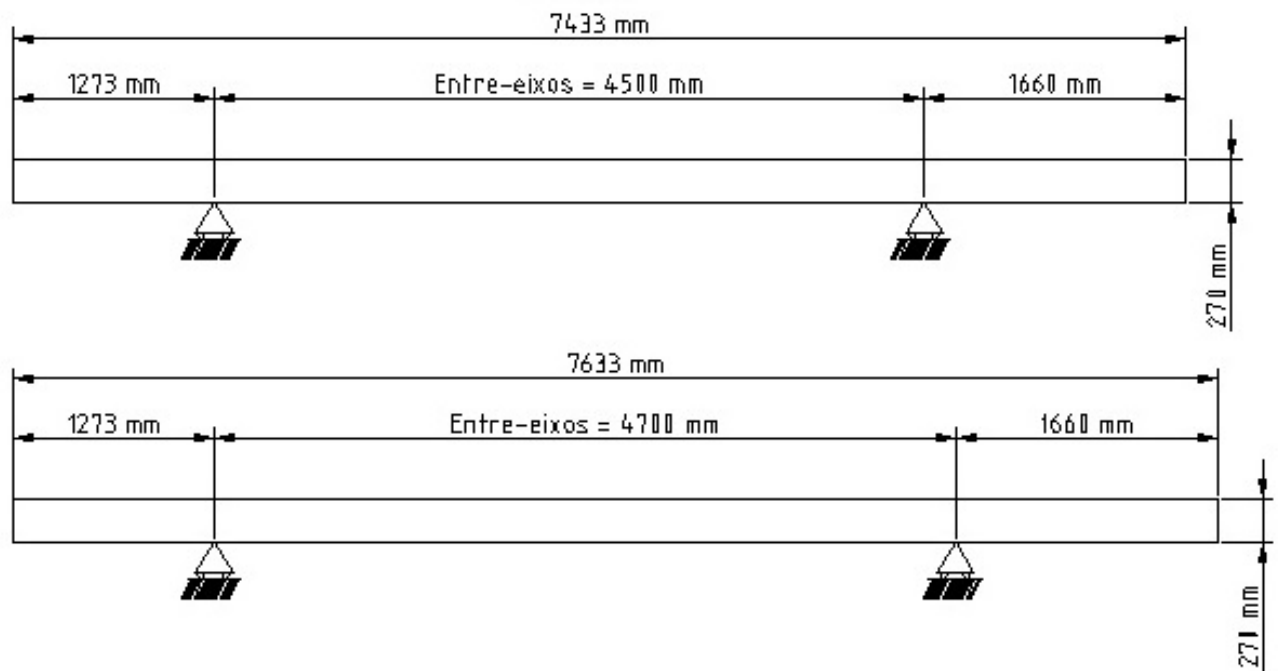


Figura 42 – Longarina com EE 4.500 mm e 4.700 mm. Fonte: Do autor

#### 7.4.2 Criação do arquivo CAD da longarina do chassi para posterior tratamento

O arquivo CAD foi gerado nos *software* Autocad para a condição 2D em que foi desconsiderada as abas, conforme figura 43, e no Solidworks para a condição 3D considerada real com o produto físico, conforme figura 44, as análises serão feitas com o desenho bidimensional (2D) e também com o desenho tridimensional (3D) real neste trabalho. Para referência foi levantado e detalhado todos os furos existentes ao longo desta longarina. A posição destes furos existentes na longarina do chassi comercial será importante em relação aos objetos de análise sintetizados posteriormente, pois por meio delas o projetista poderá decidir se mantêm um furo projetado apenas com os recursos de CAE ou elimina o mesmo do projeto da longarina do chassi com base nos dados encontradas pelo MOT. Se em trabalhos futuros for definido criar um algoritmo para a aplicação em 3D, o autor já possui os dados da longarina em 3D.

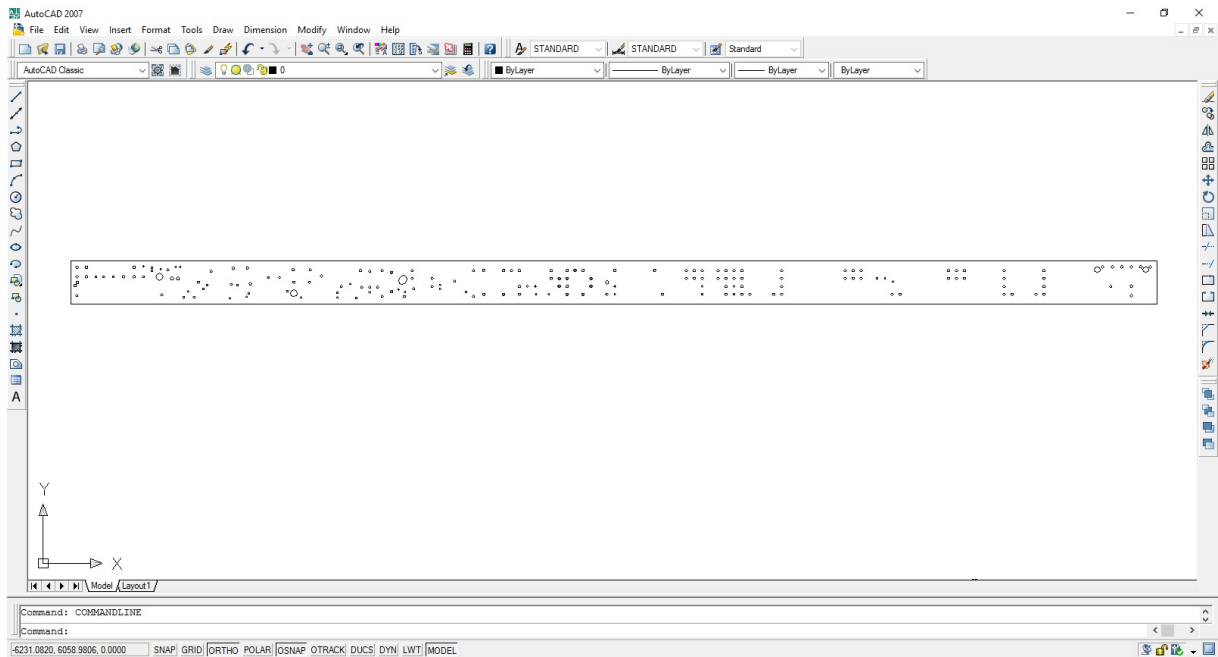


Figura 43 – Modelo 2D do chassi comercial no AutoCad. Fonte: Do autor

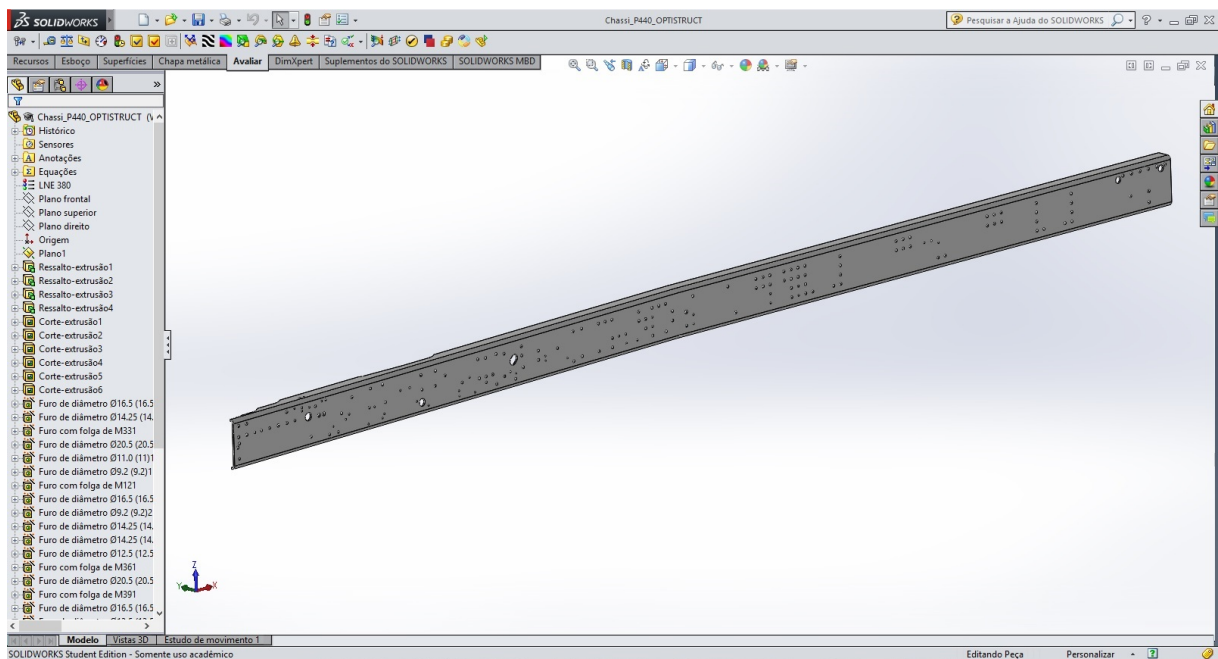


Figura 44 – Modelo 3D do chassi comercial no SolidWorks. Fonte: Do Autor.

### 7.4.3 Implementação dos algoritmos

Os algoritmos implementados baseiam-se em estruturas bidimensionais 2D, no entanto como o perfil da longarina tem abas e é do tipo "C", iremos transportar a carga aplicada sobre as abas gerando um momento sobre o plano 2D. Acharemos a força cortante por meio disto sobre este plano, e assim conseguiremos trabalhar com os algoritmos 2D simulando um perfil 3D, pois se adotarmos somente o modelo 2D sem transportar este momento de forças distribuídas sobre a aba para a sua face planar teremos um erro de

condição de contorno para o perfil.

O autor procurou desenvolver neste trabalho alguns algoritmos de OT, portanto é interessante também trabalhar com estes algoritmos em 2D porém com efeitos de carga encontrados no perfil 3D. O objeto de estudo que é uma longarina de chassi veicular, poderá ser otimizado para estruturas tridimensionais já que o perfil possui um corpo e duas abas com um algoritmo direto 3D, no entanto devido à inicial complexibilidade dos cálculos e redefinição do novo domínio do perfil C, o autor procurou trabalhar diretamente com o programa computacional ParetoWorks em 3D para colher os dados de sintetização de OT para esta aplicação. Os algoritmos 2D irão compor o fluxo a seguir para todo o processo de iteração do MOT. A proposta deve seguir alguns critérios para a solução ótima, que com base em [Sigmund \(2001\)](#), pode ter a seguinte sequência:

- I. Análise dos elementos finitos (FEM).
- II. Análise da sensibilidade ou do filtro de densidade.
- III. Aplicação do critério de otimalidade com otimização em loop.
- IV. Demonstração de resultados.
- V. Análise dos resultados.

E por base em [Challis e Guest \(2009\)](#) pode ter a seguinte sequência:

- I. Análise dos elementos finitos (FEM).
- II. Cálculo da derivada topológica e domínio.
- III. Aplicação do critério de evolução por curvas de nível.
- IV. Demonstração de resultados.
- V. Análise dos resultados.

A solução por método numérico requer que o domínio seja discretizado. Neste trabalho, o domínio da estrutura é discretizado por elementos finitos de placa/casca para o EE de 3.900mm. O fluxograma do MOT é mostrado na figura 45 para a abordagem SIMP de [Sigmund \(2001\)](#).

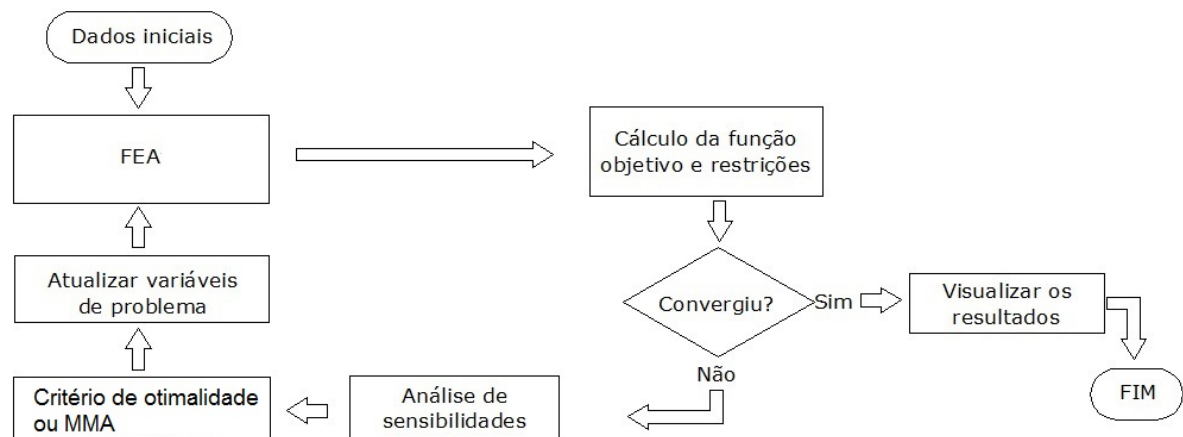


Figura 45 – Diagrama do MOT utilizado para o SIMP. Fonte: Adaptado de Libera e Lima (2012).

Já para a abordagem de LS baseado em Challis e Guest (2009), o fluxograma do MOT é mostrado conforme a figura 46.

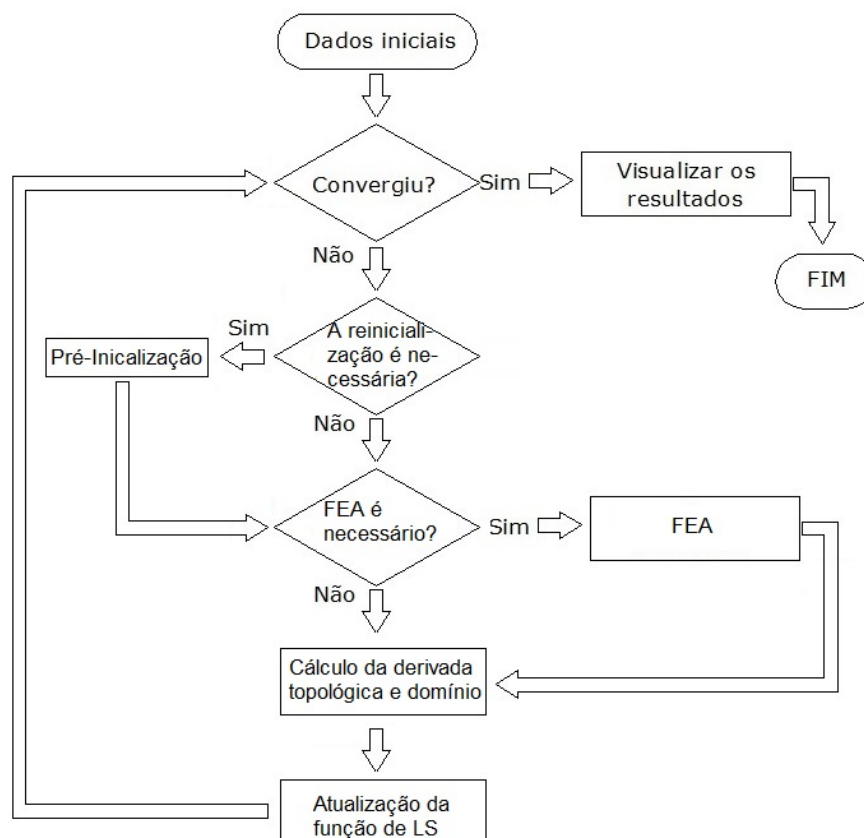


Figura 46 – Diagrama do MOT utilizado para o LS. Fonte: Adaptado de Wu e Wu (2016).

A diferença do método SIMP e do LS, segundo estes fluxogramas é que o CO será substituído pelo otimizador do LS que utiliza um método de evolução com uma função de atualização conforme vimos na equação 3.8 para poder otimizar a função de LS por meio das curvas de nível por meio de cálculo da derivada topológica e procura de domínio.

Portanto, ainda em relação as alterações do códigos anteriormente propostos por [Sigmund \(2001\)](#) e [Challis e Guest \(2009\)](#), as etapas para o desenvolvimento do código proposto foram necessárias:

- I. Alteração do tipo de viga MBB retirando a condição de simetria para a aplicação da longarina assimétrica.
- II. Alteração dos pontos de fixação da viga para a posição similar ao do chassi comercial estudado.
- III. Alteração da carga unitária para carga distribuída ao longo da viga.
- IV. Identificação do número de elementos  $nelx$  e  $nely$  para a viga com base em uma quantidade de  $N=10.000$  elementos e das dimensões do chassi comercial estudado.
- V. Possibilidade de inserir o filtro de densidade de [Andreassen et al. \(2011\)](#) em frente ao filtro de sensibilidade de [Sigmund \(2001\)](#) à ser feito em trabalhos futuros.

No algoritmo de [Andreassen et al. \(2011\)](#), temos uma mudança em relação a [Sigmund \(2001\)](#) com a inclusão do filtro de densidade além do filtro de sensibilidade já existente. O filtro de densidade é uma técnica moderna tal como o filtro Heaviside proposto por [Guest, Prévost e Belytschko \(2004\)](#) e [Sigmund \(2007\)](#). O algoritmo inicia o cálculo da função objetivo e o otimizador gera novos valores de variáveis ótimas. A iteração do processo de otimização continua até a convergência da função objetiva.

Com base na nova estrutura do algoritmo SIMP sem a inclusão do filtro de densidade, apenas se baseando nas 99 linhas do código de [Sigmund \(2001\)](#) o código MATLAB® utilizado pode ser chamado com o comando da equação 7.1.

$$topopt(nelx, nely, volfrac, penal, rmin) \quad (7.1)$$

Em que o significado de cada parâmetro de entrada foi discutido anteriormente na descrição do algoritmo de SIMP.

E de acordo com [Challis e Guest \(2009\)](#), a equação 7.2 representa a função de entrada do algoritmo de LS.

$$top\_levelset(nelx, nely, volReq, stepLength, NumReinit, topWeight) \quad (7.2)$$

Em que o significado de cada parâmetro de entrada foi discutido anteriormente na descrição do algoritmo de LS.

Uma inicial alteração frente ao algoritmo de [Sigmund \(2001\)](#) e de [Challis e Guest \(2009\)](#) é necessária para transformar a condição de viga MBB para uma viga completa do tipo longarina de chassi. Foi possível alterar as condições iniciais de contorno para transformar o código de 99 linhas de [Sigmund \(2001\)](#) e as 129 linhas de [Challis e Guest \(2009\)](#) em um projeto para o estudo de uma longarina de chassi. O carregamento distribuído que é desejado nesta viga, é conhecido expresso em [N/m], mas precisa ser transformado em carga nodal [N], pois a carga é aplicado nos DOFs (grau de liberdade de cada elemento). Assim para distribuímos o valores das cargas em cada nó de elemento considerando a direção  $x$  que é a direção que estamos adotando para as cargas basta dividirmos o valor de  $\frac{-F_{total}}{nelx+1}$ , desta forma iremos garantir que a força é distribuída igualmente em cada nó.

Portanto para definir as cargas aplicadas no elementos na linha superior, consideramos um exemplo de acordo com a figura 47, em que temos 10 elementos dispostos lado a lado, assim  $nelx = 10$  e  $nely = 1$ .

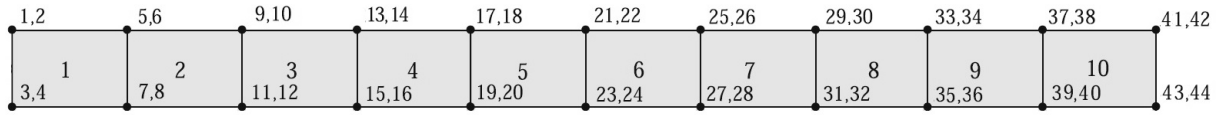


Figura 47 – Exemplo para definir cargas distribuídas nos nós do domínio. Fonte: Do autor

Como observado na figura 47, há 2 DOFs para cada nó, um representando o deslocamento na horizontal e o outro representando o deslocamento na vertical, e que são nomeados de cima para baixo e da esquerda para a direita. Além disto há a posição de cada um destes 10 elementos, portanto de acordo com este exemplo temos:

$elx$ : Posição do elementos, que pode estar entre 1 e 10.

2, 6, 10, 14, 18, 22, 26, 30, 34, 38, 42: DOFs na linha superior na posição vertical. Sendo assim a condição de carga encontrada para satisfazer estas condições está de acordo com a equação 7.3.

$$F(2 * (nely + 1) * (elx - 1) + 2, 1) = -\frac{F_{total}}{nelx+1} \quad (7.3)$$

A prova que podemos encontrar para satisfazer esta equação para um determinado DOF da linha superior qualquer conforme demonstrado na figura 48, em que há por exemplo uma força vertical no nó de DOF 22, no elemento de número 6 poderá ser descrita conforme a equação 7.4.

$$\begin{aligned} F(2 * (1 + 1) * (6 - 1) + 2, 1) &= -\frac{10000}{10+1} \\ F(2 * (2) * (5) + 2, 1) &= -9090, 91 \\ F(22, 1) &= -9090, 91 \end{aligned} \quad (7.4)$$

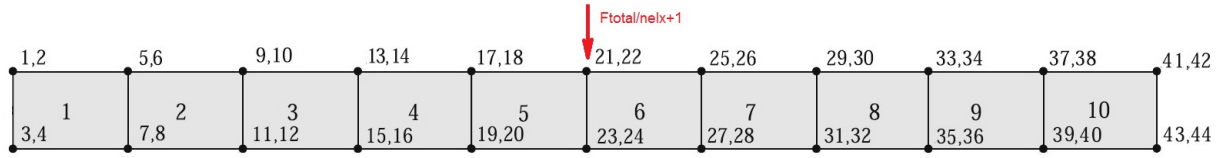


Figura 48 – Exemplo de carga no elemento 6. Fonte: Do autor

Então, de acordo com a equação 7.4, encontramos a posição da força no DOF 22 na linha 1, com o valor de 9090,21N. Para que este valor apareça em todos os nós superiores, simplesmente podemos descrever uma condição de varredura de todos os nós garantindo a força distribuída na domínio do projeto. Esta varredura poderá ser descrita de acordo com a equação 7.5.

$$\text{for } elx = 1 : nelx + 1 \quad (7.5)$$

A equação 7.5 garantirá que todos os nós superiores apresentem um carga, inclusive o nó da extremidade de acordo com a sintaxe +1.

Outra condição que deveremos avaliar no domínio são os suportes ou as fixações, como um chassi veicular apresenta abaixo de sua longarina dos eixos dianteiro e traseiros equidistantes, logo teremos dois pontos de suporte abaixo da longarina, e supondo que o eixo dianteiro seja de restrição apenas vertical e esteja localizado sob a coluna de nós “nsup1:nsup2” e que o eixo traseiro seja de restrição vertical e esteja localizado sob a coluna de nós “nsup3:nsup4”, teremos a fixação do domínio do projeto de acordo com a equação 7.6.

$$\text{fixeddofs} = \text{union}([2 * (nely + 1) * (nsup1 : nsup2)], [2 * (nely + 1) * (nsup3 : nsup4)]); \quad (7.6)$$

Assim, o código alterado para as novas condições de contorno para o algoritmo SIMP baseado em Sigmund (2001), será de acordo com a figura 49.

```

79  % DEFINIÇÃO DE CARGAS E SUPORTES (CASO LONGARINA DO CHASSI)
80  - for elx = 1:nelx+1
81  -     F(2*(nely+1)*(elx-1)+2,1) = -Ftotal/(nelx+1);
82  - end
83  - fixeddofs = union(2*(nely+1)*(nsup1:nsup2), 2*(nely+1)*(nsup3:nsup4));

```

Figura 49 – Alteração de código para longarina do chassi baseado no código SIMP. Fonte: Do autor

No código alterado para as novas condições de contorno para o algoritmo Level Set baseado em Challis e Guest (2009), teremos a substituição das linhas referente a figura 50.



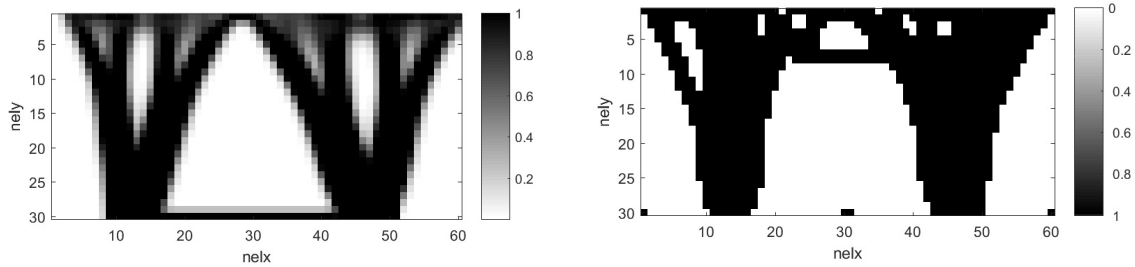
```

102 % Definição de Carregamentos e Suportes - Longarina do Chassi:
103 - F = sparse(2*(nely+1)*(nelx+1),1);
104 - fixeddofs = union(2*(nely+1)*(nsup1:nsup2),2*(nely+1)*(nsup3:nsup4));
105 - for elx = 1:nelx+1
106 -     F(2*(nely+1)*(elx-1)+2,1) = -1/(nelx+1);
107 - end
108 - alldofs = [1:2*(nely+1)*(nelx+1)];
109 - freedofs = setdiff(alldofs,fixeddofs);
110 - U(freedofs,:) = K(freedofs,freedofs) \ F(freedofs,:);

```

Figura 50 – Alteração de código para longarina do chassi baseado no código LS. Fonte: Do autor

Portanto, ficará denominado de “Ftotal”, “nsup1”, “nsup2”, “nsup3” e “nsup4” os parâmetros a serem modificados no código para cada análise, que fisicamente são números, um exemplo seria a descrição da função de entrada do algoritmo SIMP implementado `topopt(60,30,0.5,3.0,1.5)`, ou do algoritmo LS implementado `toplevelset(60,30,0.5,3,2,2)` em temos  $nelx = 60$ , que são 60 elementos na direção x, e que um apoio esteja entre os nós 10:15 (`nsup1:nsup2`) e outro apoio esteja nos nós 45:50 (`nsup3:nsup4`), então teremos  $F_{total} = -100.000N$ . Assim, na figura 51 temos a diferença entre os dois casos em que a figura 51a) apresenta os dados colhidos do SIMP implementado, e a figura 51b) apresenta dos dados do LS implementado, provando que o algoritmo foi alterado com sucesso, ou seja, as condições de contorno (domínio, fixação e cargas são equivalentes).



(a) Novo Cod. SIMP implementado,  $c_0=3,35e+05$ ,  $c_1=7,13e+04$  e 80 iterações (b) Novo Cod. LS implementado,  $c_0=-4,41e+04$ ,  $c_1=-1.18e+05$  e 168 iterações

Figura 51 – Novas implementações SIMP e LS com malha 60x30, fração de volume de 50%,  $p=3$ ,  $r_{min}=1,5$ ,  $StepLength=3$ ,  $numReinit=2$  e  $topWeight=2$ . Fonte: Do Autor.

Verificou-se que os métodos são diferentes, porém as condições de contorno demonstram figuras aproximadas, as duas tem as forças na face superior, portanto está com a cor escura preenchida, e há dois pontos fixos na face inferior também com estas cores escuras. Observa-se que na direção dos pontos de fixação e próximo a linha superior há um rasgo que quando ensaiado para relação de aspecto cada vez mais próximos da longarina do chassi do chassi ficam na linha superior. Pela teoria de otimização topológica, já que a carga é distribuída na face superior não deveria existir rasgos ou furos próximos desta região, sendo assim será ensaiado várias condições de volume para tentar minimizar ou

desaparecer com esta condição.

Observa-se também que os resultados de compliance dos dois algoritmos apresentam resultados de valor inicial e final diferentes. Isto é válido, pois as estratégias são diferentes, portanto os valores de minimização de compliance denotam valores diferentes para cada uma das sintetizações.

#### 7.4.4 Definição das condições de contorno equivalentes ao algoritmo e aos programas computacionais

A longarina do chassi estudado é existente na indústria. O modelo do caminhão é o Scania P440 CB 4x4. O chassi é fabricado para atender veículos especiais como caminhão de bombeiro, figura 52, mas pode ser aplicado em outras carrocerias como já vimos anteriormente.



Figura 52 – Exemplo de veículo especial. Fonte: Cortesia Lavrita Engenharia ([LAVRITA, 2016](#)).

O significado do termo do caminhão Scania P440 CB 4x4 é apresentado conforme a tabela 10.

Tabela 10 – Classificação de tipo de veículo. Fonte: Do autor com base em e-mail fornecido pela empresa Scania

Designação	Característica
P	Tipo de cabine (Menor)
440	Potência do motor (CV)
C	Classe do chassis (p/ construção veículo fora de estrada)
B	Caminhão rígido ou básico (sem quinta roda)
4x4	Configuração das rodas (4 pontos de apoio com 4 pontos de tração)

A designação P também poderia ser G (cabine intermediária) ou R (cabine maior). A designação C poderia ser também L (p/ construção veículo longa distância) ou D (p/ construção veículo tipo distribuição e transporte regional). A designação B poderia ser A (Caminhão articulado com quinta roda). E o traçado poderia ser 6x2, 6x4, 6x6 etc, conforme vimos no capítulo 5.2.

Como por exemplo, um caminhão Scania do tipo R440 LA 6x4 que é o caminhão em que encontramos nas rodovias é classificado conforme a tabela 11.

Tabela 11 – Classificação de tipo de veículo. Fonte: Do autor com base em e-mail fornecido pela empresa Scania

Designação	Característica
R	Tipo de cabine (Maior)
440	Potência do motor (CV)
L	Classe do chassis (p/ construção veículo longa distância)
A	Caminhão articulado (com quinta roda)
6x4	Configuração das rodas (6 pontos de apoio com 4 pontos de tração)

A longarina do chassi tem muitas furações para fixar seus componentes da estrutura. Os projetistas e engenheiros da montadoras de veículos automotivos procuram desenvolver a longarina colocando os furos e depois simulando por meio de algum programa computacional para análise FEA de toda a estrutura, mas não é utilizado o MOT para sintetizar o projeto. A longarina do chassi possui furos ao longo de sua extensão. Estes furos são importantes para fixação de todos os componentes que compõem o veículo, desde suporte de bateria, suporte de coxins do motor etc. O desenho existente com o esquema de furações da longarina do chassi comercial está de acordo com a figura 53.

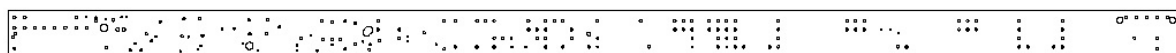


Figura 53 – Longarina Scania P440 CB 4x4. Fonte: Do autor

O material desta estrutura é o LNE 380 que segundo a ABNT NBR 6656 (2016) é um aço carbono de alta resistência com a tensão limite de escoamento entre 380-530 MPa, o material é parcialmente isotrópico. O material LNE 380 é utilizado nesta parte do veículo,

pois o chassi sofre diversos esforços ao longo de sua estrutura, como deformações a flexões, por torção, por fadiga etc. Segundo a norma [ABNT NBR 6656 \(2016\)](#), este material é fabricado por um processo de laminação a quente e possui uma massa específica de  $7,86 \text{ g/cm}^3$  e são produzidos nas espessuras de 1,20mm até 25,00mm, com larguras que podem variar de 750mm a 2100mm. As principais aplicações para este material são: fabricação de rodas, longarinas, equipamentos rodoviários, implementos / máquinas agrícolas. Um dos grandes comercializadores desta material no Brasil é a empresa Benafer, assim observamos na figura 54 a composição química e propriedades mecânicas deste material.

Especificação	Grau	Faixa de Espessura (mm)	Composição Química (% em massa)						Propriedades Mecânicas					Dobramento % mín. a 180°
									Ensaio de Tração			Alongamento		
			C	Mn	Si	P	S	Outros	Direção	LE (MPa)	LR (MPa)	Espessura (mm)	BM (mm)	
NBR6656 LNE	380	2,00-13,00	0,12 máx.	1,60 máx.	0,35 máx.	0,025 máx.	0,015 máx.	Al 0,015% mín.	Transversal	380-530	460-600	5,65	23	0,0 E

Figura 54 – Características LNE 380. Fonte: Site Benafer

A longarina do chassi do caminhão de bombeiro, quando o reservatório de água, de combustível etc., estão totalmente preenchidos, apresenta uma carga limite ao longo da longarina do chassi de 100 kN ou 10t por longarina, como o caminhão possui duas longarinas equidistantes, logo a carga limite suportada da carroceria e fluidos é de 20t. A figura 55 demonstra a distribuição destas cargas ao longo do chassi.

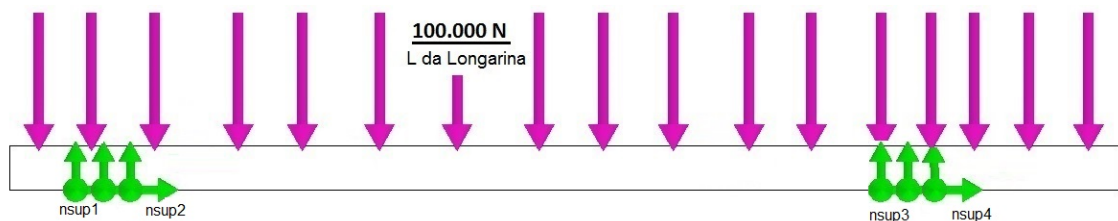


Figura 55 – Cargas e apoios da estrutura.  $L = 6.833 \text{ mm}$ . Carga = 100.000N. Não há simetria na estrutura. Fonte: Do autor

Como observado na figura o eixo é atravessado sob a longarina do chassi nos extremos transversais do quadro do chassi. Desta forma considera-se 2 pontos (nsup1:nsup2) e (nsup3:nsup4) para as fixações da longarina do chassi.

Além disto, como o perfil da longarina é tridimensional, a carga distribuída deve representar a força em uma face planar bidimensional, no entanto conforme observamos na figura 56 para transformarmos as cargas de uma estrutura tridimensional para uma face planar, seria como se tivéssemos um momento torsor sobre a face superior da alma da viga. Ocorre que tanto os algoritmos de [Sigmund \(2001\)](#), quanto o algoritmo de [Challis e Guest \(2009\)](#) utiliza elementos para o FEA que apenas tem dois DOFs por nó. No caso discutido não será possível portanto utilizar a face planar com a condição de contorno de momento torsor no lugar da carga distribuída.

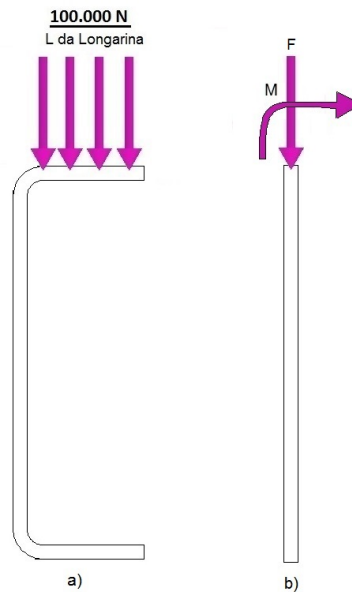


Figura 56 – a) Representação do perfil 3D b) e perfil 2D para as mesmas condições de contorno. Fonte: Do autor

Uma alternativa possível de se obter a carga sobre a face planar que representa a carga distribuída sobre o perfil 3D, é equilibrarmos a mesma linha elástica para os dois perfis, deixando a carga vertical superior como incógnita, ou seja, encontramos qual é a linha elástica do perfil tridimensional para a carga de  $100.000 \text{ N} / \text{L}$ , e alteramos a carga do perfil planar para que contenha a mesma linha elástica. Um programa utilizado no meio acadêmico para obter a linha elástica das vigas é o FTOOL (Tecgraf-PUC Rio), versão educacional 3.1. Com este programa foi possível inserir os dois domínios da figura 56 na viga, e obter a linha elástica. Na 57 é observado portanto a linha elástica para o perfil tridimensional

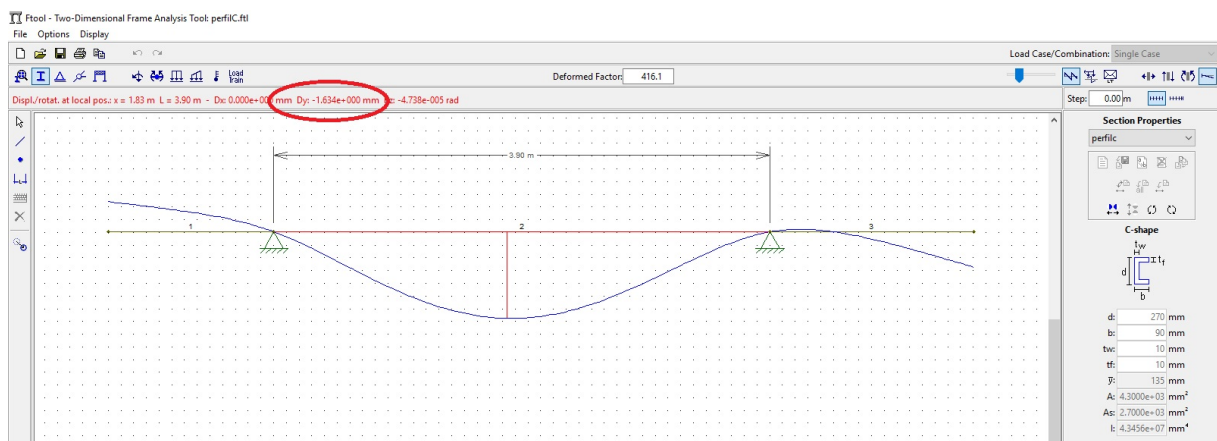


Figura 57 – Linha Elástica do Perfil Tridimensional. Fonte: Do autor

Como resposta para o perfil da figura 56 a) e conforme o cálculo da linha elástica da viga conforme figura 57 obtemos o valor de 1,634 mm, este valor equivale a uma



carga distribuida de 100kN na sobre a aba de todo o comprimento da longarina do chassi comercial.

Sabe-se que quando eliminamos a aba o perfil fica mais frágil, pois as abas de uma longarina, além de estarem configuradas para apoio de quadro do chassi e de demais componentes, também ajuda a aumentar o  $W_x$  (módulo de resistência) do perfil estrutura. Obtemos portanto alguns valores para a linha elástica com o perfil bidimensional planar esperando valores de carga inferiores à 100kN, conforme tabela 12.

90kN	3,897mm
80kN	3,466mm
70kN	3,030mm
60kN	2,598mm
50kN	2,166mm
40kN	1,731mm
30kN	1,299mm

Tabela 12 – Valores de linhas elástica do perfil bidimensional para várias cargas. Fonte: Do autor.

Para chegarmos na altura de 1,634 mm por meio da linha elástica conforme figura 57, podemos interpolar os valores de 30 e 40kN encontrados e que estão descritos na tabela 12. Desta forma, obtemos o valor de 37,75kN para a carga aplicada na vertical para o perfil bidimensional. Na figura 58 observamos este resultado.

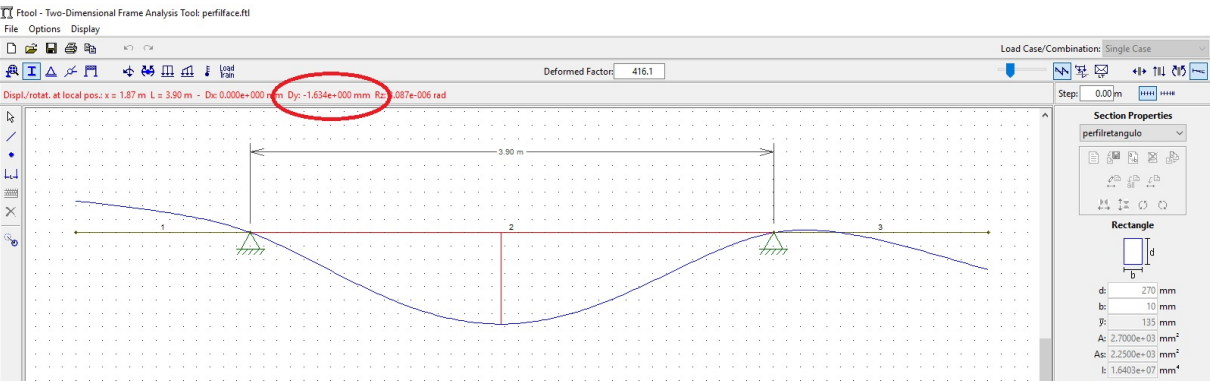


Figura 58 – Linha Elástica do Perfil Bidimensional. Fonte: Do autor

Portanto, a partir daqui utilizaremos os dados de carga vertical para os casos bidimensionais e tridimensionais conforme a tabela 13.

2D	37,75kN
3D	100kN

Tabela 13 – Valores de cargas para o caso 2D e o 3D. Fonte: Do autor.

Para conseguirmos comparar o algoritmo com as sintetizações executadas pelos programas computacionais comerciais adotamos alguns padrões para os cálculos, um dos padrões é a definição do mesmo número de elementos  $N$  para a malha que no caso definimos como 10.000 elementos devido ao fato da restrição do número de elementos para a versão estudantil do programa computacional ANSYS Mechanical. Este programa liberado para pesquisas academicamente possui em sua licença de uso uma limitação de 10.000 elementos e de 32.000 nós, porém esta limitação não influenciou significativamente os cálculos pois a longarina estudada apresenta um comprimento grande em relação a pequenas peças, e por experiência do autor é possível trabalhar com esta quantidade de elementos e que gera um tamanho de elemento compatível para a análise FEA. Assim com base neste valor e com base nas dimensões do chassi da figura 39 em que temos as dimensões comprimento (L) x altura (h), conseguimos achar o valor de  $x$ , que seria a dimensão do comprimento e dos lados do elemento conforme figura 59.

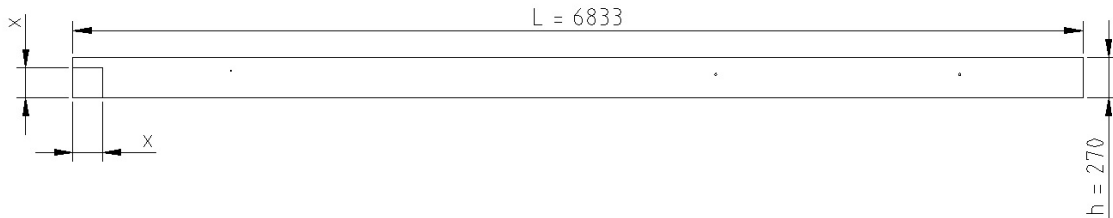


Figura 59 – Dimensão do elemento da estrutura. Fonte: Do autor

Para definirmos o  $nelx$  e  $nely$  à serem utilizados para o objeto de estudo, com forma quadrática com lados iguais (base é igual a altura), temos o valor de  $x$  conforme equações:

$$\frac{L}{x} = nelx; \quad \frac{6833}{x} = nelx; \quad (7.7)$$

$$\frac{h}{x} = nely; \quad \frac{270}{x} = nely; \quad (7.8)$$

$$nelx \times nely = 10000 \quad (7.9)$$

Resolvendo um sistema para encontrar a variável  $x$ , temos da equação 7.9 em 7.8.

$$\frac{270}{x} = \frac{10000}{nelx}; \quad (7.10)$$

Da equação 7.10 em 7.7, encontramos a equação:

$$x^2 = (6833 \times 270)/10000 \quad (7.11)$$

Assim, o valor de  $x$  que é o tamanho de cada elemento tem a dimensão de:

$$x \approx 14\text{mm} \quad (7.12)$$

Portanto da equação 7.12 em 7.7, temos o número de elementos em  $x$ :

$$nelx \approx 503 \quad (7.13)$$

E da equação 7.12 em 7.8, temos o número de elementos em  $y$ :

$$nely \approx 20 \quad (7.14)$$

Observa-se que as equações 7.12, 7.13 e 7.14 tiveram que ser arredondadas, pois os algoritmos permitem apenas que sejam introduzidos números inteiros. Assim os valores de  $nelx$  e  $nely$  serão considerados como referência a medida real existente no chassi comercial. Desta forma, com os valores encontrados nas equações para encontrar  $nelx$  e  $nely$ , temos as dimensões e posicionamento dos nós conforme figura 60.



Figura 60 – Posicionamento dos nós. Fonte: Do autor

E por comparação com a figura 60 e as figuras 41 e 42, podemos ter as condições de contorno para os valores da posição dos nós dos pontos fixos, conforme figuras 61 e 62.

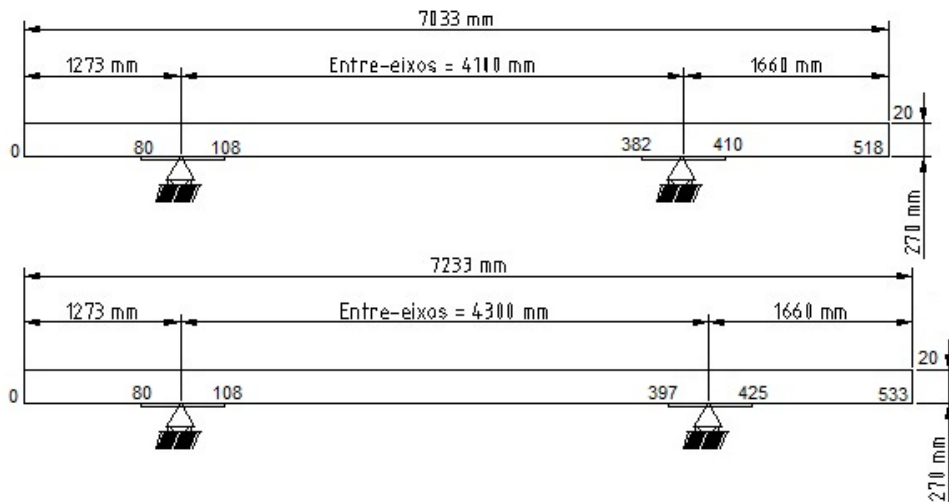


Figura 61 – Posicionamento dos nós em EE 4.100 e 4.300mm. Fonte: Do autor



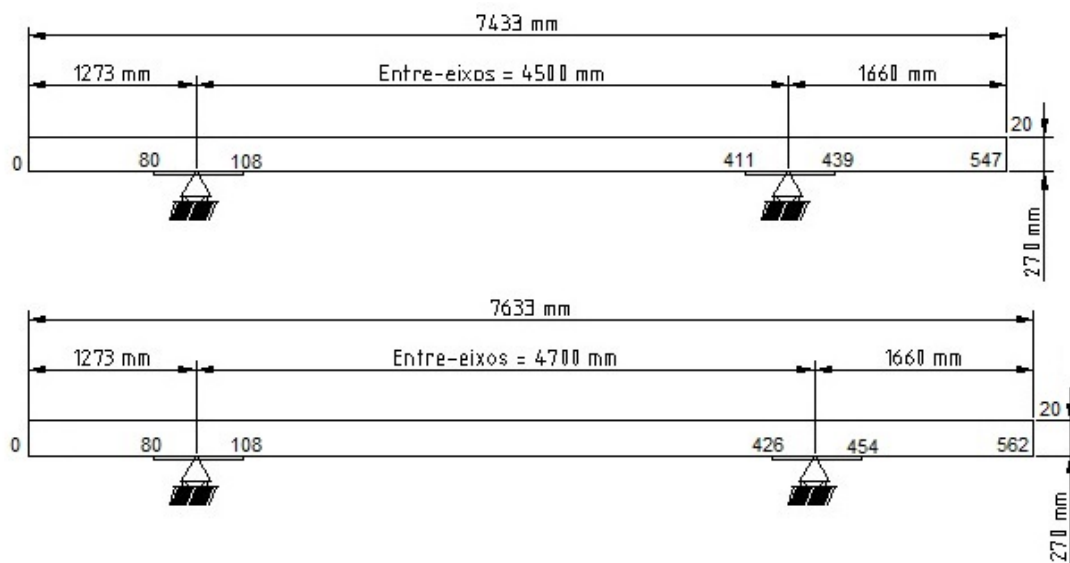


Figura 62 – Posicionamento dos nós em EE 4.500 e 4.700mm Fonte: Do autor

Assim, de forma resumida, os dados podem ser coletados com o novo posicionamento de nós da tabela 14.

Tabela 14 – Lista de dados de entrada requeridos pelos algoritmos

Variáveis de entrada	Caso 1	Caso 2	Caso 3	Caso 4	Caso 5
Entre-Eixo (mm)	3.900	4.100	4.300	4.500	4.700
Nelx (adm.)	503	518	533	547	562
Nely (adm.)	20	20	20	20	20
Nsup1 (adm.)	80	80	80	80	80
Nsup2 (adm.)	108	108	108	108	108
Nsup3 (adm.)	367	382	397	411	426
Nsup4 (adm.)	395	410	425	439	454
volfrac ou VolREq	0,7	0,7	0,7	0,7	0,7
Penal (adm.)	3,0	3,0	3,0	3,0	3,0
rmin (mm)	1,5	1,5	1,5	1,5	1,5
stepLength (s)	3	3	3	3	3
numReinit (Quant.)	2	2	2	2	2
topWeight (%)	2	2	2	2	2

## 7.5 Análise de dados

### 7.5.1 Coletados dos algoritmos

Com base nos algoritmos propostos e para comparar todos os parâmetros nas 5 análises (2 nos algoritmos e 3 nos programas computacionais), temos a proposta da equação 7.15 para o comando do código e casos 1,2,3,4 e 5, respectivamente. A descrição de cada uma destas variáveis é denominada na tabela 15. Estes parâmetros são diretamente inseridos nos algoritmos e implicitamente configurados nos programas computacionais.

Tabela 15 – Requisitos de entrada requeridos pelo programa

Dado de entrada	Descrição
Nelx	Número de elementos na direção x.
Nely	Número de elementos na direção y.
volfrac	Restrição de volume.
Penal	Penalização para o SIMP.
rmin	Raio de atuação do filtro.
stepLength	Intervalo de tempo da evolução da função de LS.
numReinit	Frequência de reinicialização.
topWeight	Parâmetro de redução de peso.

$$\begin{aligned}
 & \text{topopt}(503, 20, 0.7, 3.0, 1.5) \\
 & \text{topopt}(518, 20, 0.7, 3.0, 1.5) \\
 & \text{topopt}(533, 20, 0.7, 3.0, 1.5) \\
 & \text{topopt}(547, 20, 0.7, 3.0, 1.5) \\
 & \text{topopt}(562, 20, 0.7, 3.0, 1.5)
 \end{aligned} \tag{7.15}$$

E para o código de LS implementado temos a função de entrada para os cinco casos descritos na equação 7.16.

$$\begin{aligned}
 & \text{top\_levelset}(503, 20, 0.7, 3, 2, 2) \\
 & \text{top\_levelset}(518, 20, 0.7, 3, 2, 2) \\
 & \text{top\_levelset}(533, 20, 0.7, 3, 2, 2) \\
 & \text{top\_levelset}(547, 20, 0.7, 3, 2, 2) \\
 & \text{top\_levelset}(562, 20, 0.7, 3, 2, 2)
 \end{aligned} \tag{7.16}$$

### 7.5.2 Coletados do algoritmo SIMP implementado

Assim, para os 5 casos de EE para o algoritmo de SIMP com diferentes medidas para o veículo, obtivemos os seguintes resultados conforme figura 63.

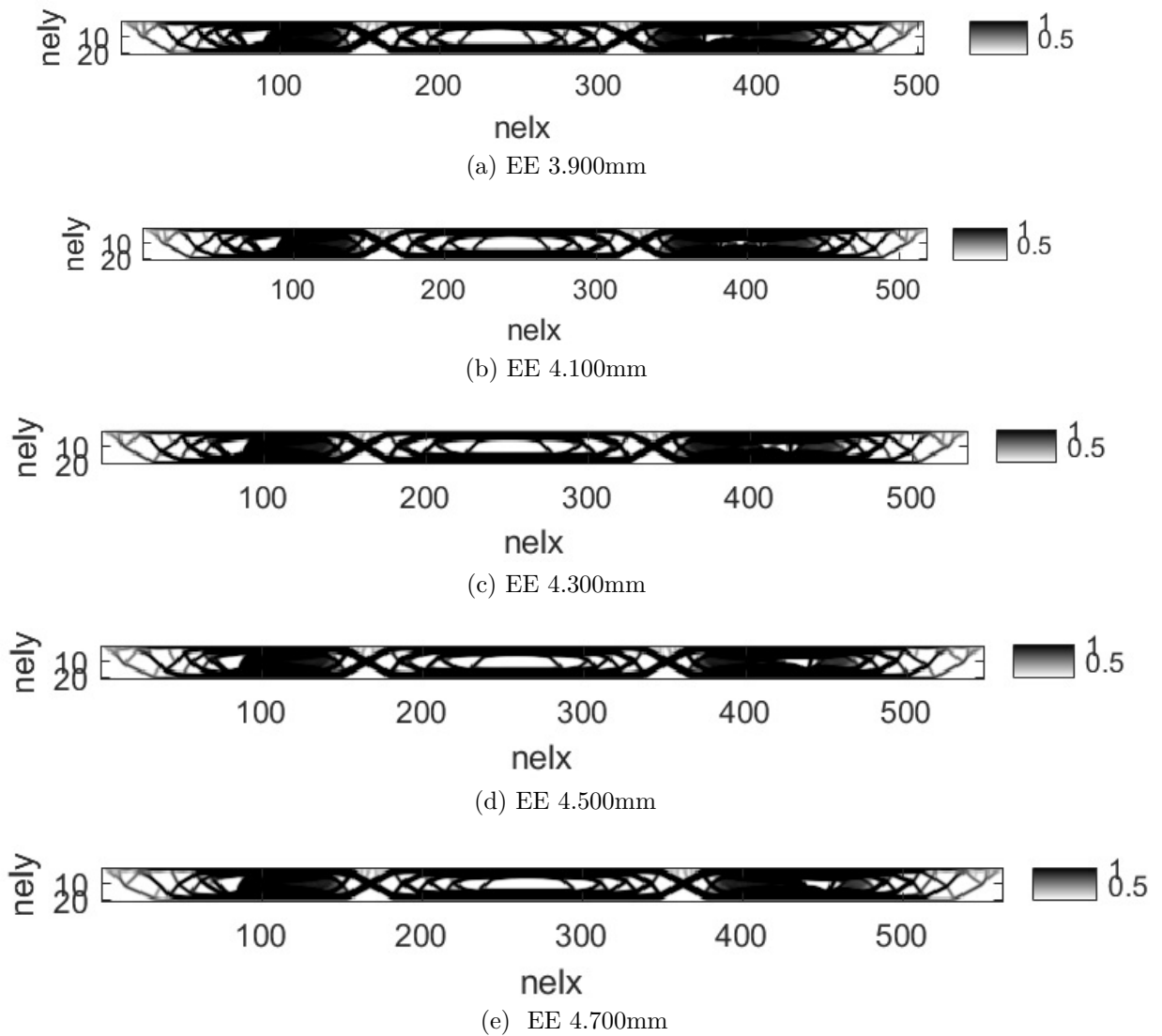


Figura 63 – Algoritmo SIMP sintetizado com  $V=70\%$ . Fonte: Do Autor.

Para o algoritmo implementado SIMP, foi alterado o valor da convergência do *step* de 0,01 para 0,1 fazendo com que aceleramos um pouco o processo de sintetização.

Observa-se que em todas as configurações, foram encontradas os denominados *hinges*, que são regiões enrijecidas que se entrelaçam ligados por elementos que tem o formato de uma figura 'X' conforme figura 64. Mais à frente iremos investigar e discutir porque isto ocorre com estes algoritmos de Sigmund (2001) e de Challis e Guest (2009).

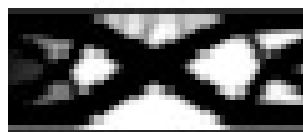
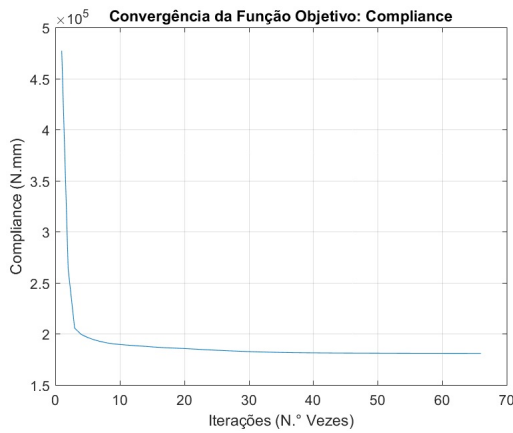


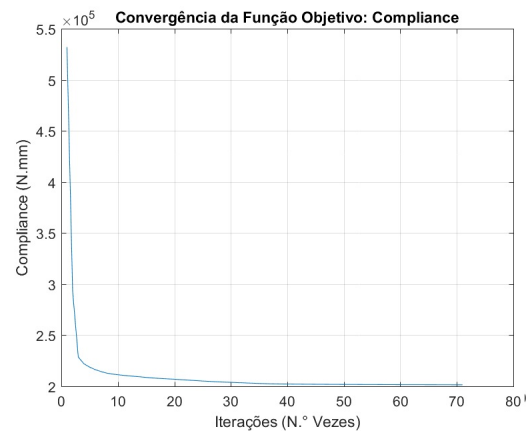
Figura 64 – Aparecimento do formato *hinge*. Fonte: Do autor

Foram coletados os gráficos de variações de Iterações x Compliance, assim demons-

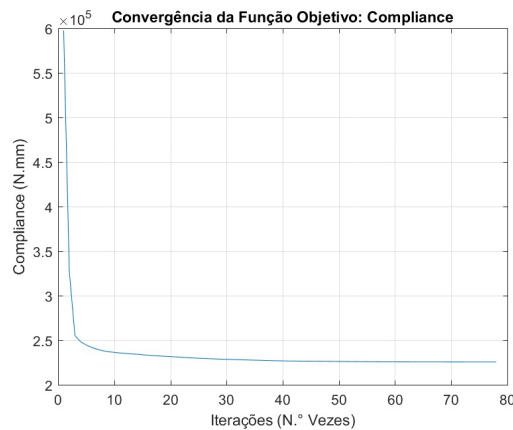
tramos cada um deles na figura 65.



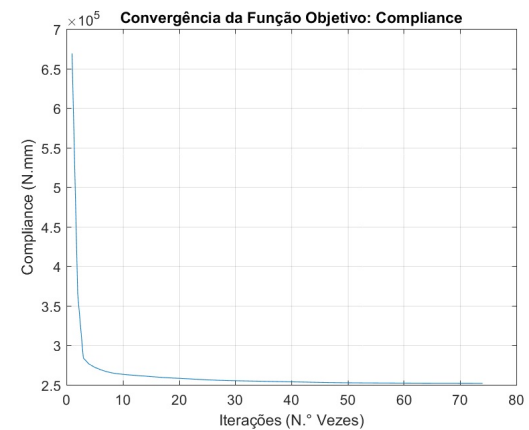
(a)  $c_0=4,77e+05$ ,  $c_1=1,81e+05$ , 66 iterações e 853s.



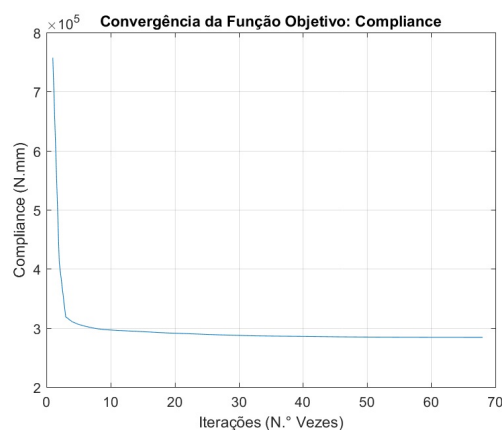
(b)  $c_0=5,32e+05$ ,  $c_1=2,01e+05$ , 71 iterações e 948s.



(c)  $c_0=5,98e+05$ ,  $c_1=2,25e+05$ , 78 iterações e 1117s.



(d)  $c_0=6,69e+05$ ,  $c_1=2,52e+05$ , 74 iterações e 1103s.



(e)  $c_0=7,57e+05$ ,  $c_1=2,84e+05$ , 68 iterações e 1069s.

Figura 65 – Chassi sintetizado para: a) EE 3.900mm com  $N=503 \times 20$ , b) EE 4.100mm com  $N=518 \times 20$ , c) EE 4.300mm com  $N=533 \times 20$ , d) EE 4.500mm com  $N=547 \times 20$ , e) EE 4.700mm com  $N=562 \times 20$ ,  $V=70\%$ ,  $p=3$ ,  $r_{\min}=1,5$ . Fonte: Do Autor.

Nos gráficos da figura 65 é possível observar que o menor valor de compliance é

com longarina com EE de 3.900mm, quando vai aumentando o comprimento da longarina progressivamente (4.100mm, 4.300mm, 4.500mm, 4.700mm) o valor mínimo de compliance é maior, isto se deve porque com comprimentos maiores, mais energia de deformação terá a estrutura. Observa-se uma similaridade entre todos os comprimentos de longarina, já que as condições de contorno não foram trocadas, apenas houve alteração na posição dos pontos móveis, que no caso seriam os eixos direcional e traseiro do veículo. Nas comparações computacionais com os algoritmos e com os programas computacionais, limitaremos o EE com 3.900mm.

É importante observarmos as etapas e a velocidade do algoritmo para cada análise, os números de iterações e os tempos coletados para os 5 tipos de EE, que estão demonstrados na tabela 16.

Tabela 16 – Tempo e Número de Iterações

EE (mm)	N. Iterações	Tempo (s)
3.900	66	853
4.100	71	948
4.300	78	1117
4.500	74	1103
4.700	68	1069

Todas as análises tiveram uma variação de tempo entre 14 a 19 minutos para resolver cada uma delas. O número de iterações foi bem próximo, observa-se que independente do fato de aumento do EE, não há um significativo aumento de número de iterações e de tempo que injustifique o uso do método para quaisquer que sejam os comprimentos da longarina do chassi dos diversos veículos comerciais existentes no mercado. Foi também observado para estes 5 casos de EE que pequenas oscilações ocorrem na discretização da fração de volume ao longo das iterações, inicialmente a sintetização já converge para a proximidade da fração de volume desejada e ao longo das iterações fica alternando até convergir para uma solução ótima.

Além destas análises foi possível observar a variação do número de iterações pela fração de volume desejada  $V(x)$  em diversos casos. Em que observamos um decaimento até a convergência. Os dados avaliados foram de restrição para um volume de 70%, 75%, 80%, 85% e 90%.

Observa-se portanto por meio da figura 66, que o método SIMP, apresenta uma similaridade na sintetização por meio de frações de volume. Esperava-se obter mudanças bruscas e até a eliminação do formato *hinge*, no entanto para esta condição de domínio permanece este formato em todas as frações de volume desejadas.

As diversas frações de volume que resultam a figura de otimização poderão posteriormente serem simuladas por elementos finitos para comprovarmos o critério de aprovação de rigidez da condição imposta à estrutura.

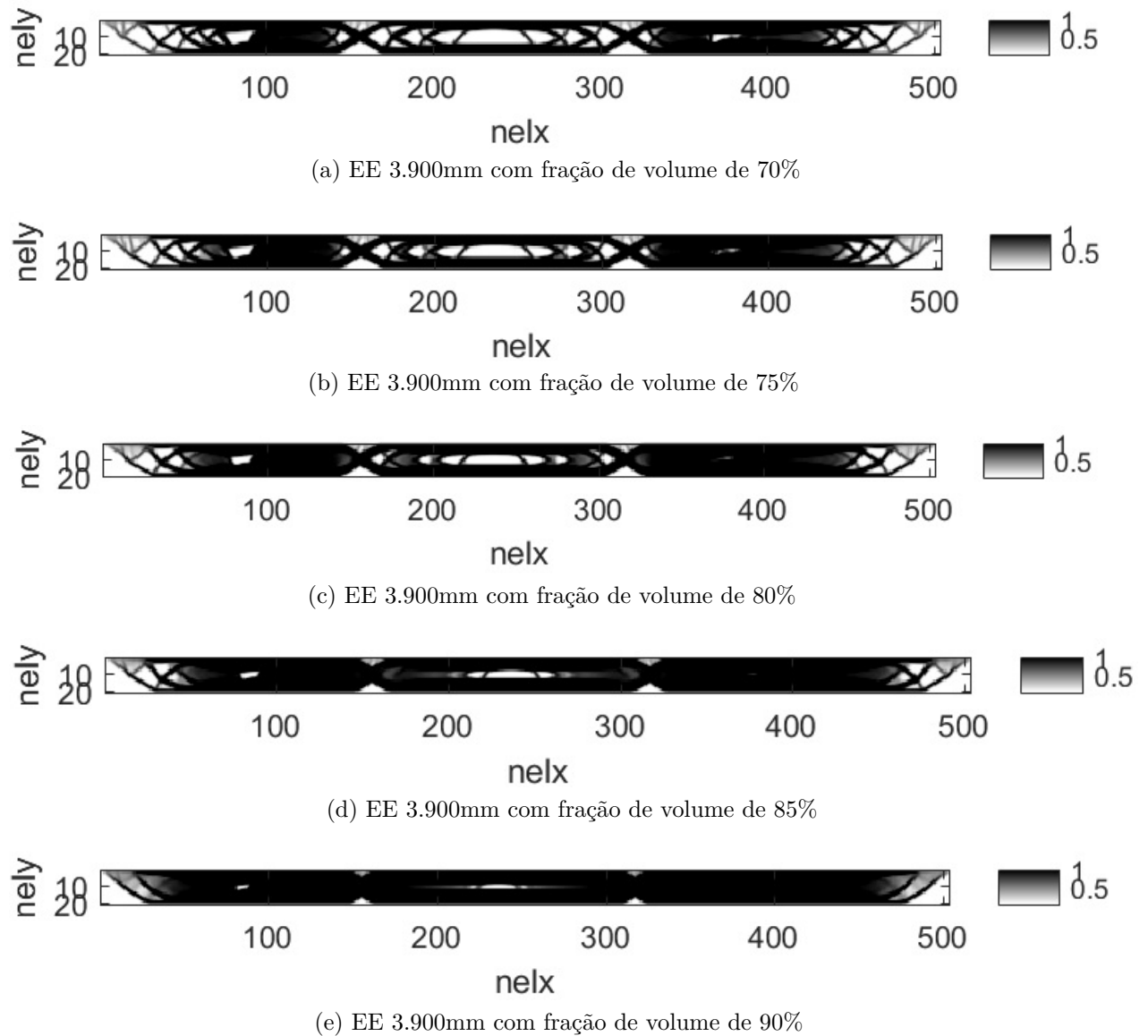


Figura 66 – Sintetização SIMP implementado com diversas frações de volume. Fonte: Do Autor.

Na figura 66 observamos o aumento da região escura 1, conforme aumentamos a fração de volume. Valores entre 85% e 90% não se tornam interessantes pois o número de "buracos" (espaços vazios com região clara) no domínio é pequeno ao longo da estrutura. No entanto observamos que para qualquer fração de volume as extremidades da longarina tem sempre a criação de buracos, demonstrando o quão baixo é o nível de energia nestas regiões.

Ainda com este algoritmo SIMP procura-se estudar e investigar o porque da formação das regiões de *hinges* entre os eixos da estrutura conforme segue.

### 7.5.3 Investigação da região de formação de *hinges* na estrutura

Observamos que a longarina do chassi comercial estudada tem uma relação de aspecto de 26:1 conforme já discutimos no capítulo 4.4, ou seja, o comprimento da longarina é 26 vezes maior do que a largura ou altura dela. Não foi encontrado fontes de referência em que se discuti relações de aspecto especificamente para a aplicações em longarinas, no entanto outras referências foram encontradas para outras aplicações. Assim verifica-se que outras aplicações trabalham com relações de aspecto iguais ou menores do que esta como no caso de asa de aeronaves. Procura-se portanto, a partir desta relação de aspecto da longarina comercial estudada de 26:1, analisar outras relações considerando o mesmo tipo de domínio, que tem carga distribuída na parte superior e dois pontos fixos equidistantes na parte inferior. Espera-se que haja uma relação ideal em que os *hinges* deixem de existir. Adotamos a variação da longarina na horizontal com a letra "a" e a vertical com a letra "b", portanto temos uma relação 26a:1b. Na figura 67 temos as relações de variações para a medida "a".

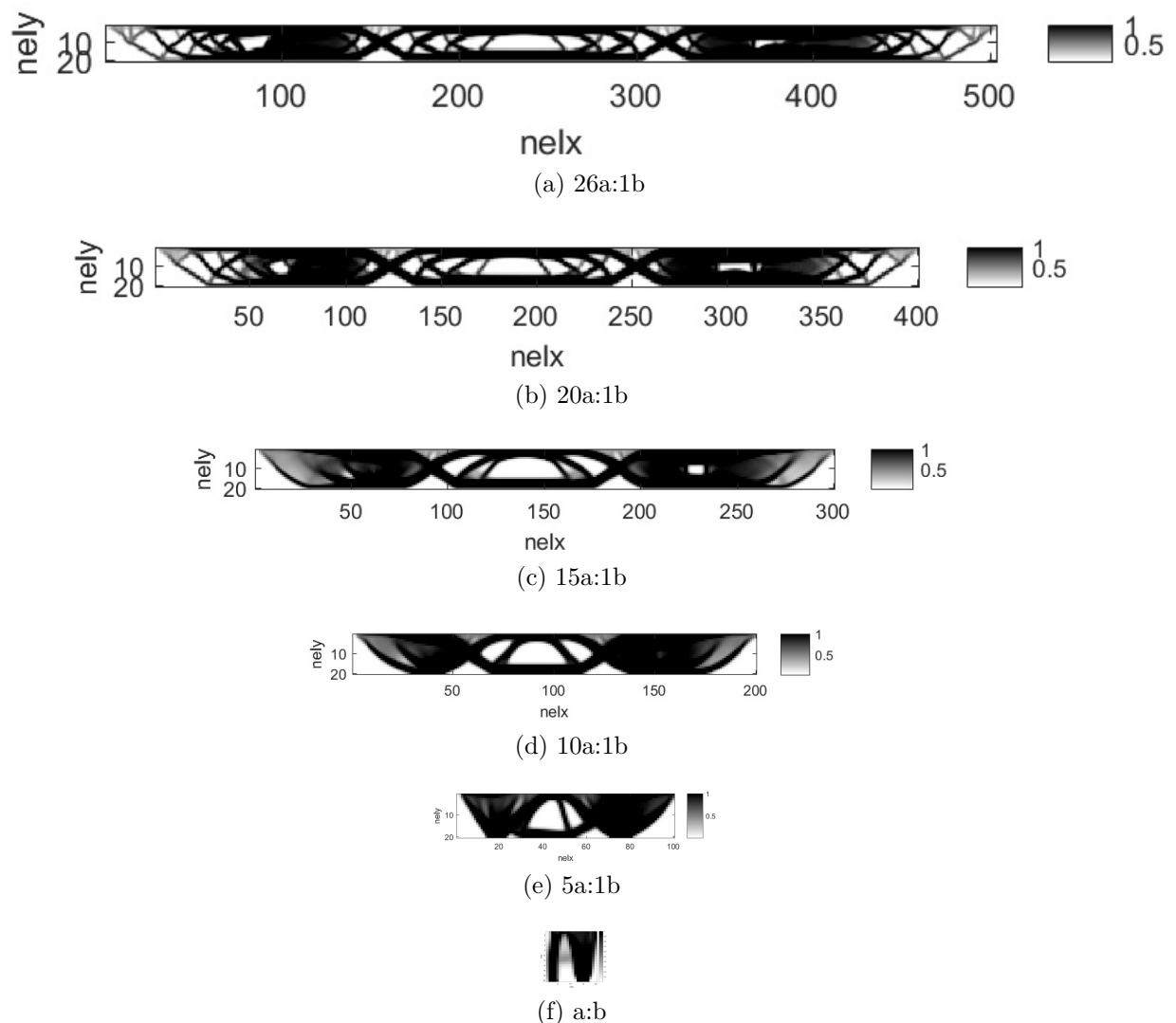


Figura 67 – Modelo padrão para relações de aspecto da longarina. Fonte: Do Autor.

Também observamos a relação de aspecto para o modelo padrão, mantendo o valor de "a" e alterando o valor de "b". Na figura 68 encontramos as figuras sintetizadas.

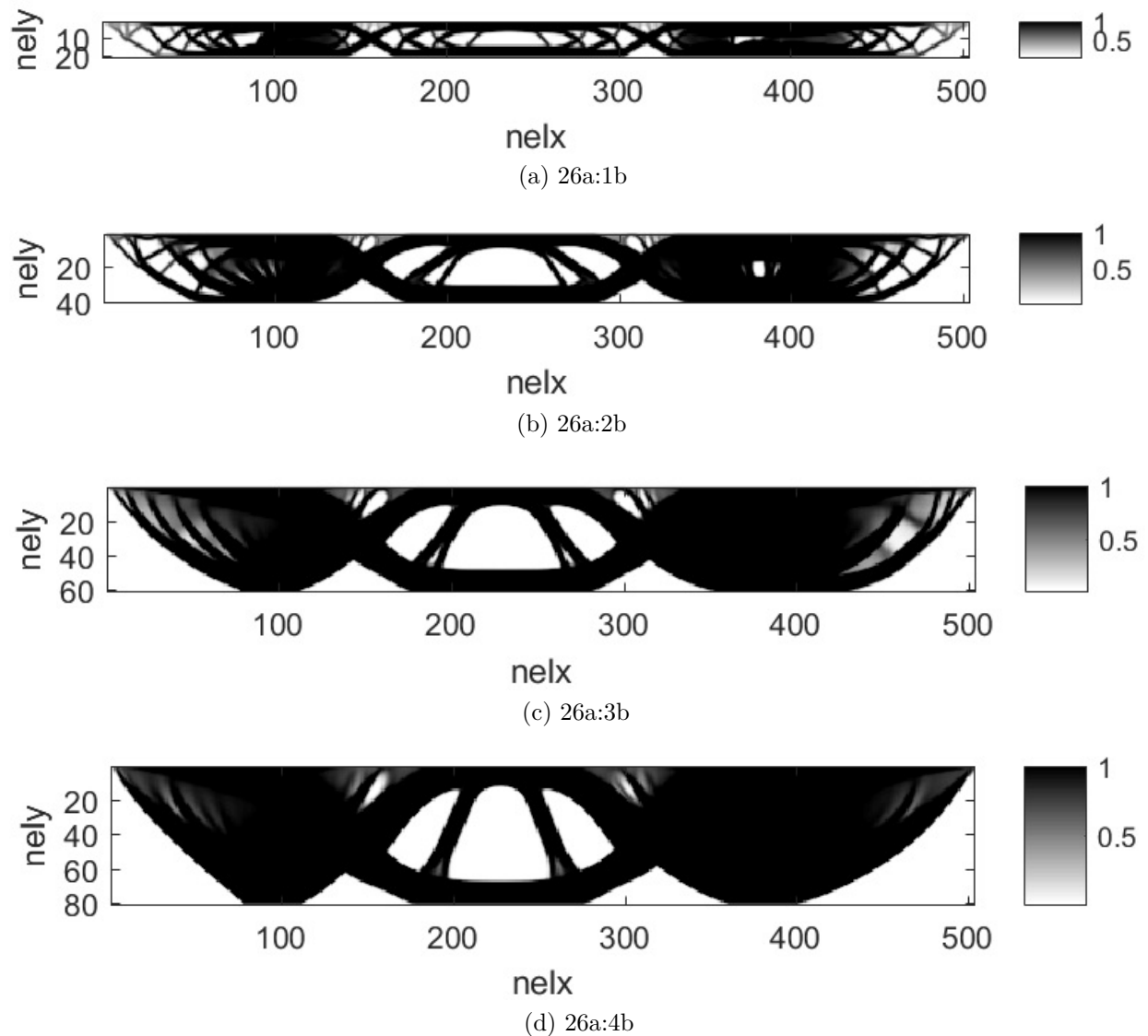


Figura 68 – Modelo padrão para relações de aspecto da longarina. Fonte: Do Autor.

Conforme observamos nas figuras 67 e 68, quando diminuimos a razão "a" mantendo "b", ou quando mantemos a relação da razão "a" da estrutura e aumentamos o "b", existe um ponto ou relação de aspecto em que o hinge desaparece, a relação é 5:1 ou 26:4, ou seja a relação de aspecto para que este algoritmo trabalhe apresentando todo o domínio sintetizado corretamente deveria ser entre 5 e 6:1, portanto verifica-se um problema para a utilização deste algoritmo para esta aplicação de longarina de chassi específica para este veículo em estudo. E que para outros veículos, novamente deve ser analisado a relação de aspecto, fazendo com que este algoritmo trabalhe com a discretização do domínio de forma correta.

A regiões dos "hinges" aparece provavelmente, pois em todos os algoritmos estudados, Sigmund (2001) com modelo SIMP ou Challis e Guest (2009) consideram a seção do domínio



no algoritmo sendo retangular, se a seção é retangular, o objeto parece não servir para o objeto de estudo que é uma longarina de chassi que tem um perfil no formato de um "C", diferente de retangular. A seção no perfil C tem um momento de inércia diferente do perfil retangular, que foi encontrada para valores diferentes de carga discutidos nos capítulos anteriores. O modelo 2D portanto não é compatível para o objeto de estudo. Porém as regiões de hinge formam uma rótula, por exemplo nos mecanismos flexíveis que são um outro tipo de domínio tem-se um movimento de ligação que forma regiões por flexibilidade, alguns outros exemplos podem ser discutidos como um joelho de um corpo humano, ou uma dobradiça de porta, que é uma região que facilite a ter-se rotações. Como trabalhamos com formulação de rigidez, a forma que se encontra é retirar materiais de certas regiões para tornar-se menos rígidas em relação a outras regiões, assim o efeito dobradiça aparece pelo fato da linha elástica que precisa formar uma curvatura entre os dois eixos do domínio por exemplo entre as fixações do eixo dianteiro e eixo traseiro. Se comparar os hinges com a linha elástica, os pontos de inflexão são coincidentes com os hinges, na frente e atrás dos eixos dianteiros e traseiros tem-se um ponto de inflexão com a concavidade para cima, entre os pontos tem uma concavidade com um ponto central, assim neste ponto central que há uma grande inflexão comparado com outros pontos há a necessidade da formação do hinge, e só é possível fazer esta curvatura se deixar a região central como uma rótula, ou seja, sem material.

Assim, os hinges no 2D aparecem, no 3D tem uma configuração diferente e não aparece hinge pois a aba superior reforça a região, desta forma visualizando-se os domínios 2D e 3D e comparando-os com a região de aparecimento do hinge, tem-se o melhor resultado com o objeto de domínio 3D.

Foram verificados outros algoritmos educacionais existentes, como por exemplo o algoritmo de 88 linhas de [Andreassen et al. \(2011\)](#) uma evolução frente ao algoritmo de [Sigmund \(2001\)](#) de 99 linhas, assim apresentou-se o formato de *hinge* de enriquecimento da estrutura, tanto para o filtro de sensibilidade, quanto para o filtro de densidade, e também com filtro PDE, filtro de projeção Heaviside e com a função `conv2` do MATLAB, que são outras técnicas demonstradas por este autor para a aplicação deste algoritmo. Desta forma pode-se concluir que o formato de enriquecimento da estrutura (aparecimento de uma função tipo rótula) no ponto de *hinge* é padrão em todos os algoritmos 2D que apresentam o elemento quadrilátero e que para se ter um formato com o preenchimento do domínio sobre este *hinge* deve-se modificar as relações de aspecto conforme discutidos acima, ou utilizar o domínio no formato 3D apenas.

#### 7.5.4 Coletados do algoritmo LS implementado

Apresentamos o algoritmo implementado com o LS. O modelo sintetizado apresentado para os 5 EE estão diagramados de acordo com a figura 69. Observa-se que assim

como o SIMP este também produz *hinges* na estrutura, e aparece de uma forma mais acentuada já que o LS não trabalha com elementos intermediários de escala de cinza.

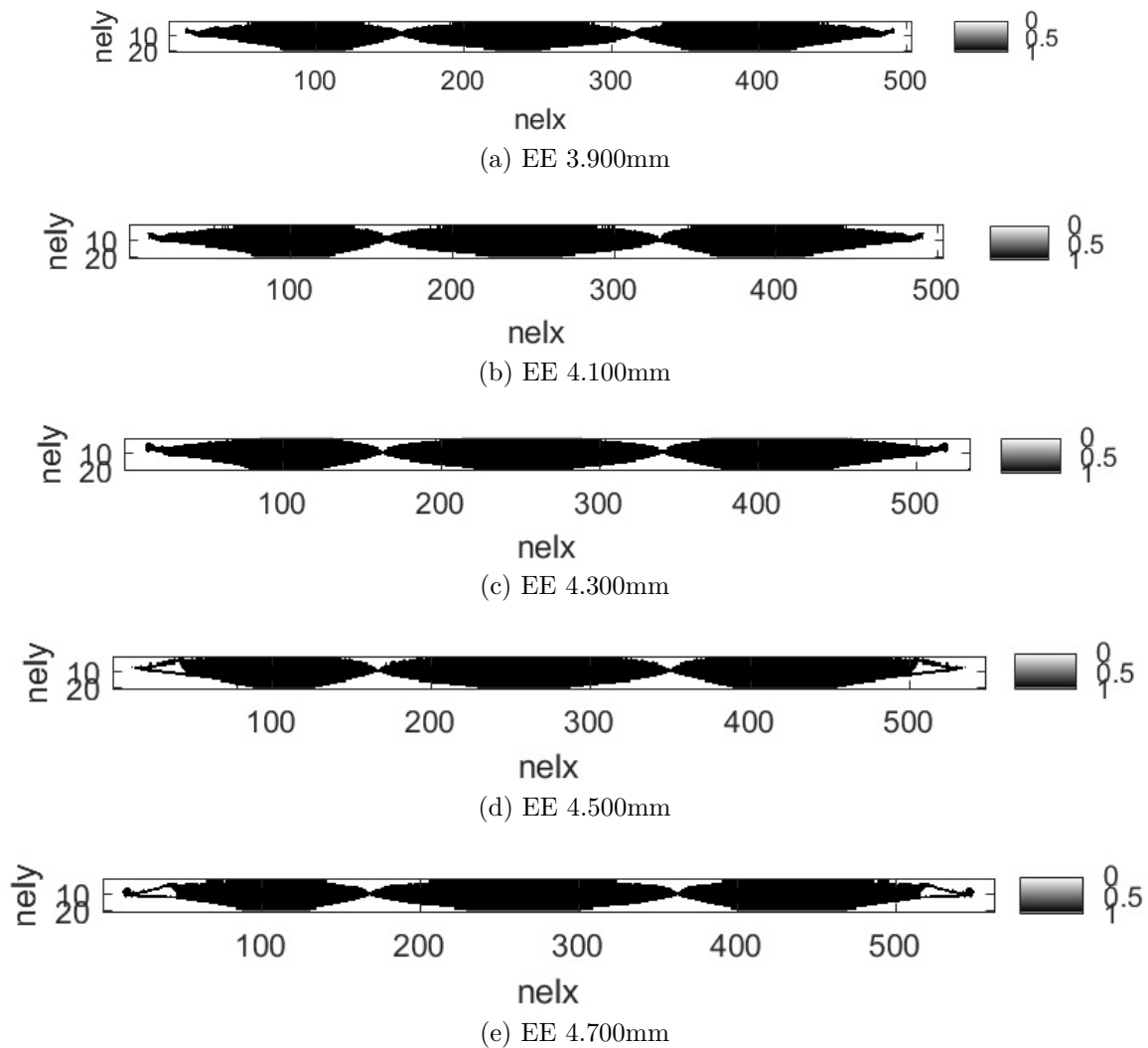
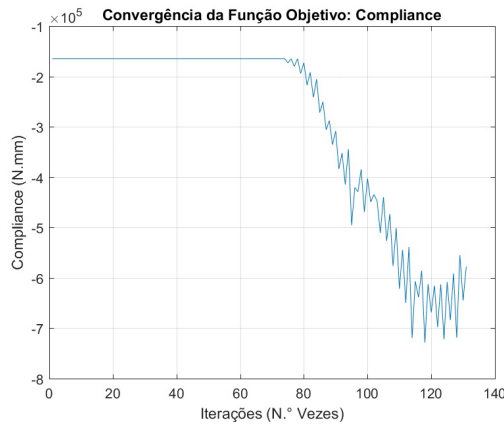


Figura 69 – Algoritmo LS sintetizado com  $V=70\%$ . Fonte: Do Autor.

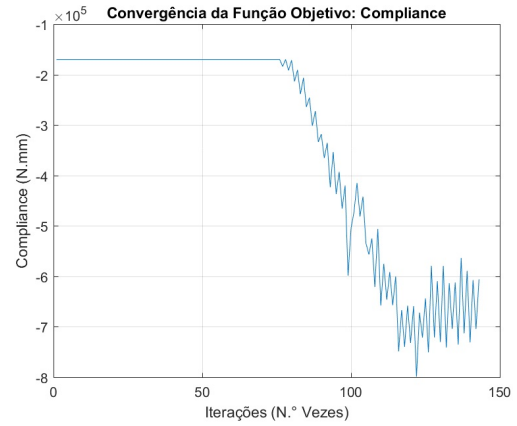
É possível verificar na figura 69, que os 5 casos estudados com a abordagem LS tem formas finais equivalentes. Nos pontos fixos e na parte superior apresenta cores escuras significando que nesta localização a região tem o valor de densidade 1. Assim como vimos no algoritmo SIMP implementado o formato de *hinge* em X que é apresentado entre os EE e que tem a cor vazia na face inferior é factível, porém próximo a face superior deveria ser encontrado um valor escuro ao invés de branco, pois naquela posição não deveremos introduzir novos furos na topologia.

O valor de compliance das análises decresce durante as iterações até obtermos um valor mínimo de compliance. Sendo assim a figura 70 demonstra estes gráficos.

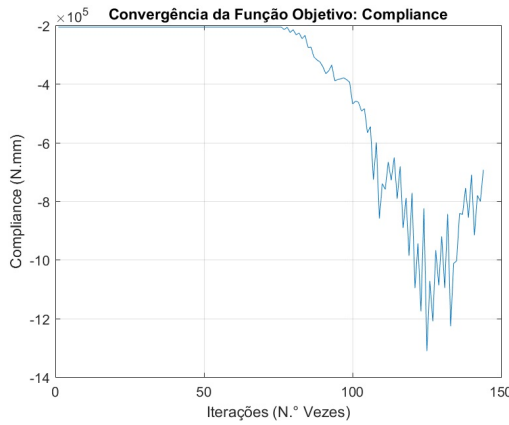
O valor de compliance vai diminuindo a medida que o EE aumenta, e com aproximadamente 120 iterações ele gera um pico e começa a alternar a condição de energia para restringir para o volume requerido. É possível observar que o algoritmo de LS só começa a



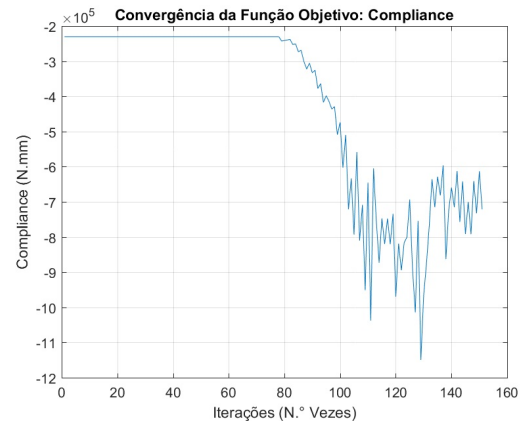
(a)  $c_0 = -1,65e+05$ ,  $c_1 = -5,78e+05$ , 131 iterações e 1320s.



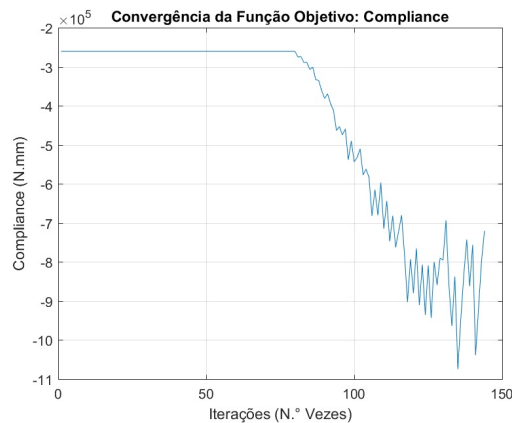
(b)  $c_0 = -1,70e+05$ ,  $c_1 = -6,06e+05$ , 143 iterações e 1676s.



(c)  $c_0 = -2,06e+05$ ,  $c_1 = -6,93e+05$ , 144 iterações e 2127s.



(d)  $c_0 = -2,30e+05$ ,  $c_1 = -7,21e+05$ , 151 iterações e 2163s.



(e)  $c_0 = -2,61e+05$ ,  $c_1 = -7,20e+05$ , 144 iterações e 2190s.

Figura 70 – Chassi sintetizado para: a) EE 3.900mm com  $N=503 \times 20$ , b) EE 4.100mm com  $N=518 \times 20$ , c) EE 4.300mm com  $N=533 \times 20$ , d) EE 4.500mm com  $N=547 \times 20$ , e) EE 4.700mm com  $N=562 \times 20$ ,  $V=70\%$ ,  $\text{steplength}=3$ ,  $\text{numReinit}=2$  e  $\text{topWeight}=2$ . Fonte: Do Autor.

gerar valores significativos de mudança a partir de 100 iterações. Nos gráficos das figuras 70 e 71, conseguiremos comparar a curva de compliance com a curva de fração de volume,

que mostra que a restrição está ativa no final da OT.

Diferentemente do SIMP, tem-se um valor negativo para o compliance, que decresce a medida que ocorre a sintetização, este valor de sinal negativo não poderia existir já que o compliance é a somatória da matriz de rigidez pelos deslocamentos, que gera ao aumento da rigidez com a minimização da função, no entanto originalmente o algoritmo de LS de Challis e Guest (2009) trabalha com uma função na linha 19 do algoritmo de -max, e a função objetivo da linha 25 aparece também como -sum (-max), ou seja, o sinal negativo encontrado nestas duas linhas faz com que a energia de deformação compliance aumente ao longo das iterações ao invés de diminuir, isto é uma condição errada pensando-se que o valor de compliance tem que diminuir a energia de deformação. para corrigir isto, de forma que tenhamos um valor de energia de deformação decrescendo, alterou-se a linha 19 do algoritmo de -max para min, e manteve-se a linha da função objetivo como -sum (min), desta forma os valores da convergência da função objetiva diminuem, porém com sinal negativo. Há uma falha no algoritmo para isto, pois se alterarmos a condição da função objetivo para um sinal positivo ao invés de negativo, denotando a função como +sum (min), que seria o real, a função começa a crescer o nível de energia de deformação e isto torne-se errado pois volta a condição inicial do algoritmo. Portanto, mantêm-se a função com sinal negativo, apenas como critério de qualidade para visualizarmos como a energia de deformação do compliance trabalha ao longo das iterações para compararmos com o algoritmo SIMP, porém para sabermos o delta da real energia de deformação, deveremos verificar a diferença da energia final para a energia inicial, e assim obtermos o valor real e positivo da energia de deformação dos gráficos da figura 70.

Observou-se que na figura 71, que também igualmente como vimos nos gráficos de compliance, a restrição só começa a agir após 100 iterações, porém após este número a função age rapidamente.

Na tabela 17 demonstra-se o número de iterações e o tempo utilizados para estas convergências com base no método LS.

Tabela 17 – Número de Iterações e Tempo

EE (mm)	N. Iterações	Tempo (s)
3.900	131	1320
4.100	143	1676
4.300	144	2127
4.500	151	2163
4.700	144	2190

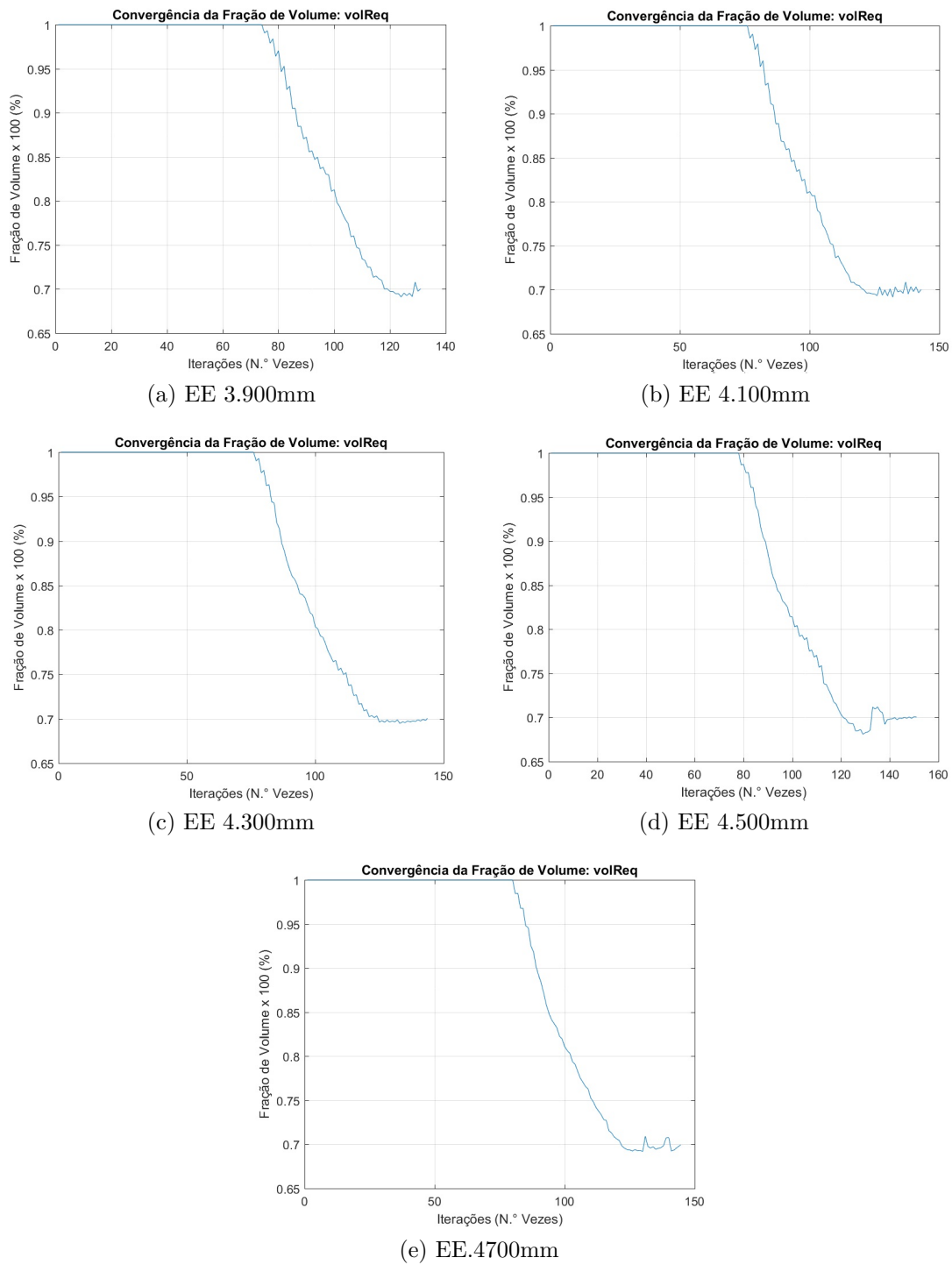


Figura 71 – Convergência da Fração de Volume para o LS implementado. Fonte: Do Autor.

Verificou-se também que a sintetização dos algoritmos nas diversas frações de volume de restrição que podemos ter para o estudo de aplicação com 70%, 75%, 80%, 85% e 90%, conforme figura 72.

Podemos observar da figura 72, que não há a formação dos rasgos na região central da estrutura, diferente do que encontramos nas sintetizações com o algoritmo SIMP implementado. Uma análise possível de ser feita seria a simulação por elementos finitos

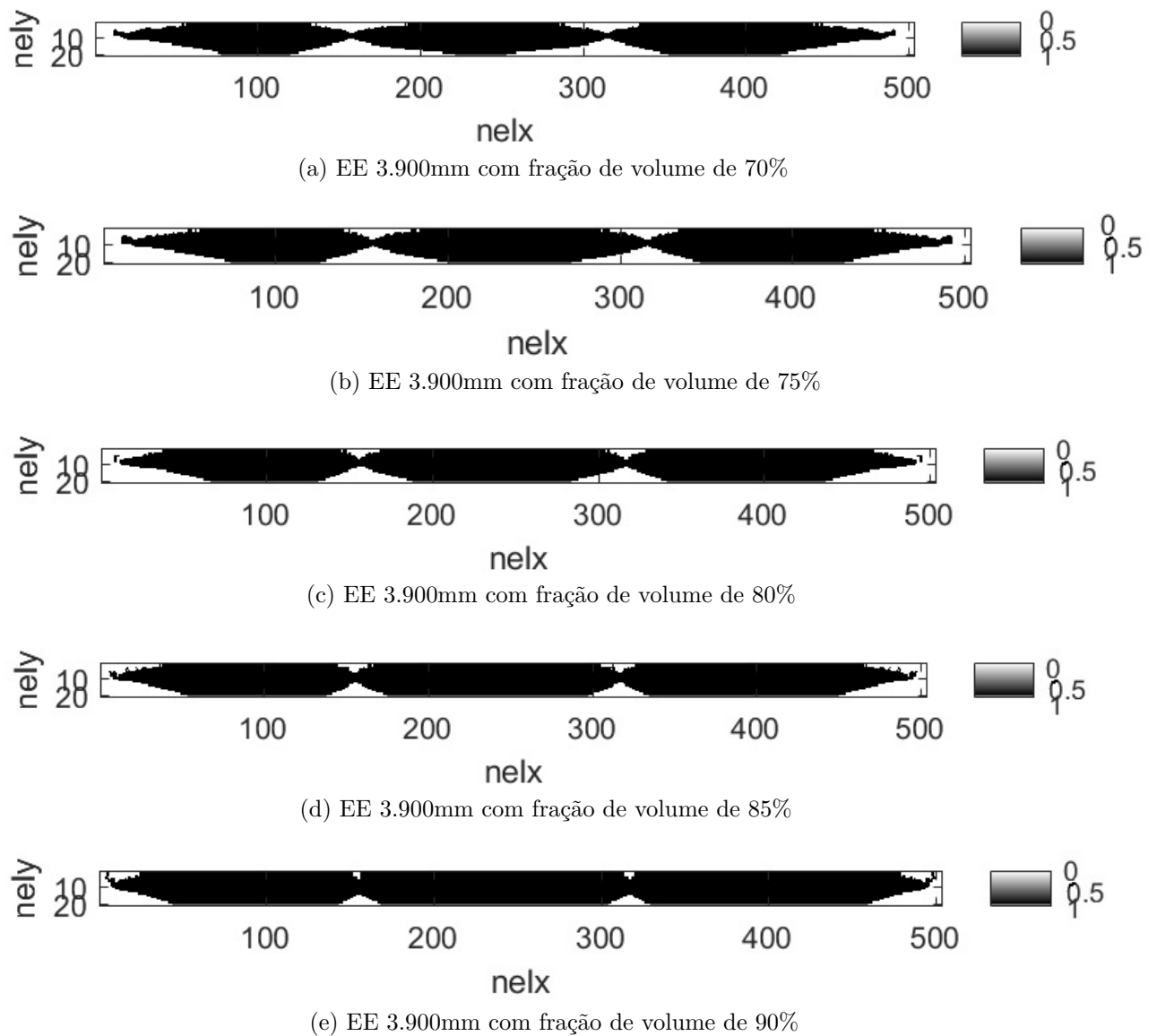


Figura 72 – Sintetização LS implementado com diversas frações de volume. Fonte: Do Autor.

das diversas frações de volume para verificamos em qual fração de volume o critério de segurança para resistência a flexão será atendido. Esta análise será feita nos próximos tópicos.

Portanto, foram analisados os dois algoritmos propostos pelo trabalho, apesar de apresentarem resultados diferentes, mostram que poderão ser utilizados para o processo de demonstração de posicionamento de furos na topologia da longarina do chassi veicular. Além dos algoritmos, conforme já descrito em capítulos anteriores há atualmente programas comerciais utilizados para a metodologia de OT de determinadas estruturas.

### 7.5.5 Coletados dos programas computacionais

#### 7.5.5.1 Programa Computacional Virtual.PYXIS

O Virtual.PYXIS trabalha com os otimizadores descritos na figura 73. Para o trabalho foi utilizado o otimizador CO, pois procurou-se utilizar o mesmo critério abordado pelo algoritmo SIMP implementado. Conforme já discutimos em capítulos anteriores o CO é um método computacional eficiente e tem sido aplicado em vários problemas de otimização topológica de minimização da flexibilidade média da estrutura (compliance).

Abreviação	Método de Otimização	OPT ID
<b>OC</b>	Critério de otimalidade	<b>1</b>
<b>CSA</b>	Aproximação sequencial convexa	<b>2</b>
<b>SLP</b>	Programação linear sequencial	<b>3</b>
<b>SQP</b>	Programação quadrática sequencial	<b>4</b>

Figura 73 – Otimizadores do Virtual.PYXIS. Fonte: Adaptado do manual de [Virtual.PYXIS \(2016\)](#).

O Virtual.PYXIS trabalha em conjunto com diversos Solvers de EF existente no mercado. Na figura 74 verificamos alguns deles. Para o trabalho utilizamos o solver ANSYS.

Tipo de solver externo:	Solver Externo ID:
ABAQUS	1
<b>ANSYS</b>	<b>2</b>
NASTRAN	3
ORION	4
DYNA	5

Figura 74 – Solvers do Virtual.PYXIS. Fonte: Adaptado do manual de [Virtual.PYXIS \(2016\)](#).

O Virtual.PYXIS trabalha com alguns modelos de materiais existentes demonstradas na figura 75. Para o trabalho utilizamos o modelo SIMP.

Modelo de material:	ID:
<b>SIMP</b>	<b>1</b>
Phase Field	2
Heavieside	3

Figura 75 – Modelo do material utilizada no Virtual.PYXIS. Fonte: Adaptado do manual de [Virtual.PYXIS \(2016\)](#)

O Virtual.PYXIS trabalha com alguns filtros conforme figura 76. Para o trabalho utilizamos o filtro com penalização.

Filtros	ID:
Sem projeção / Sem filtragem	0
Projeção linear (Sem penalização)	1
Projeção Não linear (Com penalização contínua)	2
Projeção Heaviside	3

Figura 76 – Filtros utilizados no Virtual.PYXIS. Fonte: Adaptado do manual de Virtual.PYXIS (2016)

Primeiramente foi definido o modo de análise estática no ANSYS, foi carregado o modelo 2D da longarina do chassi, foi introduzido as propriedades do material, e na tela em que se encontra esta figura foi configurado a malha, e depois as condições de contorno conforme figura 55, logo após foi configurado o tipo da otimização conforme figuras 73, 74, 75 e 76. Após a execução da sintetização foi colhido o resultado conforme 78.

Foi utilizado um elemento quadrilátero isoparamétrico, sendo assim, diferentemente dos algoritmos SIMP e LS implementados, não são quadrados com lados unitários, a estrutura se adéqua de forma que a malha já apresenta quadriláteros isoparamétricos que podem ter lados com medidas diferentes.

O Programa Computacional Virtual.PYXIS tem o formato de painel de trabalho conforme figura 77. Demonstra-se simples a configuração dos dados para a sintetização de OT com este programa. Acima do painel esquerdo temos as ferramentas para a configuração pós-processamento da imagem a ser coletada.

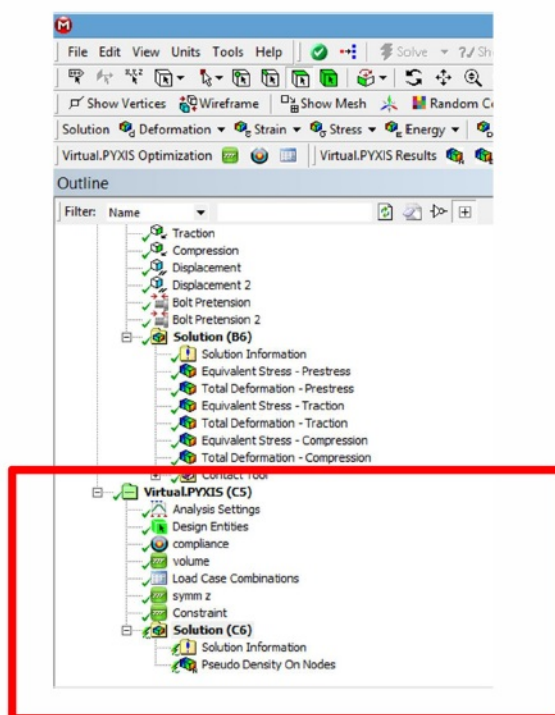


Figura 77 – Painel de Interface de OT no Virtual.PYXIS. Fonte: Adaptado do manual de Virtual.PYXIS (2016).



Por meio dos dados inseridos, e considerando um domínio bidimensional sem os furos da langarina comercial pre-existent, mas contendo os furos principais dos jumelos da suspensão e apoio dos amortecedores, foi discretizado a malha com quadriláteros uniformes isoparamétricos, e obtivemos a sintetização de OT do objeto de estudo com o resultado encontrado na figura 78.

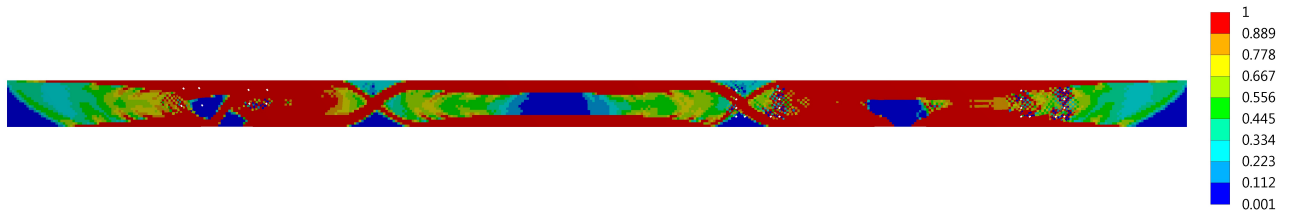


Figura 78 – Sintetização com Virtual.PYXIS (2D) com  $V=70\%$ , EE 3.900mm. Fonte: Do autor

Para uma melhor visualização desta sintetização, selecionamos o valor de corte de 0,7, ou seja, todos os elementos acima estão demarcados com cores sólidas, e os elementos abaixo disso são claros. Assim, na figura 79, ilustramos como podem ocorrer as regiões denominadas de baixa energia ou claras, que poderiam ser eliminadas ou recortadas da langarina do chassi.

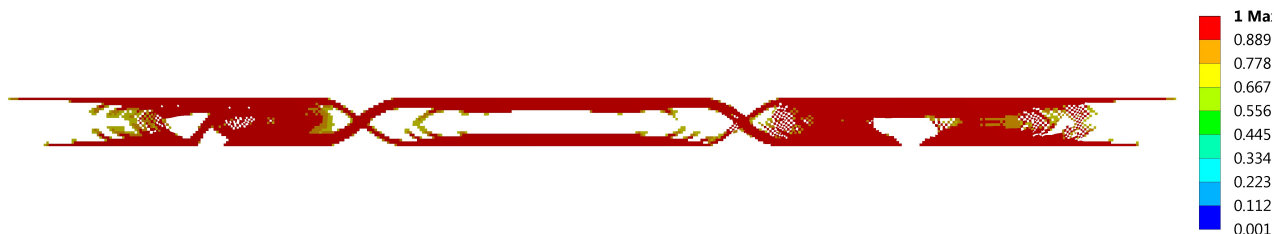


Figura 79 – Sintetização com Virtual.PYXIS (2D) com  $V=70\%$  e corte de pseudo-densidade em 0,7, EE 3.900mm. Fonte: Do autor

No entanto espera-se manter na estrutura pelo menos as pseudo-densidades com valores até 0,3. Desta forma como resultado da sintetização para uma pseudo-densidade até 0,3, e trabalhando no CAD do domínio afim de obtermos uma análise estática com esforços atuantes de tensão e de deformação, obtemos a figura 80.

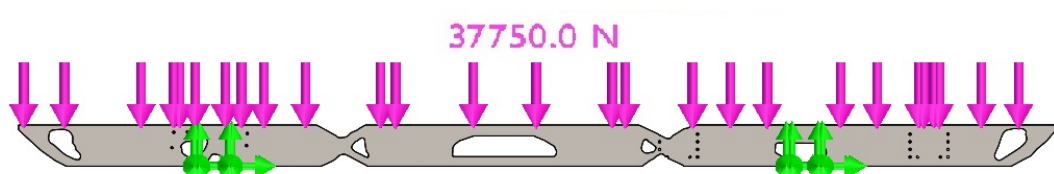


Figura 80 – CAD da Sintetização com Virtual.PYXIS (2D) com  $V=70\%$  e corte de pseudo-densidade em 0,3 com condições de contorno, EE 3.900mm. Fonte: Do autor

Com o CAD gerado na figura 80, encontramos portanto a deformação e o deslocamento de acordo com a figura 81 e 82.

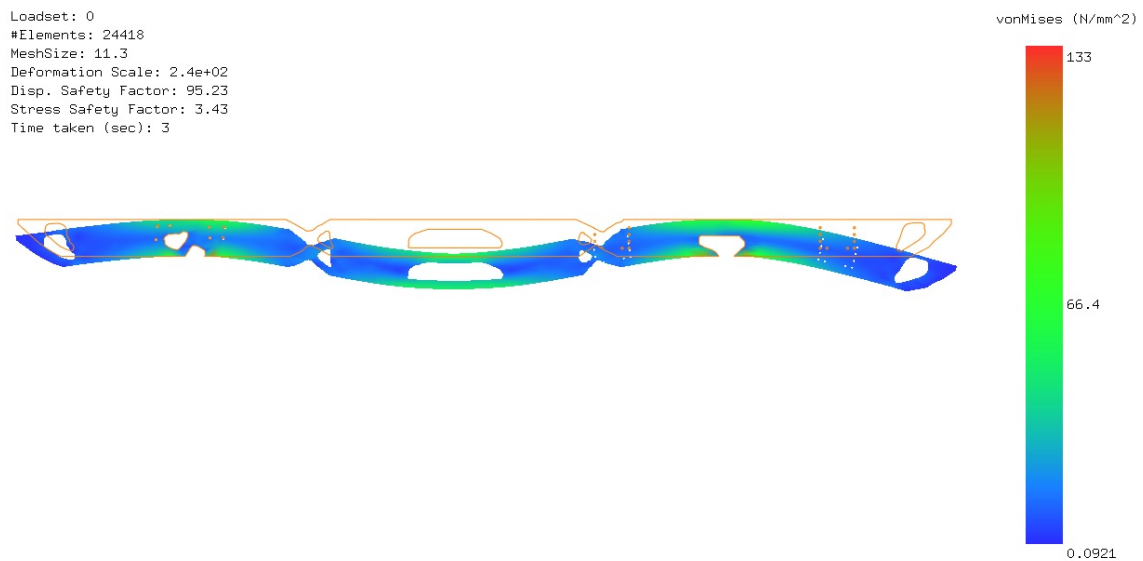


Figura 81 – Simulação de tensão pós-sintetização com ParetoWorks com V=70% com resultados do Virtual.PYXIS, EE 3.900mm. Fonte: Do autor

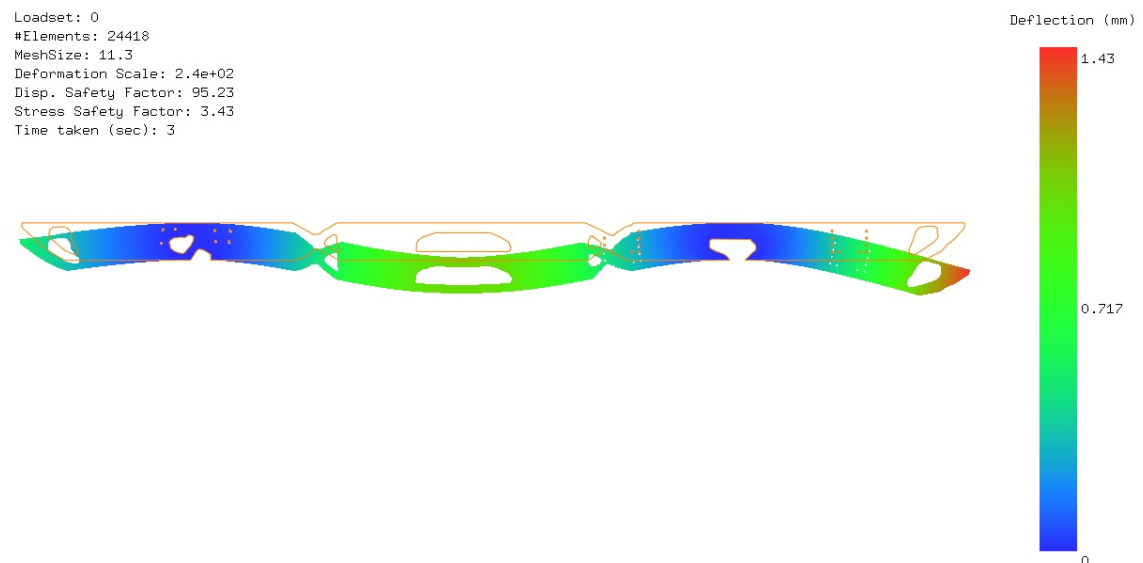


Figura 82 – Simulação de deslocamentos pós-sintetização com ParetoWorks com V=70% com resultados do Virtual.PYXIS, EE 3.900mm. Fonte: Do autor

#### 7.5.5.2 Coleta de dados com Programa Computacional ParetoWorks 2D

Algumas etapas são desenvolvidas para a utilização do programa computacional Pareto.Works para otimizar e simular um domínio qualquer. Para o domínio da longarina do chassi comercial estudada, primeiramente, no pré-processador CAD SolidWorks foi gerado o modelo de placa bidimensional baseado na face planar da longarina do chassi de acordo com a dimensão 2D comercial da figura 55. Obs.: No desenho desta placa foi

eliminado os furos da longarina comercial pois a ideia é verificar como a otimização se comporta em um domínio totalmente sólido, e a partir disto os projetistas podem decidir manter furos ou eliminar, mas sempre levando em consideração que há a necessidade de uma nova sintetização considerando os furos no domínio do projeto. No entanto nesta etapa considera-se apenas os furos de fixação dos jumelos da suspensão e o apoio dos amortecedores, considerados furos principais para a longarina.

Após esta etapa, foi necessário abrir o add-on Paretoworks 2017. Para fazer isto, foi necessário seguir a sequência:

Ferramentas – ParetoWorks – Setup

Após a inicialização do ParetoWorks apareceu uma janela com todas as possibilidades para trabalhar FEA e OT, que é demonstrada de acordo com a figura 83.

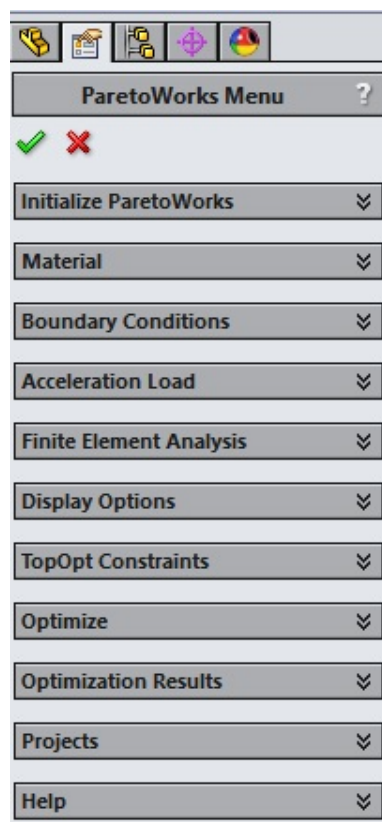


Figura 83 – Quadro lateral de opções do ParetoWorks. Fonte: Do autor

O primeiro passo foi a escolha do sistema de medidas. Para facilitar a visualização da FEA é recomendável utilizar o sistema de medidas mm,N,s, pois com estas unidades os valores de tensão serão visualizados como resposta em MPa e o deslocamento em mm. Sendo assim a caixa da figura 84 demonstra as unidades inseridas. Nesta caixa também é possível selecionar o comando de visualização em formato MS-DOS para visualizarmos instantaneamente as iterações.

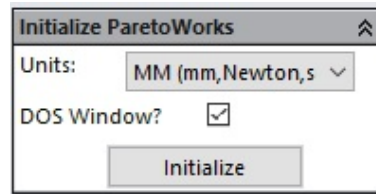


Figura 84 – Caixa de unidades. Fonte: Do autor

O segundo passo foi configurar as definições do material. Assim conforme consideramos no trabalho que o material é parcialmente isotrópico e que estamos usando um aço especial LNE380, temos as seguintes medidas inseridas na figura 85. Como o material LNE380 tem um LE entre 380-530 adotamos a média 455 MPa.

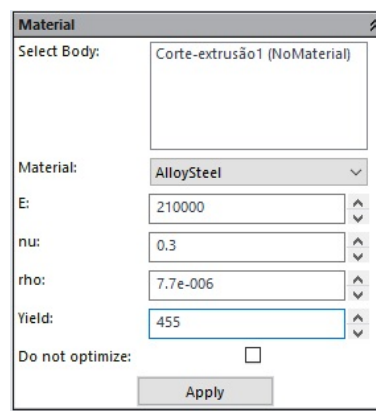


Figura 85 – Propriedades do material para aço LNE380. Fonte: Do autor

O terceiro passo foi configurar as condições de contorno, em que temos os DOFs das fixações em dois pontos fixos e a carga distribuída 3,775ton. sobre a face da longarina, conforme ilustrado na figura 86.

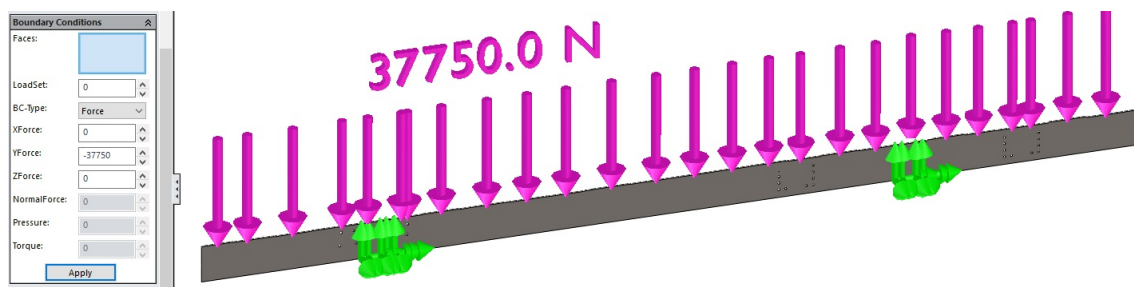


Figura 86 – Definição das condições de contorno do material. Fonte: Do autor

No quarto passo é inserido os dados da malha de FEA, que no caso  $N=10.000$  elementos e tamanho da malha = 14 mm, conforme vemos na figura 87.

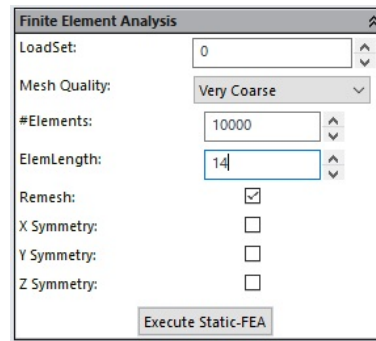


Figura 87 – Condições da malha. Fonte: Do autor

No quinto passo temos que confirmar as condições de restrição, no caso já temos uma restrição que é a fração de volume, no entanto o programa apresenta outra restrição que é o de máximo deslocamento, pois conforme já discutimos o programa ParetoWorks pode trabalhar com funções multi-objetivos. No entanto não utilizaremos esta condição de restrição. Na figura 88 visualizamos estas condições de restrição.

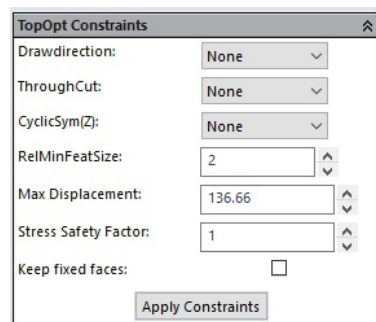


Figura 88 – Condições de restrição. Fonte: Do autor

No sexto passo nós configuramos as condições de otimização, no caso inserimos uma restrição de fração de volume objetivo de 70%, conforme vemos na figura 89.

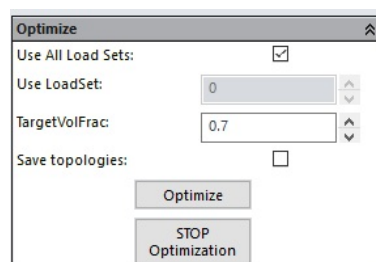


Figura 89 – Condições de otimização, V=70%. Fonte: Do autor

Assim após processamento com o programa computacional ParetoWorks, resultou-se a sintetização conforme as figuras 90 com condição de contorno e a ilustração limpa, figura 91.

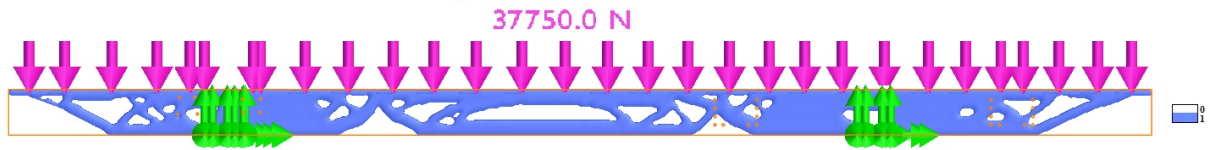


Figura 90 – Síntese com ParetoWorks com  $V=70\%$  e com condições de contorno, EE 3.900mm. Fonte: Do autor



Figura 91 – Síntese com ParetoWorks com  $V=70\%$ , EE 3.900mm. Fonte: Do autor

Diferente dos resultados demonstrados pela síntese do Virtual.PYXIS que gera valores intermediários em que tivemos que definir um corte das pseudo-densidades em 0,3 para representar o CAD e gerar as simulações pós-síntese, a síntese no Virtual.PYXIS por trabalhar com a estratégia de LS, não produz valores de pseudo-densidades intermediários, neste caso não será necessário adotarmos um corte.

No programa ParetoWorks com a síntese 2D da figura 90, encontramos o resultado da simulação por elementos finitos pós-síntese. Na figura 92 observamos o resultado das tensões e também o valor do coeficiente de segurança encontrado pelo próprio programa.

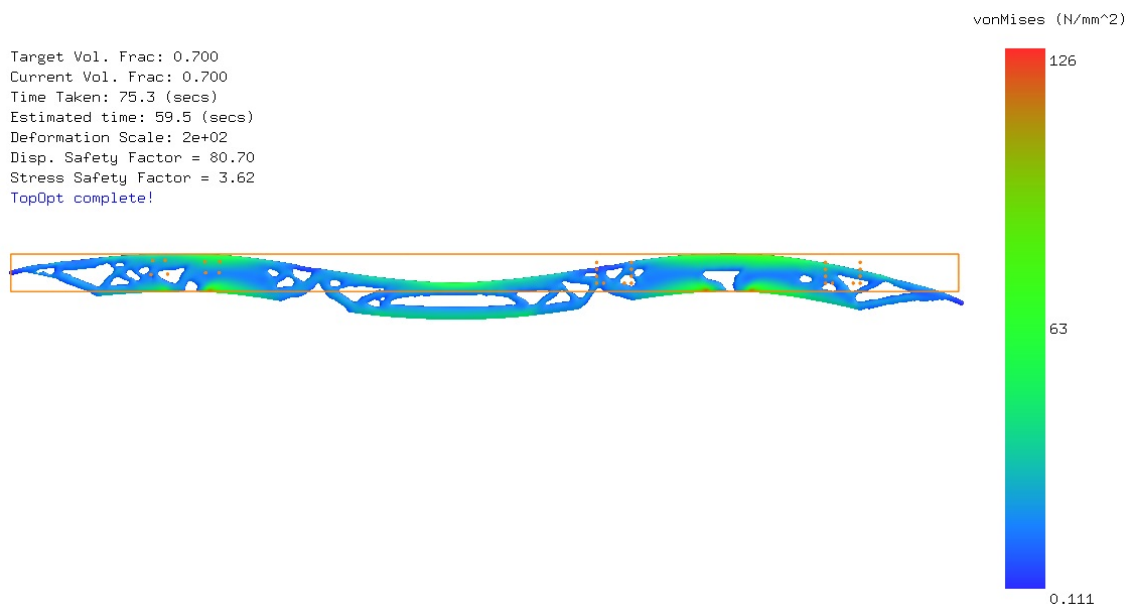


Figura 92 – Simulação de tensão pós-síntese com ParetoWorks com  $V=70\%$ , EE 3.900mm. Fonte: Do autor

E também tem-se o resultado dos deslocamentos pós-síntese da figura 90, ilustrados na figura 93.

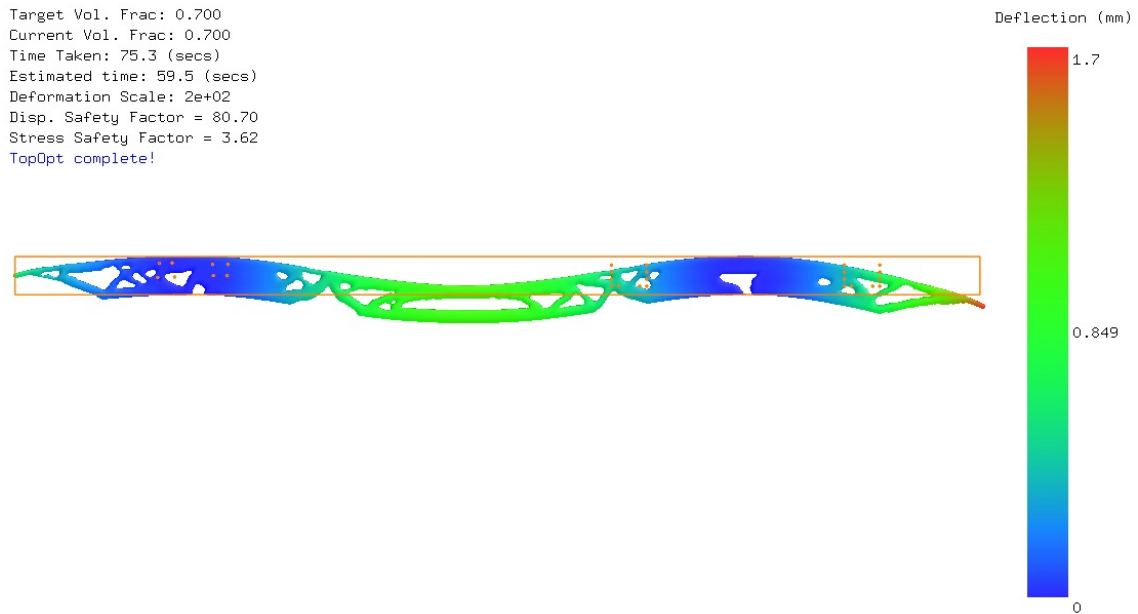


Figura 93 – Simulação de deslocamentos pós-sintetização com ParetoWorks com  $V=70\%$ , EE 3.900mm. Fonte: Do autor

Com os resultados encontrados que tem um coeficiente de segurança de 3,62 observamos que a estrutura está bem dimensionada considerando as cargas estáticas. No entanto procura-se testar a longarina bidimensional para verificar até qual valor de restrição de volume teríamos a máxima rigidez da estrutura sem o critério da falha ou de descolamentos dos elementos ao longo da estrutura. Na figura 94, verificamos a porcentagem de sintetização encontrada para este coeficiente que foi de 38,3% com um coeficiente de segurança de 2,23.

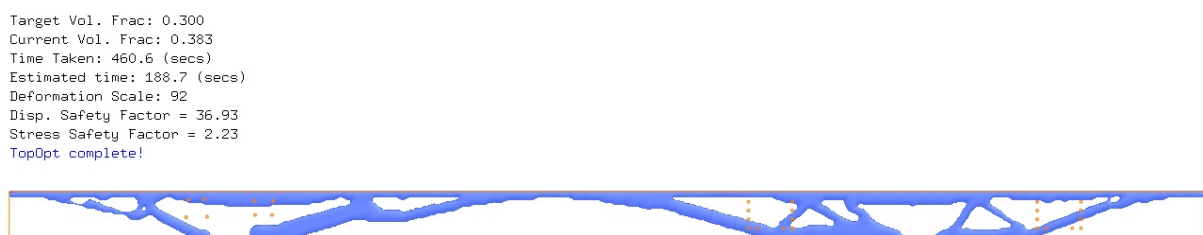


Figura 94 – Máxima rigidez da estrutura com sintetização 2D com ParetoWorks, EE 3.900mm. Fonte: Do autor

Com este resultado observamos a capacidade deste longarina de material LNE380 de ser otimizada.

#### 7.5.5.3 Coleta de Dados com Programa Computacional ParetoWorks 3D

Além da otimização de um plano bidimensional da longarina do chassi, no ParetoWorks sintetizamos diretamente com o perfil 3D e com a carga de 10ton. Sendo assim para este procedimento adotou-se os mesmos valores das propriedades e das condições



de contorno, considerando a carga máxima real aplicada nesta longarina. Observa-se que foram incluídos os furos principais da longarina, que são os furos de fixação dos jumelos (suportes da suspensão traseira), e fixação dos amortecedores no eixo dianteiro.

A sintetização encontrada para esta análise com a restrição de volume em 70% está ilustrada de acordo com as figuras 95, 96 e 97.

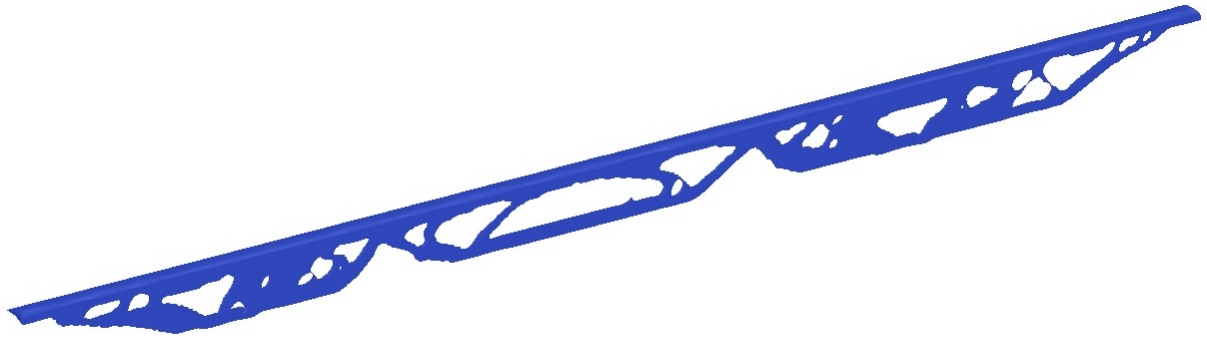


Figura 95 – Sintetização em perspectiva em 3D, com  $V=70\%$ . Fonte: Do autor

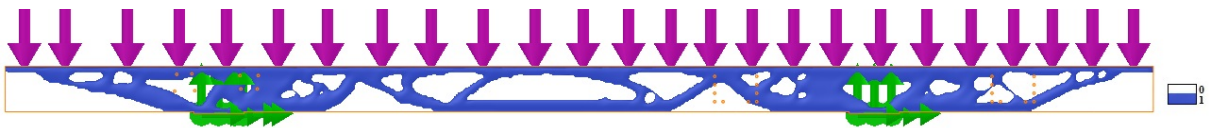


Figura 96 – Sintetização vista frontal em 3D com condições de contorno, com  $V=70\%$ . Fonte: Do autor



Figura 97 – Sintetização vista frontal em 3D, com  $V=70\%$ . Fonte: Do autor

Com a visualização das figuras 90 da sintetização bidimensional e a figura 96 tridimensional, é possível observar que houve uma equivalência na disposição dos buracos no modelo 3D em relação a 2D, provando que a estratégia adotada para utilizar os dois domínios balanceados pela deformação da linha elástica estão corretas.

No programa ParetoWorks com a sintetização 3D da figura 96, também encontramos o resultado da simulação por elementos finitos pós-sintetização. Na figura 98 observamos o resultado das tensões e também o valor do coeficiente de segurança encontrado pelo próprio programa.



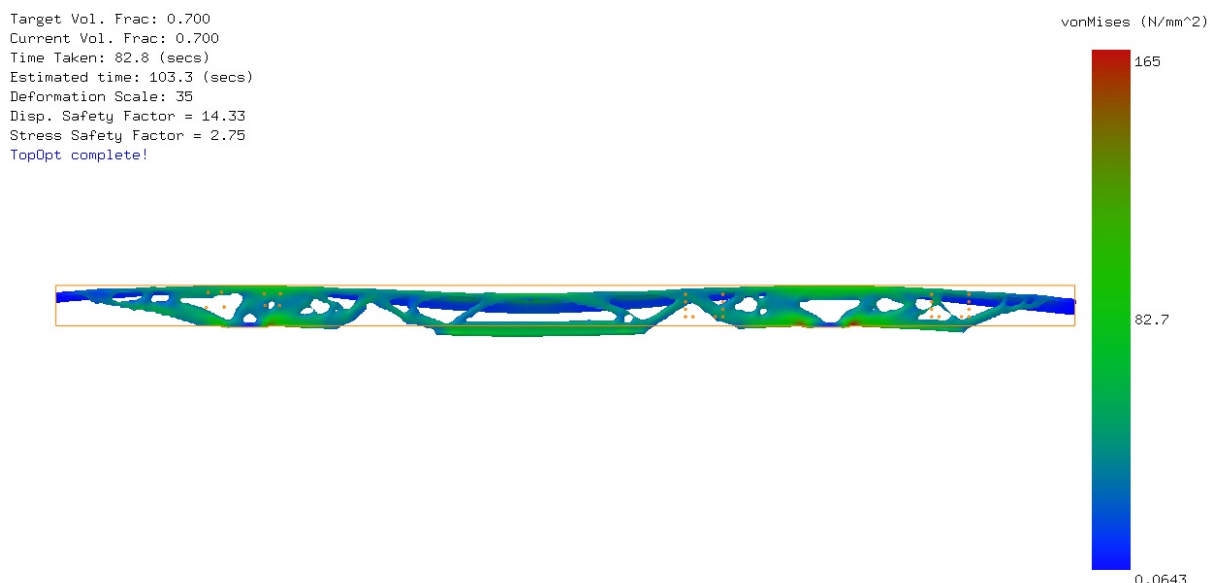


Figura 98 – Simulação de tensão pós-sintetização em 3D com ParetoWorks com V=70%, EE 3.900mm. Fonte: Do autor

E também tem-se o resultado dos deslocamentos pós-sintetização da figura 96, ilustrados na figura 99.

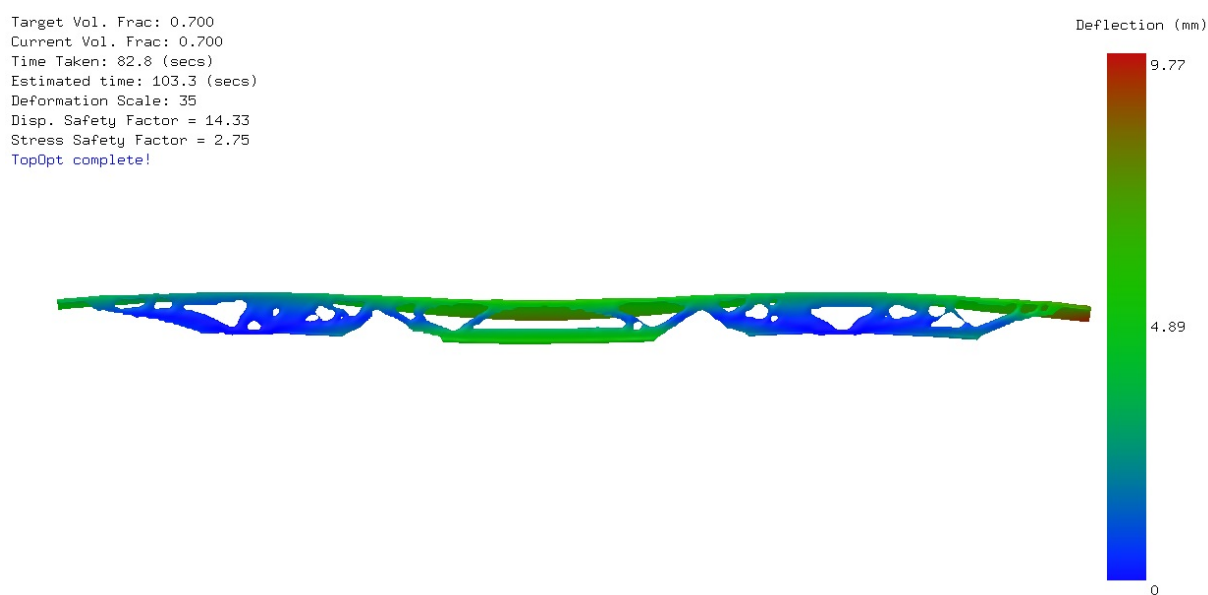


Figura 99 – Simulação de deslocamentos pós-sintetização em 3D com ParetoWorks com V=70%, EE 3.900mm. Fonte: Do autor

Com os resultados encontrados que tem um coeficiente de segurança de 2,75 observamos que a estrutura está bem dimensionada considerando as cargas estáticas. No entanto procura-se testar assim como foi feito para a longarina 2D, a longarina tridimensional para verificar até qual valor de restrição de volume teríamos a máxima rigidez da estrutura sem o critério da falha ou de descolamentos dos elementos ao longo

da estrutura. Na figura 100, verificamos a porcentagem de sintetização encontrada para este coeficiente que foi de 50,7% com um coeficiente de segurança de 1,03.

```
Target Vol. Frac: 0.400  
Current Vol. Frac: 0.507  
Time Taken: 214.8 (secs)  
Estimated time: 206.2 (secs)  
Deformation Scale: 11  
Disp. Safety Factor = 4.35  
Stress Safety Factor = 1.03  
TopOpt complete!
```



Figura 100 – Máxima rigidez da estrutura com sintetização 3D com ParetoWorks, EE 3.900mm. Fonte: Do autor

Podemos ter uma avaliação de qual é o comportamento da longarina do chassi comercial com o processo de otimização topológica aplicado. portanto adicionalmente ao estudo de OT e para critério de decisão do projetista, não somente a sintetização de OT deverá ser considerada para uma análise com argumentos, a análise FEA pós-sintetização é extremamente importante para avaliação. Não foi possível colher os dados de deslocamento e máxima tensão nos algoritmos, no entanto os dois programas computacionais analisados nos dão a opção de analisarmos estes dados.

Assim o ensaio foi feito considerando apenas a análise estática, a análise dinâmica ou modal foi desconsiderada, pois foi de interesse da pesquisa apenas a síntese do MOT na redução de peso e furos da longarina estaticamente para este trabalho. Sugere-se portanto uma análise dinâmica em nova síntese. É recomendado portanto, que após qualquer sintetização de OT, avalie a condição do domínio com uma análise de elementos finitos.

A análise de elementos finitos da longarina do chassi em comparação com os dois programas computacionais deveriam apresentar resultados aproximados, pois a teoria de MEF que é adicionado nestes programas é equivalente, assim as condições de contorno equivalentes (domínio, carga e fixações) geram confiabilidade nos resultados. Elas apenas divergem em função dos resultados encontrados na sintetização de OT, pois como a simulação é feita posteriormente a estes dados, ela depende muito da condição de saída da sintetização.

Neste trabalho, conseguimos verificar os limites de tensão máxima atuante no local em que a estrutura poderia falhar. Com a utilização dos programa computacional ParetoWorks, que é um módulo independente da análise de sintetização e demonstra rapidamente os resultados ilustrando a máxima tensão a flexão e ao deslocamento.

Na tabela 18 é possível verificar os dados de tensão atuante, deslocamento máximo e fração máxima de otimização para que o domínio não falhe, segundo o critério de análise estática, para os casos analisados com os programas computacionais.

Tabela 18 – Resultados numéricos da análise FEA.

Software	Nº.Elem.	Tensão at. (MPa)	Desl. at. (mm)	coef. seg. (adm.)
Virtual.PYXIS 2D	10.000	133,00	1,43	3,43
ParetoWorks 2D	10.000	126,00	1,7	3,62
ParetoWorks 3D	10.000	165,00	9,77	2,75

Observou-se que a tensão atuante é maior no programa computacional ParetoWorks 3D do que nos outros casos, isto ocorre pois com a aba incorporada ao perfil 3D, temos um momento torsor no sentido da região interna da longarina diferentemente dos casos 2D que a força é aplicada verticalmente acima do plano. No entanto, observa-se que as simulações encontradas com as sintetizações 2D do Virtual.PYXIS e do ParetoWorks são bem aproximadas tanto em relação a tensão atuante, quanto ao deslocamento, significando que a sintetização parece trabalhar de forma coerente. Já se espera um deslocamento atuante diferente no caso 3D, por questão do momento torsor já discutido acima.

No ParetoWorks o MOT é por LS, que é um dos tipo de otimização estrutural como vimos nos capítulos anteriores, este procura trabalhar com a sensibilidade topológica e a função de LS para ver os limites da estrutura. Observou-se que foi utilizada uma quantidade de aproximadamente 10.000 elementos, para caracterizar uma amostra similar dos dados inseridos pelo projetista. Pela fato do ANSYS que é o solver que trabalha com o Virtual.PYXIS apresentar-se como versão estudantil, houve um limite no número máximo de elementos da malha discretizados e também no número de nós. Talvez por este fato podemos ter tido o desvio nos deslocamentos entre as duas análises, pois se diminuirmos o tamanho dos elementos teremos uma maior acuracidade da análise.

Foi interessante observar que em relação aos domínios 2D e 3D, houve pequenas alterações nos valores de tensão atuante à flexão. O aumento da tensão atuante e do deslocamento no caso do domínio tridimensional é factível quando observamos que temos a inclusão de abas no perfil, isto demonstra que as abas devem ser consideradas para o modelo final de análise de OT.

Para avaliar até qual seria a restrição de volume para a otimização aplicada na longarina segundo um critério de segurança considerado ideal pelo projetista, foi extrapolado a restrição de volume para o domínio tridimensional encontrado na figura 97, considerando encontrar um coeficiente de segurança de valor 2. Este será o coeficiente utilizado para definir a condição final da longarina otimizada com base no critério de análise estática e resistência a flexão para o material LNE380. Assim na figura 101 ilustra-se a sintetização para esta condição.



Figura 101 – Síntese de vista frontal em 3D, com coeficiente de segurança  $s=2$ . Fonte: Do autor

Portanto encontra-se uma restrição máxima de volume para este material de 60%. A tensão e os deslocamentos pós-síntese são ilustradas nas figuras 102 e 103 respectivamente.

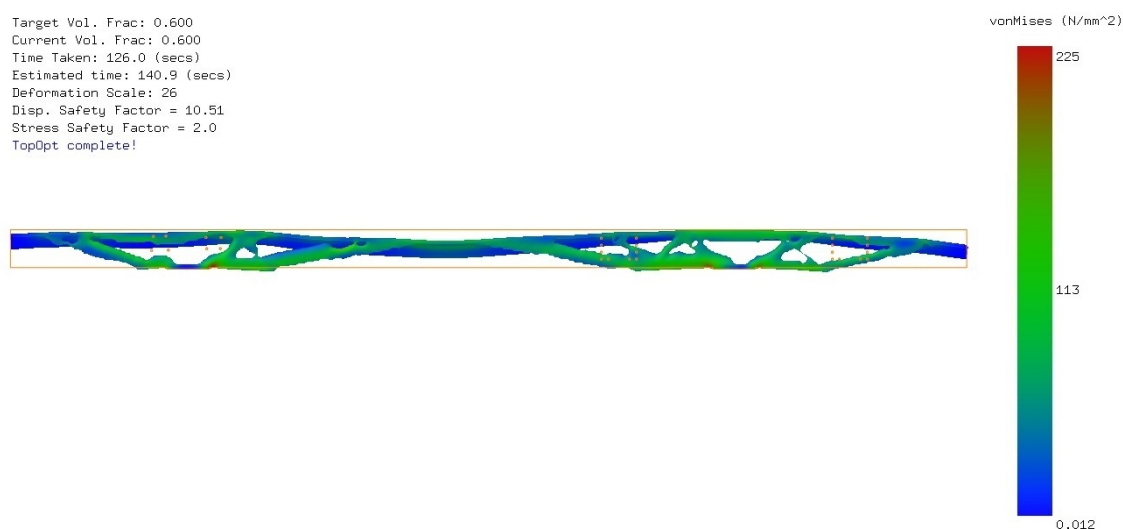


Figura 102 – Simulação de tensão pós-síntese em 3D com ParetoWorks com  $V=60\%$ , EE 3.900mm. Fonte: Do autor

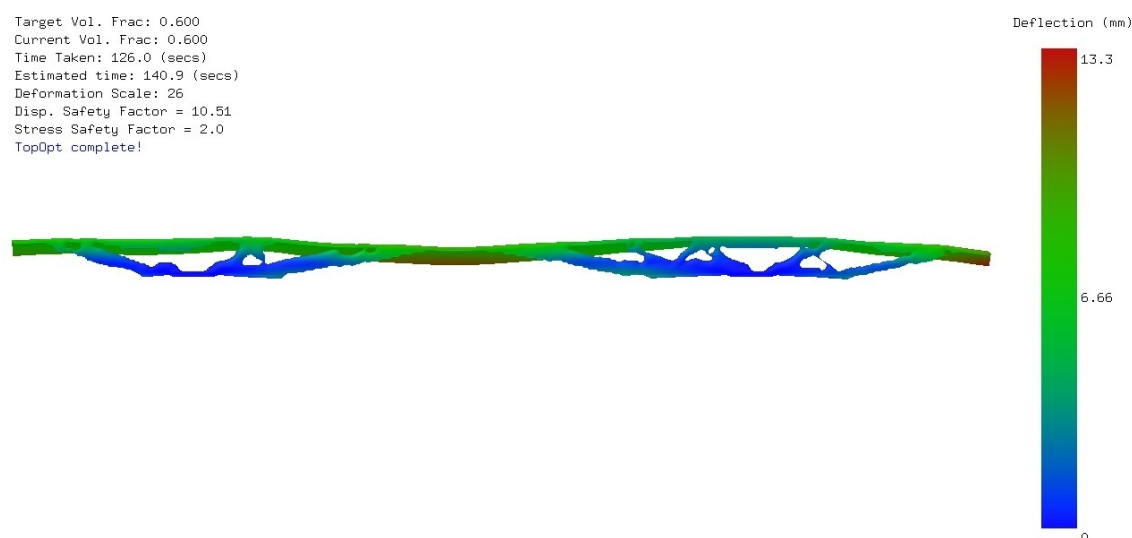


Figura 103 – Simulação de deslocamentos pós-síntese em 3D com ParetoWorks com  $V=60\%$ , EE 3.900mm. Fonte: Do autor

## 7.6 Análise Comparativa dos resultados encontrados

O projeto tem como premissa o comparativo por meio do MOT para reduzir peso de uma longarina de um chassi de caminhão minimizando a flexibilidade média ou compliance (com restrição de volume), utilizando programas computacionais ParetoWorks e Virtual.PYXIS e algoritmos de MOT implementados com o uso de alguns destes existentes na literatura. Neste trabalho foram combinados a sintetização de otimização topológica que utiliza o método de análise por elementos finitos, observando a melhor eficiência para a função de minimizar flexibilidade média por meio de ciclo de iterações. Com a síntese do MOT e análise FEA, é possível reduzir custos para fabricar em escala produtiva uma nova topologia da longarina do chassi, pois tem-se a redução do peso da estrutura. Esta pesquisa tem sido altamente incentivada pela indústria automotiva, por meio de empresas como a Scania do Brasil (fabricante do chassi), Lavrita Engenharia (fabricante das diversas carrocerias) e Virtual CAE (serviços de otimização), que forneceram recursos materiais com informações técnicas etc. Foram considerados a análise 2D e 3D desta longarina, pois além do meio de se ter um real uso para o método nesta aplicação, também foi investigado algoritmos para que se possa ter uma visualização do estado da arte de OT e também, utilizar os algoritmos implementados em outros componentes que não seja a longarina. Como resultado da análise comparativa entre os programas computacionais e os algoritmos é possível chegarmos em uma conclusão de como a sintetização atua em um processo de otimização de uma longarina de um chassi veicular.

### 7.6.1 Resultados da análise MOT

Adotou-se por condição de critério do projetista, a condição de minimizar o compliance (flexibilidade média da estrutura) com restrição de volume em 70% da estrutura, sendo que a flexibilidade média poderia chegar em uma restrição de volume menor, que no caso foi de 60% por meio de testes feitos para esta aplicação. Qualquer uma destas porcentagens tem-se um certo coeficiente de segurança para a proposta do novo plano de furação garantindo que não haverá falha do material para a condição estática dom resistência a flexão e ao deslocamento. Portanto temos 6 resultados encontrados, 2 algoritmos 2D, 2 sintetizações de programas computacionais em 2D e 2 sintetizações de programa computacional em 3D, conforme figura 104. Foi adotado aqui apenas o EE de 3.900mm, que é chassi mais comum disponibilizado comercialmente pela montadora de veículos.

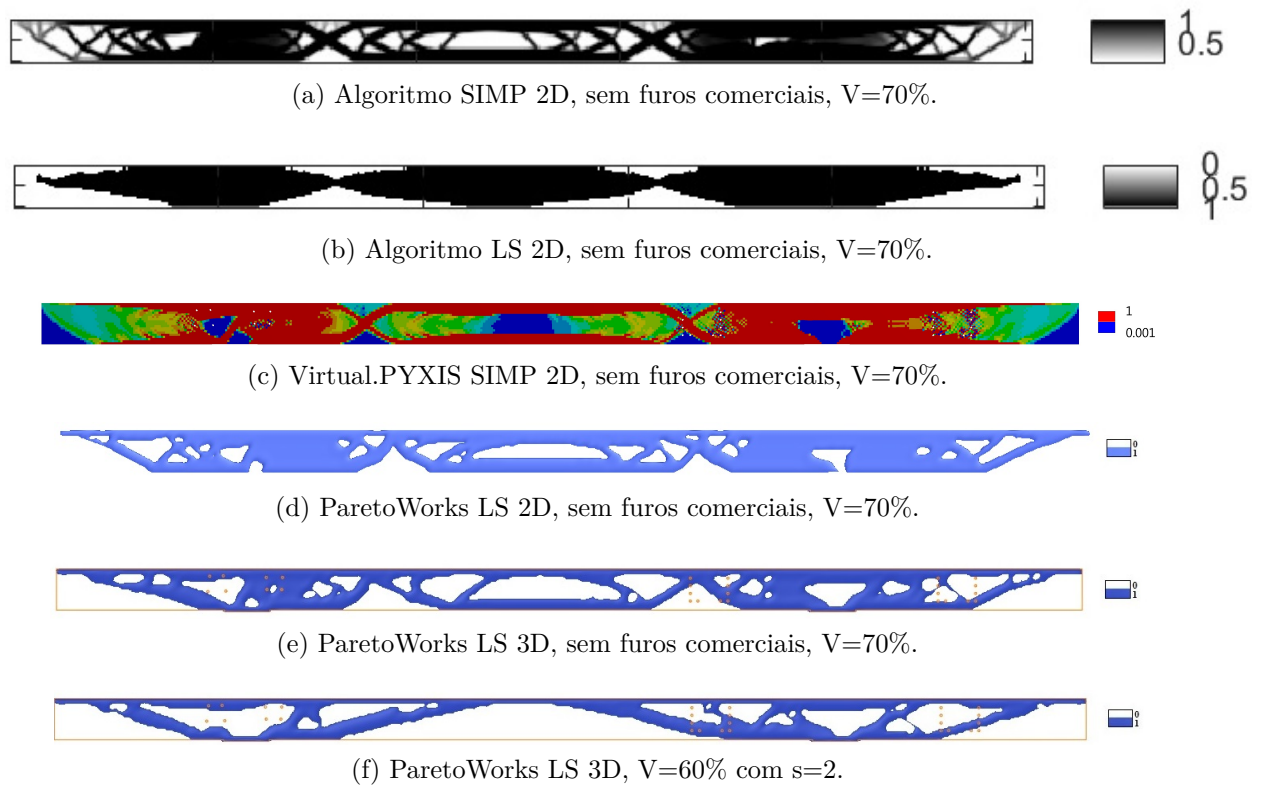


Figura 104 – Sintetizações de MOT encontradas. Fonte: Do Autor.

Observa-se que as 6 análises apresentam formas de topologia diferentes, sejam elas pela característica 2D ou 3D, ou pelo tipo de abordagem SIMP ou LS, ou até pelos filtros utilizados em cada uma das abordagens. Porém verifica-se que as regiões tidas como claras, que tem as densidades com o valor 0 são passíveis de receber furos em uma topologia. No entanto conforme discutimos anteriormente todos os algoritmos computacionais, figuras 104a, b, e também os programas computacionais com o domínio 2D Virtual.PYXIS, figura 104c, e ParetoWorks figura 104d, por terem formação de enrijecimento do tipo hinge parecem não apresentar resultados confiáveis para esta aplicação. O objeto 3D no caso analisado com o ParetoWorks obteve resultados sem formação do hinge e que parecem portanto ser confiável, que é o objeto que será tratado posteriormente para a proposta de uma nova topologia da longarina.

## 7.7 Proposta de uma nova topologia da longarina

### 7.7.1 Análise de longarina comercial (todos os furos existentes)

O presente trabalho propôs identificar uma nova topologia para a longarina de um chassi comercial. Na figura 104 foi apresentado o local em que os rasgos de uma longarina de chassi poderão estar localizados para que nestas regiões claras (0) poderíamos inserir os furos. Observa-se que nas sintetizações com os programas computacionais da

figura 104, os furos denominados "principais", que são aqueles que suportam os jumelos e amortecedor da suspensão foram mantidos. Porém mais análises são necessárias para avaliarmos as condições longarina comercial conforme está sendo fabricada atualmente com todos os furos existentes. Na figura 105 ilustramos a análise FEA para a condição inicial da longarina comercial contemplando os furos já definidos pelos projetistas, já na 106 temos a sintetização desta longarina com o domínio 3D e restrição de volume em 70% para observarmos como os furos existentes influenciam na sintetização com esta restrição. Na figura 107 observa-se o critério de máxima rigidez, para avaliarmos qual restrição de volume e em que coeficiente de segurança de valor a estrutura pode chegar com a condição de manutenção de todos os furos.

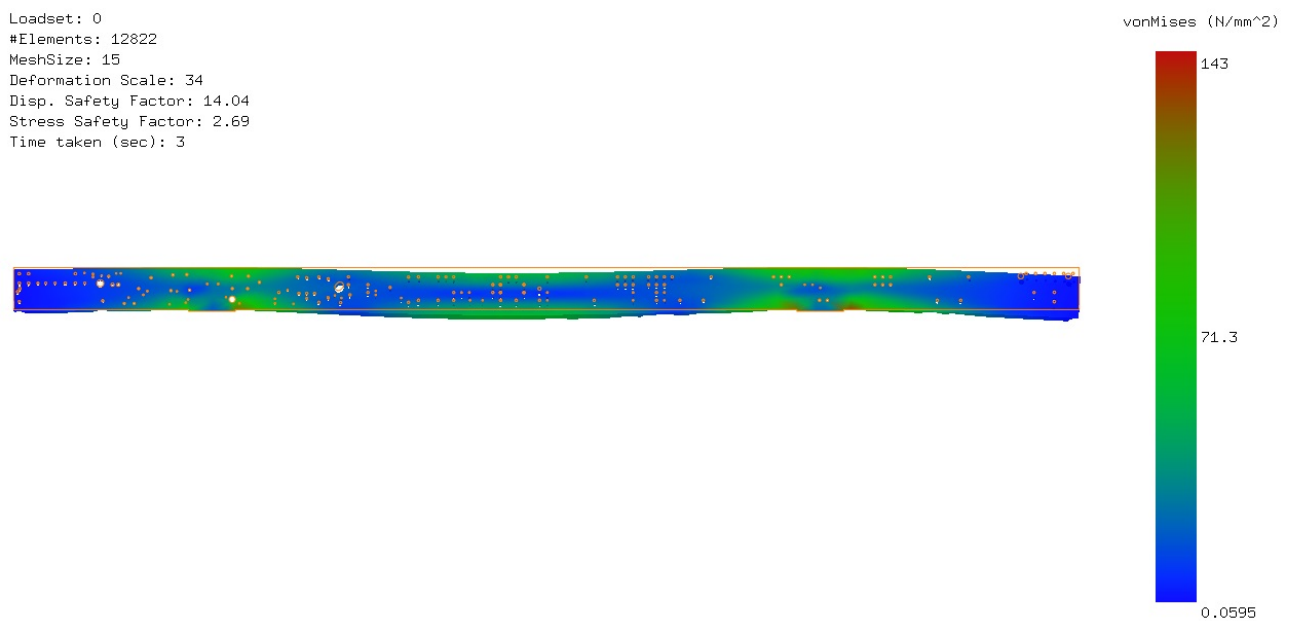


Figura 105 – FEA da Longarina comercial existente contendo todos os furos. Fonte: Do autor



Figura 106 – Sintetização da longarina comercial em 3D (ParetoWorks) com restrição de V=70%. Fonte: Do autor



Target Vol. Frac: 0.500      vonMises máx. = 169 MPa  
 Current Vol. Frac: 0.672      Deslocamento máx = 9,97mm  
 Time Taken: 66.2 (secs)  
 Estimated time: 150.7 (secs)  
 Deformation Scale: 34  
 Disp. Safety Factor = 14.04  
 Stress Safety Factor = 2.69  
 TopOpt complete!



Figura 107 – Sintetização da longarina comercial em 3D (ParetoWorks) com maximização de rigidez. Fonte: Do autor

Na tabela 19 demonstrou-se portanto os dados encontrados com as simulações e sintetizações das figuras 105, 106 e 107.

Tabela 19 – Resultados de FEA e sintetização da longarina comercial que contém todos os furos

Análise	Tensão at. (MPa)	coef. seg. (adm.)	Desl. at. (mm)	Restrição Vol. (%)
FEA	143,00	3,18	6,17	100
OT	168,52	2,70	9,77	70
OT	169,14	2,69	9,97	67,2

### 7.7.2 Estratégia para otimização final da longarina

Como avaliado a longarina comercial que contém todos os furos apresentou um critério de máxima rigidez com o valor de restrição de  $V=67,2\%$ , em comparação com a longarina avaliada com somente os furos necessários para fixação dos jumelos da suspensão e de apoio dos amortecedores esta restrição foi de  $V=60,0\%$ . Portanto, é fácil concluir que quanto mais eliminamos furos da longarina existente maior será a restrição de volume que poderemos utilizar na sintetização.

Com base nas duas sintetizações obtidas com restrição de volume de  $V=67,2\%$  e de  $V=60,0\%$ , utiliza-se como estratégia a com menor restrição por garantir que todos os furos possam estar locados na topologia da estrutura.

Sendo assim a estrutura com a inclusão dos furos apresentará a ilustração da figura 108.



Figura 108 – Otimização escolhida com projeção dos furos. (ParetoWorks 3D) Fonte: Do autor



### 7.7.3 Nova longarina do Chassi

Por meio da figura 108, que foi a escolha pelo critério de ser sintetizado com a composição de todos os furos na estrutura, propõe-se readequar a mesma para poder deslocar os furos das regiões claras (vazias) para as regiões sólidas, e quando não for possível este deslocamento re-inserir elementos sólidos no domínio nas regiões que são claras. Desta forma teremos a nova longarina do chassi com novo reposicionamento de furos e otimizada. Num primeiro momento foi utilizado a longarina comercial existente e foi colocado em paralelo a nova longarina sintetizada. E da longarina sintetizada foi transposto os recortes do perfil da topologia deixando limites ao critério do projetista observando as furações. Na posição em que se encontravam-se os jumelos da suspensão e o apoio dos amortecedores foram mantidos os domínios eliminando os recortes. Assim, na figura 109 observamos uma na cor "vermelha" a longarina sintetizada e na cor "cinza" a nova longarina do chassi desejada.

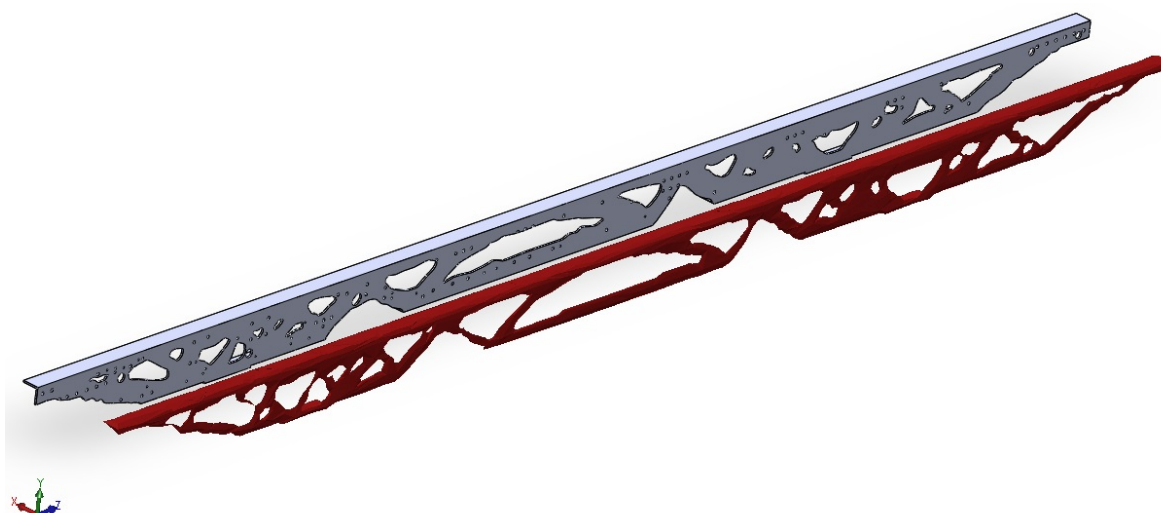


Figura 109 – Utilização das máscara do perfil sintetizado pra produzir nova longarina  
Fonte: Do autor

Neste primeiro momento, observou-se 174 furos existentes na longarina comercial original. Com a projeção destes recortes 123 furos ficarão no domínio sólido e 51 furos foram reposicionados. Assim na figura 110 é ilustrado os 51 furos que ficarão fora do domínio de elementos sólidos.

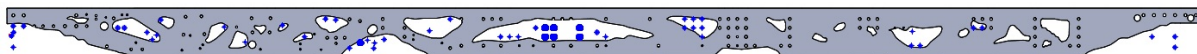


Figura 110 – Furos existentes fora dos elementos sólidos. Fonte: Do autor

No próximo passo, cabe a critério do projetista mover estes furos existentes que estão posicionados na região clara para dentro da região que contém os elementos sólidos. Portanto, na figura 111, temos a novo reposicionamento dos furos pelo critério do projetista.

Cabe observar que o projetista deverá verificar os acessórios constantes no equipamento como suporte de baterias, das travessas do quadro do chassi etc. para avaliar o impacto da alteração do posicionamento destes furos.

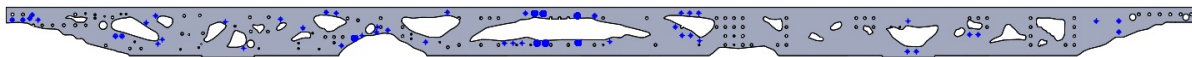


Figura 111 – Furos alocados para dentro dos elementos sólidos. Fonte: Do autor

Com a disposição dos furos alocados dentro dos elementos sólidos do domínio d estrutura, temos todos os furos do chassi comercial original alocados dentro desta nova longarina sintetizada. Portanto, nas figuras 112 e 113, ilustramos o projeto da longarina final nas vistas frontal e perspectiva respectivamente.

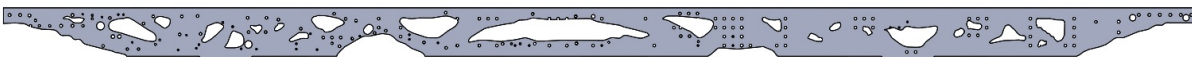


Figura 112 – Longarina do chassi final, vista frontal. Fonte: Do autor

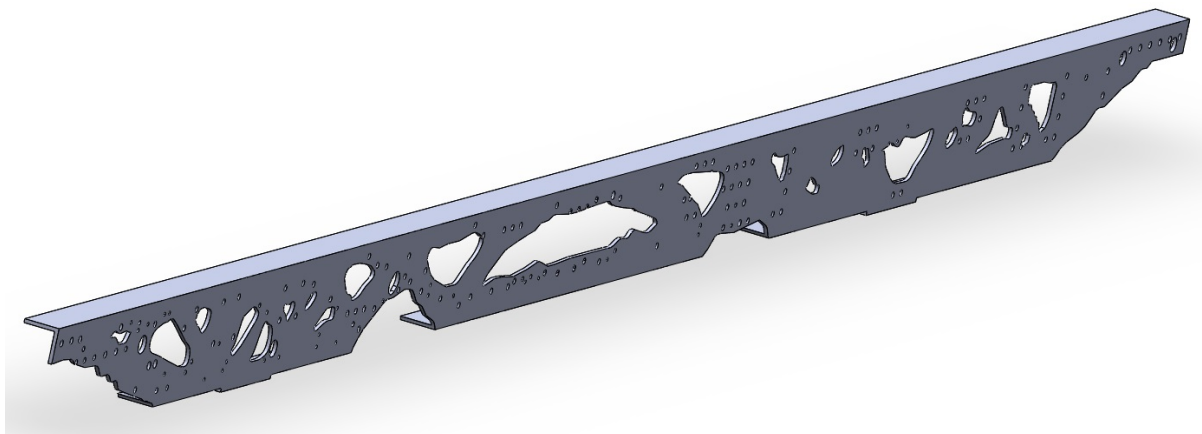


Figura 113 – Longarina do chassi final, vista em perspectiva. Fonte: Do autor

Para avaliarmos as condições de volume final da longarina, bem como avaliar as condições estáticas de tensão e de deslocamento, uma nova análise por elementos finitos foi feita para o domínio do projeto pós-sintetizado. Na figura 114, ilustramos a tensão máxima atuante para este perfil, e na figura 115 ilustramos o deslocamento máximo para o perfil.

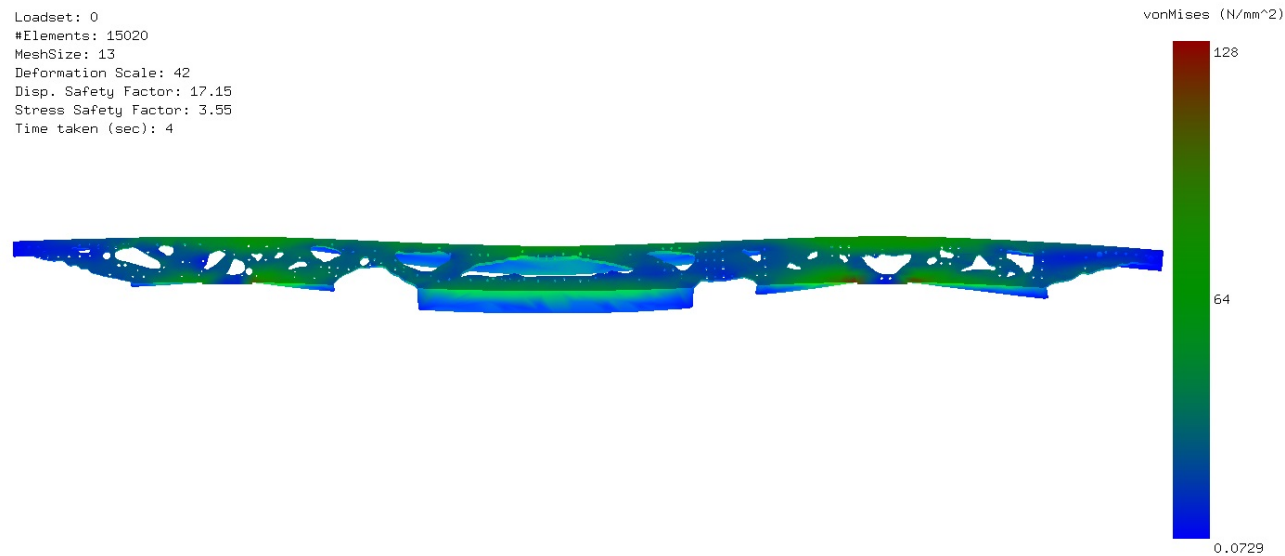


Figura 114 – Análise FEA da longarina final. Tensão máxima. Fonte: Do autor

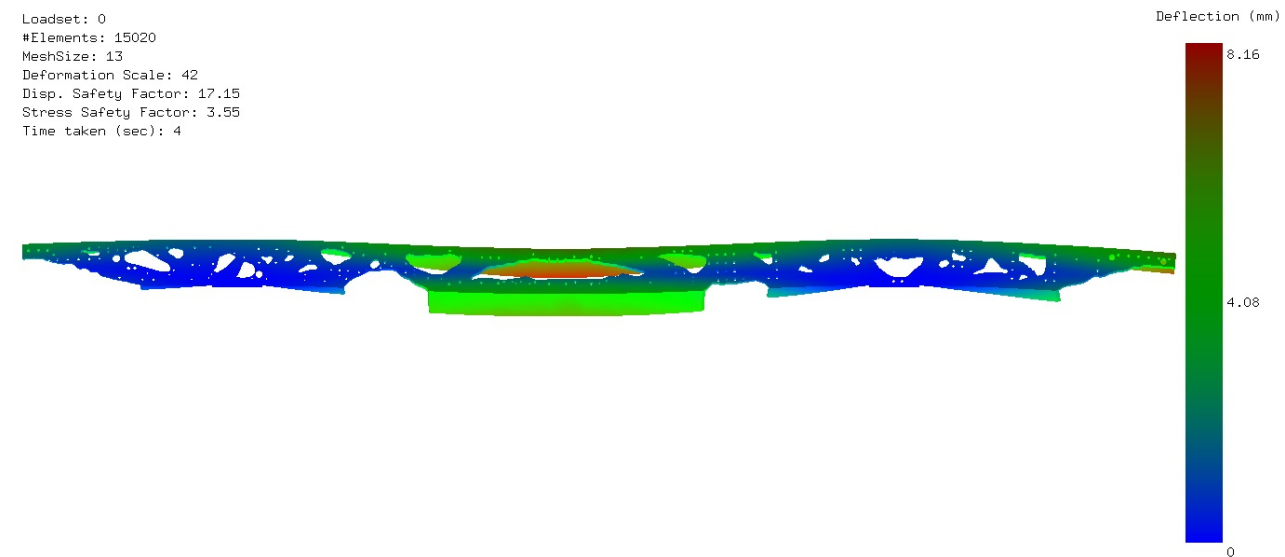


Figura 115 – Análise FEA da longarina final. Deslocamento máximo. Fonte: Do autor

Na tabela 20 demonstra-se portanto os dados encontrados com as simulações e sintetizações das figuras 105 e 112.

Tabela 20 – Resultados de FEA e sintetização da longarina comercial que contém todos os furos

Análise	Tensão at. (MPa)	coef. seg. (adm.)	Desl. at. (mm)	Massa (kg)
Chassi Comercial	143,00	3,18	6,17	282,12
Chassi Final	128,00	3,55	8,16	173,29

Observa-se portanto, uma redução de massa (peso) da longarina do chassi em torno de 38,58%, que gerará redução de custos de matéria-prima etc.

Os projetistas e engenheiros da indústria automotiva tem portanto uma nova forma de avaliar a topologia da longarina do chassis de seus caminhões. Foi possível verificar que para os 6 casos estudados há diferenças, sejam elas pelo tipo de abordagem, pelo tipo de malha MEF etc, e estas devem ser avaliadas pelo projetista para definir com qual método trabalhar, no entanto no aspecto qualitativo o método funciona e garante o que foi proposto como objetivo deste trabalho.

Verifica-se que para todos os casos as posições possíveis para as furos são em sua maior parte nas extremidades e no centro da longarina, o que justifica ter uma longarina com dois eixos (dianteiro e traseiro), em que os furos estão em maior parte distantes da posição dos eixos, em que nesta posição e na face superior em que estão aplicadas as cargas temos a predominância de elementos de cor escura.

---

## 8 Conclusões e observações

A premissa principal do projeto em fazer uma nova topologia de uma longarina de chassi veicular com o uso da otimização topológica foi demonstrada.

Simulou-se a técnica de otimização topológica 2D para identificação dos espaços vazios (recortes), e os elementos sólidos em que tem-se os furos. Esta sintetização foi feita com os dois programas computacionais Virtual.PYXIS (Virtual CAE) e ParetoWorks (Sciart), porém não apresentou-se uma técnica confiável para o caso de domínio 2D pois há uma formação de hinge na estrutura.

Selecionou-se e implementou-se um algoritmo 2D com base na abordagem SIMP e outro algoritmo 2D com base na abordagem de *Level Set*, idem ao tópico anterior também não apresentou resultados confiáveis devido a formação do hinge.

Sintetizou-se a longarina 3D com os programas computacionais ParetoWorks e Virtual.PYXIS, avaliando a influência da aba no resultado em relação a longarina 2D, este resultado foi bastante confiável, pois avaliou-se que as condições de formação de hinge deixam de existir neste caso, demonstrando de uma figura final com a forma básica de um domínio sintetizado, constituído por dois pontos fixos e cargas distribuídas superiores, sendo aproximados como a característica na literatura de uma ponte (*bridge*).

Como o tamanho da malha, devido a limitação do número de elementos para a versão estudantil do programa computacional ANSYS definiu-se em 10.000 elementos e o tamanho do elemento em aproximadamente 14 mm, e com base na aplicação da malha para os casos 2D de elemento quadrado e para o caso 3D de elementos tetraédricos, e como os furos ao longo da longarina são para parafusos de 1/2pol, ou seja para furos de 14 mm, provavelmente a malha nestes furos tenha transformado estes para o formato quadrado. Sugere-se também analisar posteriormente os casos estudados com elementos triangulares para o caso 2D ou tetraédricos para o caso 3D para verificar se há alteração nos resultados.

Comparou-se os resultados dos algoritmos e dos programas computacionais do domínio sintetizado e foi escolhido a análise 3D por ParetoWorks para projeto de nova longarina do chassi.

A nova longarina do chassi apresentou uma redução de massa de **38,58%** em relação a longarina original comercial.

Os dois programas computacionais e os algoritmos foram analisados com as mesmas cargas e fixações e domínio do projeto. ParetoWorks tem facilidade em processar 10.000

elementos nesta aplicação de longarina de chassi, tornando-se rápido para o processamento. Acima de 10.000 elementos o ParetoWorks calcula facilmente topologias mais suaves, porém foram utilizados para todas as análises 10.000 elementos devido a restrição no ANSYS student. O ParetoWorks foi o mais rápido em relação ao projetista preparar as condições iniciais de contorno no projeto, este fato é importante na indústria pois há uma diminuição da mão-de-obra do projetista para esta aplicação. Além disto, foi observado que para pequenas estruturas é possível preparar mais do que 1.000.000 de elementos facilmente neste programa computacional.

Virtual.PYXIS é muito eficiente para MOT, tem muitas opções de otimizadores, solvers e funções internas e por trabalhar com o solver (ANSYS student) é rápido para resolver o cálculo matemático do problema, pois este solver tem muito poder de processamento matemático. Além disto, com o Virtual.PYXIS é possível escolher algumas opções de otimizador para determinada estrutura, neste caso escolhemos o CO. Algumas opções existentes no programa computacional como a possibilidade de alterar o número de penalizações, faz com que tenhamos melhores resultados para a estrutura. Virtual.PYXIS tem uma pequena desvantagem para simulação pós-sintetização, pois há a necessidade de gerar um arquivo CAD STL ou IGES da sintetização de OT, refiná-lo no CAD e após inseri-lo no ANSYS para fazer a análise de simulação, enquanto no ParetoWorks, a simulação por elementos finitos já é feita no próprio comando após a análise de sintetização, eliminando a possibilidade de sair com um arquivo STL ou IGES e depois simulá-lo novamente.

Com os algoritmos é possível alterar rapidamente o código com as condições de contorno e trabalhar com diversas opções de resultado, foi possível então analisar a sintetização para várias condições de EE dos veículos. Porém observa-se a necessidade de aplicação de filtros adicionais para melhorar o contorno dos furos como resultado da sintetização, assim é sugerido pelo autor a possibilidade de adicionar algoritmos mais eficientes e que contenham por exemplo o filtro de densidade para suavizar os contornos do resultado da sintetização.

Os diferentes resultados encontrados para a topologia da longarina do chassi convergiram para uma restrição de volume determinada pelo usuário, independente de que cada programa computacional tem o seu específico algoritmo, e os algoritmos tem as suas abordagens e sensibilidades específicas. Foi observado um ótimo resultado para a análise 3D garantindo bons resultados de topologia para a longarina do chassi veicular. Assim a análise 3D foi considerada pelo autor como o melhor uso-benefício.

Os programas computacionais e algoritmos podem ser testados em diversos outros componentes do veículo, assim foi estudado o método MOT com bastante eficiência para a aplicação.

Percebe-se que a área de otimização estrutural é nova e que poderá ser bem utilizada para reduzir custos na indústria automotiva. Este trabalho se propôs a encontrar

na bibliografia e por meio dos programas computacionais e algoritmos, um novo modelo para descobrir a melhor topologia de furação da longarina do chassi de um veículo. Algoritmos específicos a esta aplicação, como já mencionado foram possíveis de serem implementados por meio deste estudo.

Como sugestão de futuros estudos apresenta-se os seguintes tópicos que poderão ser trabalhados.

- Desenvolvimento de um novo estudo com a inserção do comportamento dinâmico (modal) da longarina do chassi, como efeito de vibração, fadiga, etc pois só foi considerado para este trabalho o critério de análise estática, mas sabe-se que a longarina trabalha com esforços cíclicos, de torção e de resistência a colisão frontal e lateral.
- Implementação de um algoritmo 3D com base nesta aplicação, pois o intuito foi apenas demonstrar a existência de tais algoritmos como citado em [Liu e Tovar \(2013\)](#).
- Implementação de novos filtros como o filtro de densidade nos algoritmos.
- Como o trabalho baseou-se em identificar uma longarina, sugere-se para trabalhos futuros avaliar o quadro do chassi completo, considerando as longarinas esquerda e direita e as travessas. Uma sintetização com todos estes componentes passa a ser mais real do que simplesmente o estudo da longarina.
- Sugere-se incorporar aos algoritmos 2D a técnica de alocação de furos já no domínio da OT, pois os algoritmos deste trabalho discretizaram domínios cheios sem a condição inicial de ter furos ao longo da longarina. Diferentemente ao que foi feito com a análise 2D e 3D dos programas computacionais. Outra alternativa é reduzir o Volfrac das sintetizações, para ter um domínio sólido em que pode-se inserir diretamente os furos da longarina comercial sem a necessidade de movê-los de posição. Cabe ao projetista avaliar quais condições de Volfrac pode-se trabalhar.
- Propor um estudo de otimização paramétrica em relação a otimização topológica para avaliar grandes relações de aspectos como o caso estudado de uma longarina de chassi veicular.

Por fim, conclui-se com resultados favoráveis a hipótese do trabalho, que era fazer uma nova topologia de longarina de um chassi veicular com o uso da técnica de otimização topológica, assim o intuito do trabalho não é somente de implementar algoritmos e utilizar programas computacionais para sintetização de OT, mas também de aplicá-los em vários estudos de diversas peças/componentes na indústria automotiva. Espera-se que com esta nova proposta de definição de topologia de longarina de um chassi veicular, a indústria possa a utilizar os conceitos de otimização topológica como uma técnica de apoio à engenharia e de desenvolvimento para posterior fabricação do produto.

## Referências

- AAGE, N. et al. Interactive topology optimization on hand-held devices. ***Structural and Multidisciplinary Optimization***, v. 47, n. 1, p. 1–6, 2013. ISSN 1615-1488. Disponível em: <<http://dx.doi.org/10.1007/s00158-012-0827-z>>. Citado na página 69.
- ABBEY, T. Topology optimization methods - part i. ***DE - Digital Engineering***, 2017. Disponível em: <<http://www.digitaleng.news/de/topology-optimization/>>. Citado 4 vezes nas páginas 40, 41, 51 e 55.
- ABBEY, T. Topology optimization methods - part ii. ***DE - Digital Engineering***, 2017. Disponível em: <<http://www.digitaleng.news/de/topology-optimization-methods/>>. Citado 2 vezes nas páginas 44 e 48.
- AHMED, S. ***Topology optimization of load-carrying structures using three different types of finite elements***. Stockholm, Sweden: [s.n.], 2013. 1–84 p. Dissertação de Mestrado. Citado 9 vezes nas páginas 10, 35, 51, 55, 56, 59, 60, 69 e 70.
- ALBA, M. H. M. ***Otimização paramétrica de Chassi Veicular***. Brasília: [s.n.], 2015. 1–71 p. TCC de Engenharia Mecânica Automotiva. Citado na página 84.
- ALLAIRE, G.; JOUVE, F.; TOADER, A.-M. A level-set method for shape optimization. ***Comptes Rendus Mathématique***, v. 334, n. 12, p. 1125 – 1130, 2002. ISSN 1631-073X. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1631073X02024123>>. Citado 2 vezes nas páginas 37 e 44.
- ALLAIRE, G.; JOUVE, F.; TOADER, A.-M. Structural optimization using sensitivity analysis and a level-set method. ***Journal of Computational Physics***, v. 194, n. 1, p. 363 – 393, 2004. ISSN 0021-9991. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S002199910300487X>>. Citado na página 44.
- ALLAIRE, G.; PANTZ, O. Structural optimization with freefem++. ***Structural and Multidisciplinary Optimization***, v. 32, n. 3, p. 173–181, 2006. ISSN 1615-1488. Disponível em: <<http://dx.doi.org/10.1007/s00158-006-0017-y>>. Citado na página 69.
- ALPHEN, J. van. ***Structural optimization for 3D concrete printing***. Holanda: [s.n.], 2017. 1–115 p. Dissertação de Mestrado. Citado 6 vezes nas páginas 10, 44, 45, 46, 48 e 56.
- ALVES FILHO, A. ***Elementos finitos: a base da tecnologia CAE***. 1. ed. São Paulo: Ed. Érica, 2000. ISBN 85-7194-7414. Citado na página 24.
- AMIGO, R. C. R. ***Otimização e fabricação de dispositivos piezelétricos com gradação funcional de material***. São Paulo: [s.n.], 2013. 1–91 p. Dissertação de Mestrado. Citado 4 vezes nas páginas 10, 33, 34 e 38.



- AMSTUTZ, S. Connections between topological sensitivity analysis and material interpolation schemes in topology optimization. ***Structural and Multidisciplinary Optimization***, v. 43, n. 6, p. 755–765, 2011. ISSN 1615-1488. Disponível em: <<http://dx.doi.org/10.1007/s00158-010-0607-6>>. Citado na página 47.
- ANDREASSEN, E. et al. Efficient topology optimization in matlab® using 88 lines of code. ***Structural and Multidisciplinary Optimization***, v. 43, n. 1, p. 1–16, 2011. ISSN 1615-1488. Disponível em: <<http://dx.doi.org/10.1007/s00158-010-0594-7>>. Citado 5 vezes nas páginas 11, 68, 76, 97 e 116.
- ARORA, J. S. ***Optimization of structural and mechanical systems***. 1. ed. University of Iowa, USA: World Scientific Publishing Co. Pte. Ltd., 2007. 1–610 p. ISBN 978-981-256-962-2 (Print) 981-256-962-6 (Online). Citado 3 vezes nas páginas 15, 28 e 29.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. ***ABNT NBR 6656:2016***: Bobinas e chapas laminadas a quente de aço acalmado com características especiais de propriedades mecânicas, conformabilidade e soldabilidade - requisitos. Rio de Janeiro, 2016. 7 p. Citado 2 vezes nas páginas 102 e 103.
- BENDSØE, M. P. Optimal shape design as a material distribution problem. ***Structural optimization***, v. 1, n. 4, p. 193–202, 1989. ISSN 1615-1488. Disponível em: <<http://dx.doi.org/10.1007/BF01650949>>. Citado 3 vezes nas páginas 37, 38 e 55.
- BENDSØE, M. P. ***Optimization of structural topology, shape, and material***. New York: Springer-Verlag Berlin Heidelberg, 1995. v. 414, 1st edition. 1-271 p. ISBN 978-3-662-03117-9 (Print) 978-3-662-03115-5 (Online). Disponível em: <<http://dx.doi.org/10.1007/978-3-662-03115-5>>. Citado 2 vezes nas páginas 31 e 35.
- BENDSØE, M. P.; BEN-TAL, A.; ZOWE, J. Optimization methods for truss geometry and topology design. ***Structural optimization***, v. 7, n. 3, p. 141–159, 1994. ISSN 1615-1488. Disponível em: <<http://dx.doi.org/10.1007/BF01742459>>. Citado na página 29.
- BENDSØE, M. P.; KIKUCHI, N. Generating optimal topologies in structural design using a homogenization method. ***Computer Methods in Applied Mechanics and Engineering***, Elsevier, v. 71, n. 2, p. 197 – 224, 1988. ISSN 0045-7825. Disponível em: <<http://www.sciencedirect.com/science/article/pii/0045782588900862>>. Citado 4 vezes nas páginas 35, 37, 38 e 69.
- BENDSØE, M. P.; SIGMUND, O. Material interpolation schemes in topology optimization. ***Archive of Applied Mechanics***, v. 69, n. 9, p. 635–654, 1999. ISSN 1432-0681. Disponível em: <<http://dx.doi.org/10.1007/s004190050248>>. Citado 3 vezes nas páginas 42, 43 e 47.
- BENDSØE, M. P.; SIGMUND, O. ***Topology Optimization - Theory, Methods, and Applications***. Germany: Springer-Verlag Berlin Heidelberg, 2003. v. 1, 2nd edition. 1-380 p. ISBN 978-3-642-07698-5 (Print) 978-3-662-05086-6 (Online). Disponível em: <<http://dx.doi.org/10.1007/978-3-662-05086-6>>. Citado 9 vezes nas páginas 10, 15, 29, 30, 32, 35, 43, 70 e 73.
- BOURDIN, B.; CHAMBOLLE, A. Design-dependent loads in topology optimization. ***ESAIM: Control, Optimisation and Calculus of Variations***, EDP Sciences,

France, v. 9, p. 19–48, 001 2003. Disponível em: <<http://dx.doi.org/10.1051/cocv:2002070>>. Citado 2 vezes nas páginas 37 e 47.

CARVALHO, S. L.; ALMEIDA, S. R. M. Otimização de topologia de estruturas 2d e 3d. **SBPC - Sociedade brasileira para o Progresso da Ciência**, 63 Reunião Anual da SBPC - UFG - Goiânia - GO, p. 1–15, 2013. Disponível em: <[http://www.sbpnet.org.br/livro/63ra/conpeex/pivic/trabalhos/SOFIA\\_LE.PDF](http://www.sbpnet.org.br/livro/63ra/conpeex/pivic/trabalhos/SOFIA_LE.PDF)>. Citado 4 vezes nas páginas 10, 41, 49 e 65.

CAVAZZUTI, M. et al. High performance automotive chassis design: a topology optimization based approach. **Structural and Multidisciplinary Optimization**, v. 44, n. 1, p. 45–56, Jul 2011. ISSN 1615-1488. Disponível em: <<https://doi.org/10.1007/s00158-010-0578-7>>. Citado na página 84.

CHALLIS, V. J.; GUEST, J. K. Level set topology optimization of fluids in stokes flow. **International Journal for Numerical Methods in Engineering**, John Wiley and Sons, Ltd., v. 79, n. 10, p. 1284–1308, 2009. ISSN 1097-0207. Disponível em: <<http://dx.doi.org/10.1002/nme.2616>>. Citado 19 vezes nas páginas 11, 25, 47, 68, 77, 78, 79, 80, 81, 82, 95, 96, 97, 98, 99, 103, 110, 115 e 119.

CHEN, S. Y. Integrating ansys with modern numerical optimization technology - part ii : A reverse parametric modeling approach for structural shaping optimization. **FEA-Opt Technology**, p. 12p, 2000. Citado na página 84.

CHRISTENSEN, P.; KLARBRING, A. **An Introduction to Structural Optimization**. Springer Netherlands, 2009. v. 153. 1–211 p. (Solid Mechanics and Its Applications, v. 153). ISBN 978-1-4020-8665-6(Print) 9781402086663(Online). Disponível em: <<http://dx.doi.org/10.1007/978-1-4020-8666-3>>. Citado 6 vezes nas páginas 15, 29, 30, 31, 32 e 35.

CROCIATA, D. D. **Evaluation of design and optimization software for Additive Manufacturing with focus on topology optimization**. Suécia: [s.n.], 2016. 1–122 p. Dissertação de Mestrado. Citado na página 83.

DÍAAZ, A. R.; KIKUCHI, N. Solutions to shape and topology eigenvalue optimization problems using a homogenization method. **International Journal for Numerical Methods in Engineering**, John Wiley and Sons, Ltd, v. 35, n. 7, p. 1487–1502, 1992. ISSN 1097-0207. Disponível em: <<http://dx.doi.org/10.1002/nme.1620350707>>. Citado 2 vezes nas páginas 36 e 38.

DANTZIG, G. B. **Linear programming and extensions**. [S.l.]: Princeton Univ. Press, 1963. First edition. Citado na página 56.

DIAZ, A.; SIGMUND, O. Checkerboard patterns in layout optimization. **Structural optimization**, v. 10, n. 1, p. 40–45, 1995. ISSN 1615-1488. Disponível em: <<http://dx.doi.org/10.1007/BF01743693>>. Citado na página 50.

DIJK, N. P. van et al. Level-set methods for structural topology optimization: a review. **Structural and Multidisciplinary Optimization**, v. 48, n. 3, p. 437–472, 2013. ISSN 1615-1488. Disponível em: <<http://dx.doi.org/10.1007/s00158-013-0912-y>>. Citado na página 44.

- DIOGO, R. J. A. *Design of a High Aspect Ratio Wing Box using an Ant Colony Optimization Algorithm*. 2016. Master Thesis. Citado 2 vezes nas páginas 70 e 71.
- DORN, W. S. Automatic design of optimal structures. *Journal de mecanique*, v. 3, p. 25–52, 1964. Citado 3 vezes nas páginas 28, 29 e 38.
- ESCHENAUER, H. A.; KOBELEV, V. V.; SCHUMACHER, A. Bubble method for topology and shape optimization of structures. *Structural optimization*, v. 8, n. 1, p. 42–51, 1994. ISSN 1615-1488. Disponível em: <<http://dx.doi.org/10.1007/BF01742933>>. Citado na página 47.
- ESCHENAUER, H. A.; OLHOFF, N. Topology optimization of continuum structures: a review. *Applied Mechanics Reviews*, American Society of Mechanical Engineers, v. 54, n. 4, p. 331–390, 2001. Citado 2 vezes nas páginas 10 e 43.
- EUROPEAN COMMITTEE FOR STANDARDIZATION. 2007/46/EC: **Standard Directive for Vehicles**. EUR-Lex, 2007. 15 p. Disponível em: <<http://eur-lex.europa.eu/legal-content/EN/TXT/?qid=1486430611779&uri=CELEX:32007L0046>>. Citado na página 87.
- FUJII, D.; CHEN, B.; KIKUCHI, N. Composite material design of two-dimensional structures using the homogenization design method. *International Journal for Numerical Methods in Engineering*, Wiley Online Library, v. 50, n. 9, p. 2031–2051, 2001. Citado na página 38.
- FURTADO, D. C. *Análise estrutural de chassi de veículos automotivos*. Brasília: [s.n.], 2013. 1–88 p. TCC de Engenharia Mecânica Automotiva. Citado na página 84.
- GODOY, A. S. Pesquisa qualitativa: tipos fundamentais. *Revista de Administracao de Empresas*, scielo, v. 35, p. 20 – 29, 06 1995. ISSN 0034-7590. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S0034-75901995000300004&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0034-75901995000300004&nrm=iso)>. Citado na página 89.
- GRIVA, I.; NASH, S. G.; SOFER, A. *Linear and nonlinear optimization*. 2. ed. Siam, 2009. 1–742 p. ISBN 978-0-898716-61-0. Disponível em: <<http://bookstore.siam.org/OT108/>>. Citado 2 vezes nas páginas 10 e 54.
- GUEST, J. K.; PRÉVOST, J. H.; BELYTSCHKO, T. Achieving minimum length scale in topology optimization using nodal design variables and projection functions. *International Journal for Numerical Methods in Engineering*, John Wiley and Sons, Ltd., v. 61, n. 2, p. 238–254, 2004. ISSN 1097-0207. Disponível em: <<http://dx.doi.org/10.1002/nme.1064>>. Citado na página 97.
- HAFTKA, R.; GÜRDAL, Z. *Elements of Structural Optimization*: Solid mechanics and its applications. 3. ed. Dordrecht, The Netherlands: Kluwer Academic Publishers Norwell, MA, 1992. v. 11. ISBN 978-0-7923-1505-6 (Print) 978-94-011-2550-5 (Online). Disponível em: <<http://dx.doi.org/10.1007/978-94-011-2550-5>>. Citado 4 vezes nas páginas 29, 30, 31 e 56.
- HASSANI, B.; HINTON, E. A review of homogenization and topology optimization i—homogenization theory for media with periodic structure. *Computers and Structures*, Elsevier, v. 69, n. 6, p. 707–717, 1998. Citado na página 38.

- HASSANI, B.; HINTON, E. **Homogenization and structural topology optimization**: : theory, practice and software. 1. ed. London: Springer-Verlag London Limited, 1999. ISBN 978-1-4471-1229-7 (print) 978-1-4471-0891-7 (Online). Disponível em: <<http://dx.doi.org/10.1007/978-1-4471-0891-7>>. Citado na página 38.
- HEMP, W. Michell's structural continua. **Optimum Structures**, Clarendon Press Oxford, p. 70–101, 1973. Citado 3 vezes nas páginas 28, 29 e 38.
- HUNTER, W. **Predominantly solid-void three-dimensional topology optimisation using open source software**. Tese (Doutorado) — Stellenbosch: University of Stellenbosch, 2009. Citado na página 69.
- JENKINS, B. **Design optimization: Topology and much more**. 2015. Disponível em: <<http://www.3dcadworld.com/design-optimization-topology-and-much-more>>. Citado 6 vezes nas páginas 10, 15, 31, 32, 33 e 34.
- JOG, C. S.; HABER, R. B. Stability of finite element models for distributed-parameter optimization and topology design. **Computer Methods in Applied Mechanics and Engineering**, v. 130, n. 3, p. 203 – 226, 1996. ISSN 0045-7825. Disponível em: <<http://www.sciencedirect.com/science/article/pii/0045782595009280>>. Citado na página 50.
- KAPLAN, A.; HEGENBERG, L.; MOTA, O. S. da. **Conduta na pesquisa: metodologia para as ciências do comportamento**. São Paulo: Herder, 1972. Citado na página 88.
- LAVRITA. . SBC, 2016. Disponível em: <<http://www.lavrita.com.br/equipamentos/combate-a-incendio-aeroporto.html>>. Citado 2 vezes nas páginas 12 e 101.
- LIBERA, W. M. de M.; LIMA, C. R. de. **Aplicação do método de otimização topológica ao projeto de estruturas aeroespaciais**. Santo André, 2012. Disponível em: <[http://ic.ufabc.edu.br/II\\_SIC\\_UFABC/resumos/paper\\_5\\_105.pdf](http://ic.ufabc.edu.br/II_SIC_UFABC/resumos/paper_5_105.pdf)>. Citado 2 vezes nas páginas 11 e 96.
- LIMA, C. R. d. **Projeto de mecanismos flexíveis usando o método de otimização topológica**. 2002. Dissertação de Mestrado. Citado 10 vezes nas páginas 10, 54, 55, 56, 58, 59, 60, 61, 63 e 67.
- LIMA, C. R. d. **Estudo da Obtenção de Imagens de Tomografia de Impedância Elétrica do Pulmão pelo Método de Otimização Topológica**. 2006. Tese de Doutorado. Citado 2 vezes nas páginas 36 e 50.
- LIU, K.; TOVAR, A. An efficient 3d topology optimization code written in matlab®. **Structural and Multidisciplinary Optimization**, v. 50, n. 6, p. 1175–1196, 2013. ISSN 1615-1488. Disponível em: <<http://dx.doi.org/10.1007/s00158-014-1107-x>>. Citado 5 vezes nas páginas 35, 67, 69, 82 e 146.
- LIU, Z.; KORVINK, J.; HUANG, R. Structure topology optimization: fully coupled level set method via femlab. **Structural and Multidisciplinary Optimization**, v. 29, n. 6, p. 407–417, 2005. ISSN 1615-1488. Disponível em: <<http://dx.doi.org/10.1007/s00158-004-0503-z>>. Citado na página 69.

MARTINEZ, J. M.; SANTOS, S. A. **Métodos computacionais de otimização.** *Colóquio Brasileiro de Matemática, Apostilas*, v. 20, 1998. Citado na página 55.

MARTINS, A. C. d. S. **O método de pontos interiores no planejamento da radioterapia.** Universidade Estadual Paulista (UNESP), 2011. Dissertação de Mestrado. Disponível em: <[http://www2.ibb.unesp.br/posgrad/teses/biometria\\_me\\_2011\\_andrea\\_martins.pdf](http://www2.ibb.unesp.br/posgrad/teses/biometria_me_2011_andrea_martins.pdf)>. Citado na página 67.

MATSUI, K.; TERADA, K. Continuous approximation of material distribution for topology optimization. *International Journal for Numerical Methods in Engineering*, John Wiley and Sons, Ltd., v. 59, n. 14, p. 1925–1944, 2004. ISSN 1097-0207. Disponível em: <<http://dx.doi.org/10.1002/nme.945>>. Citado na página 50.

MESSAC, A. **Optimization in Practice with MATLAB®: For Engineering Students and Professionals.** [S.l.]: Cambridge University Press, 2015. Citado 3 vezes nas páginas 10, 67 e 68.

MICHELL, A. G. M. Lviii. the limits of economy of material in frame-structures. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, Taylor & Francis, v. 8, n. 47, p. 589–597, 1904. Citado 4 vezes nas páginas 10, 28, 29 e 38.

MLEJNEK, H. P. Some aspects of the genesis of structures. *Structural optimization*, v. 5, n. 1, p. 64–69, 1992. ISSN 1615-1488. Disponível em: <<http://dx.doi.org/10.1007/BF01744697>>. Citado na página 38.

NISHIWAKI, S. et al. Topology optimization of compliant mechanisms using the homogenization method. *International Journal for Numerical Methods in Engineering*, John Wiley and Sons, Ltd, v. 42, n. 3, p. 535–559, 1998. ISSN 1097-0207. Disponível em: <[http://dx.doi.org/10.1002/\(SICI\)1097-0207\(19980615\)42:3<535::AID-NME372>3.0.CO;2-J](http://dx.doi.org/10.1002/(SICI)1097-0207(19980615)42:3<535::AID-NME372>3.0.CO;2-J)>. Citado na página 36.

OLIVEIRA, F. C. G. **Contribuição ao desenvolvimento de uma estrutura veicular tipo spaceframe usando o método dos elementos finitos e métodos heurísticos de otimização numérica.** Uberlândia: [s.n.], 2007. 1–158 p. Dissertação de Mestrado. Citado na página 84.

OSHER, S.; SETHIAN, J. A. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, v. 79, n. 1, p. 12 – 49, 1988. ISSN 0021-9991. Disponível em: <<http://www.sciencedirect.com/science/article/pii/0021999188900022>>. Citado na página 44.

PARK, Y. K. **Extensions of optimal layout design using the homogenization method.** Tese (Doutorado) — -, University of Michigan, 1995. Disponível em: <<https://books.google.com.br/books?id=lbYeAQAAMAAJ>>. Citado na página 36.

PETERSSON, J. A finite element analysis of optimal variable thickness sheets. *SIAM Journal on Numerical Analysis*, Society for Industrial and Applied Mathematics, v. 36, n. 6, p. 1759–1778, 1999. ISSN 00361429. Disponível em: <<http://www.jstor.org/stable/2587221>>. Citado na página 43.



- PRAGER, W. A note on discretized michell structures. *Computer Methods in Applied Mechanics and Engineering*, Elsevier, v. 3, n. 3, p. 349–355, 1974. Citado 3 vezes nas páginas 28, 29 e 38.
- PRETE, A. D.; MAZZOTTA, D.; ANGLANI, A. Design optimization application in accordance with product and process requirements. *Advances in Engineering Software*, Elsevier, v. 41, n. 3, p. 427–432, 2010. Disponível em: <<https://doi-org.ez338.periodicos.capes.gov.br/10.1016/j.advengsoft.2009.03.007>>. Citado na página 84.
- RAMOS, P. S. *Estruturas Termoelásticas sob Otimização Topológica e H-adaptatividade*. Natal-RN: [s.n.], 2007. 1–137 p. Dissertação de Mestrado. Citado 3 vezes nas páginas 10, 28 e 42.
- RAO, S. S. *Engineering optimization: theory and practice*. John Wiley and Sons, Inc., 2009. v. 4nd edition. 1–830 p. ISBN 9780470549124. Disponível em: <<http://dx.doi.org/10.1002/9780470549124>>. Citado 8 vezes nas páginas 10, 15, 29, 35, 36, 52, 55 e 57.
- RAVINDRAN, A.; REKLAITIS, G. V.; RAGSDALL, K. M. *Engineering optimization: methods and applications*. [S.l.]: John Wiley & Sons, 2006. Citado na página 55.
- ROZVANY, G.; ZHOU, M. The coc algorithm, part i: Cross section optmization or sizing. *Computer Methods in Applied Mechanics and Engineering*, v. 89, n. 1, p. 281 – 3308, 1991. ISSN 0045-7825. Disponível em: <[https://doi.org/10.1016/0045-7825\(91\)90045-8](https://doi.org/10.1016/0045-7825(91)90045-8)>. Citado na página 29.
- ROZVANY, G. I. *Topology optimization in structural mechanics*. Wien: Springer-Verlag Wien GmbH, 1997. v. 374. ISBN 978-3-211-82907-3 (Print) 978-3-7091-2566-3 (Online). Disponível em: <<http://dx.doi.org/10.1007/978-3-7091-2566-3>>. Citado 2 vezes nas páginas 31 e 35.
- ROZVANY, G. I. et al. On the solid plate paradox in structural optimization. *Journal of Structural Mechanics*, Taylor and Francis, v. 10, n. 1, p. 1–32, 1982. Citado na página 35.
- SCANIA(a). *Informações sobre chassis e designacoes de modelo - Scania Bodybuilder*. Bodybuilding information, 2016. Disponível em: <[https://til.scania.com/w/bwm\\_0000027\\_16](https://til.scania.com/w/bwm_0000027_16)>. Citado 2 vezes nas páginas 11 e 85.
- SCANIA(b). *Modificacoes da distancia de eixo - Scania Bodybuilder*. Bodybuilding information, 2016. Disponível em: <[https://til.scania.com/w/bwm\\_0000273\\_16](https://til.scania.com/w/bwm_0000273_16)>. Citado 3 vezes nas páginas 11, 86 e 87.
- SCANIA(c). *Perfuracao - Scania Bodybuilder*. Bodybuilding information, 2016. Disponível em: <[https://til.scania.com/w/bwm\\_0000380\\_16](https://til.scania.com/w/bwm_0000380_16)>. Citado 2 vezes nas páginas 11 e 87.
- SEOW, Y. E. V. *Aeroelastic Topology Optmization of Aircraft High Aspect Ratio Wings*. 2016. Master Thesis. Citado na página 70.
- SIGMUND, O. *Design of Materials Structures Using Topology Optimization*. Tese (Doutorado) — Technical University of Denmark, Lyngby, Denmark, 1994. Tese. Citado na página 35.

- SIGMUND, O. On the design of compliant mechanisms using topology optimization. *Mechanics of structures and machines (Print)*, Marcel Dekker Inc., v. 25, n. 4, p. 493–524, 1997. ISSN 0890-5452. Citado na página 36.
- SIGMUND, O. A 99 line topology optimization code written in matlab®. *Structural and Multidisciplinary Optimization*, v. 21, n. 2, p. 120–127, 2001. ISSN 1615-1488. Disponível em: <<http://dx.doi.org/10.1007/s001580050176>>. Citado 25 vezes nas páginas 10, 11, 25, 40, 51, 64, 65, 67, 68, 69, 70, 72, 73, 74, 75, 77, 82, 95, 97, 98, 99, 103, 110, 115 e 116.
- SIGMUND, O. Morphology-based black and white filters for topology optimization. *Structural and Multidisciplinary Optimization*, v. 33, n. 4, p. 401–424, 2007. ISSN 1615-1488. Disponível em: <<http://dx.doi.org/10.1007/s00158-006-0087-x>>. Citado na página 97.
- SIGMUND, O.; MAUTE, K. Topology optimization approaches. *Structural and Multidisciplinary Optimization*, v. 48, n. 6, p. 1031–1055, 2013. ISSN 1615-1488. Disponível em: <<http://dx.doi.org/10.1007/s00158-013-0978-6>>. Citado 6 vezes nas páginas 10, 15, 32, 34, 35 e 37.
- SIGMUND, O.; PETERSSON, J. Numerical instabilities in topology optimization: A survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Structural optimization*, v. 16, n. 1, p. 68–75, 1998. ISSN 1615-1488. Disponível em: <<http://dx.doi.org/10.1007/BF01214002>>. Citado na página 50.
- SILVA, E. C. N. Otimização aplicada ao projeto de sistemas mecânicos: Apostila para disciplina de pós-graduação do departamento de engenharia mecatronica. *PMR 5215 - Otimização aplicada ao projeto de sistemas mecânicos*, p. 154p, 2003. Disponível em: <<http://sites.poli.usp.br/d/pmr5215/a1-5215.pdf>>. Citado 3 vezes nas páginas 15, 28 e 29.
- SILVA, E. C. N. Técnicas de otimização aplicadas no projeto de peças mecânicas: Apostila para disciplina de pós-graduação do departamento de engenharia mecatronica. *PMR 5215 - Otimização aplicada ao projeto de sistemas mecânicos*, p. 23p, 2003. Disponível em: <<http://sites.poli.usp.br/d/pmr5215/otimizacao.pdf>>. Acesso em: 23 abr. 2017. Citado 3 vezes nas páginas 10, 39 e 50.
- SILVA, J. G. S. Análise estrutural de chassis de veículos pesados com base no emprego do programa ansys. *Cobenge 2001*, p. 8p, 2001. Citado na página 84.
- SOKÓŁ, T. A 99 line code for discretized michell truss optimization written in mathematica. *Structural and Multidisciplinary Optimization*, v. 43, n. 2, p. 181–190, 2011. ISSN 1615-1488. Disponível em: <<http://dx.doi.org/10.1007/s00158-010-0557-z>>. Citado na página 69.
- SOKOLOWSKI, J.; ZOCHOWSKI, A. On the topological derivative in shape optimization. *SIAM Journal on Control and Optimization*, v. 37, n. 4, p. 1251–1272, 1999. Disponível em: <<http://dx.doi.org/10.1137/S0363012997323230>>. Citado na página 37.
- SURESH, K. A 199-line matlab® code for pareto-optimal tracing in topology optimization. *Structural and Multidisciplinary Optimization*, v. 42, n. 5, p. 665–679, 2010. ISSN 1615-1488. Disponível em: <<http://dx.doi.org/10.1007/s00158-010-0534-6>>. Citado 4 vezes nas páginas 25, 47, 68 e 77.

- SURESH, K. *Introduction to Design Optimization*. 1. ed. National Science Foundation, 2016. 1–368 p. Disponível em: <<http://www.ersl.wisc.edu/publications/2016/OptimizationTextTOC.pdf>>. Citado 4 vezes nas páginas 10, 44, 58 e 59.
- SVANBERG, K. The method of moving asymptotes—a new method for structural optimization. *International Journal for Numerical Methods in Engineering*, John Wiley and Sons, Ltd, v. 24, n. 2, p. 359–373, 1987. ISSN 1097-0207. Disponível em: <<http://dx.doi.org/10.1002/nme.1620240207>>. Citado 2 vezes nas páginas 54 e 56.
- THOMSEN, J. Topology optimization of structures composed of one or two materials. *Structural and Multidisciplinary Optimization*, Springer, v. 5, n. 1, p. 108–115, 1992. Citado na página 38.
- VANDERPLAATS, G. N. *Numerical optimization techniques for engineering design*. 3. ed. [S.l.]: McGraw-Hill College, 1999. 1–441 p. ISBN 0-944956-00-9. Citado 2 vezes nas páginas 30 e 31.
- VICTORIA, M.; MARTÍ, P.; QUERIN, O. M. Topology design of two-dimensional continuum structures using isolines. *Computers and Structures*, v. 87, n. 1–2, p. 101 – 109, 2009. ISSN 0045-7949. Disponível em: <<http://doi.org/10.1016/j.compstruc.2008.08.001>>. Citado na página 28.
- VIGDERGAUZ, S. Energy-minimizing inclusions in a planar elastic structure with macroisotropy. *Structural optimization*, v. 17, n. 2, p. 104–112, 1999. ISSN 1615-1488. Disponível em: <<http://dx.doi.org/10.1007/BF01195935>>. Citado na página 47.
- VIRTUAL.PYXIS. Virtual.pyxis lcc/gmbh, virtual.pyxis input deck manual, v2.10 m00. *Developer Team*, Chicago (USA) e Hannover (DE), 2016. Citado 3 vezes nas páginas 13, 122 e 123.
- WANG, M.; CHEN, S.; XIA, Q. **Structural topology optimization with the level set method**. URL <http://ihome.ust.hk/~mywang/Download.html>, 2004. Citado na página 68.
- WANG, M. Y.; WANG, X.; GUO, D. A level set method for structural topology optimization. *Computer Methods in Applied Mechanics and Engineering*, v. 192, n. 1–2, p. 227 – 246, 2003. ISSN 0045-7825. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0045782502005595>>. Citado na página 44.
- WU, J.; WU, L. Revised level set-based method for topology optimization and its applications in bridge construction. *The Open Civil Engineering Journal*, p. 14p, 2016. Citado 2 vezes nas páginas 11 e 96.
- XIE, Y.; STEVEN, G. A simple evolutionary procedure for structural optimization. *Computers and Structures*, v. 49, n. 5, p. 885 – 896, 1993. ISSN 0045-7949. Disponível em: <[http://dx.doi.org/10.1016/0045-7949\(93\)90035-C](http://dx.doi.org/10.1016/0045-7949(93)90035-C)>. Citado 2 vezes nas páginas 37 e 48.
- ZHAO, Y.-b. A novel substructure-based topology optimization method for the design of wing structure. 2016. Article. Citado 3 vezes nas páginas 10, 70 e 71.



ZHONG, W. et al. Multi-objective topology and sizing optimization of bus body frame. ***Structural and Multidisciplinary Optimization***, v. 54, n. 3, p. 701–714, Sep 2016. ISSN 1615-1488. Disponível em: <<https://doi.org/10.1007/s00158-016-1431-4>>. Citado na página 84.

ZHOU, M.; ROZVANY, G. The coc algorithm, part ii: Topological, geometrical and generalized shape optimization. ***Computer Methods in Applied Mechanics and Engineering***, v. 89, n. 1, p. 309 – 336, 1991. ISSN 0045-7825. Disponível em: <[http://dx.doi.org/10.1016/0045-7825\(91\)90046-9](http://dx.doi.org/10.1016/0045-7825(91)90046-9)>. Citado 2 vezes nas páginas 37 e 38.

ZHOU, S.; WANG, M. Y. **3d structural topology optimization with the simp method**. 2005. Disponível em: <<http://www2.acae.cuhk.edu.hk/cmdl/download.htm>>. Citado na página 69.

# APÊNDICE A – Algoritmo 2D SIMP de Sigmund, 2001

```

1  %%%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLE SIGMUND, JANUARY 2000 %%%
2  %%%% CODE MODIFIED FOR INCREASED SPEED, September 2002, BY OLE SIGMUND %%%
3  function top99(nelx,nely,volfrac,penal,rmin)
4  % INITIALIZE
5  x(1:nely,1:nelx) = volfrac;
6  loop = 0;
7  change = 1.;
8  % START ITERATION
9  while change > 0.01
10     loop = loop + 1;
11     xold = x;
12     % FE-ANALYSIS
13     [U]=FE(nelx,nely,x,penal);
14     % OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
15     [KE] = lk;
16     c = 0.;
17     for ely = 1:nely
18         for elx = 1:nelx
19             n1 = (nely+1)*(elx-1)+ely;
20             n2 = (nely+1)* elx +ely;
21             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);
22             c = c + x(ely,elx)^penal*Ue'*KE*Ue;
23             dc(ely,elx) = -penal*x(ely,elx)^(penal-1)*Ue'*KE*Ue;
24         end
25     end
26     % FILTERING OF SENSITIVITIES
27     [dc] = check(nelx,nely,rmin,x,dc);
28     % DESIGN UPDATE BY THE OPTIMALITY CRITERIA METHOD
29     [x] = OC(nelx,nely,x,volfrac,dc);
30     % PRINT RESULTS
31     change = max(max(abs(x-xold)));
32     disp([' It.: ' sprintf('%4i',loop) ' Obj.: ' sprintf('%10.4f',c) ...
33           ' Vol.: ' sprintf('%6.3f',sum(sum(x))/(nelx*nely)) ...
34           ' ch.: ' sprintf('%6.3f',change )])
35     % PLOT DENSITIES
36     colormap(gray); imagesc(-x); axis equal; axis tight; axis off;pause(1e-6)
37 end
38 %%%%%%%%%% OPTIMALITY CRITERIA UPDATE %%%%%%%%%%
39 function [xnew]=OC(nelx,nely,x,volfrac,dc)
40 l1 = 0; l2 = 100000; move = 0.2;
41 while (l2-l1 > 1e-4)
42     lmid = 0.5*(l2+l1);
43     xnew = max(0.001,max(x-move,min(1.,min(x+move,x.*sqrt(-dc./lmid))));
44     if sum(sum(xnew)) - volfrac*nelx*nely > 0;
45         l1 = lmid;

```

```

46 -     else
47 -         l2 = lmid;
48 -     end
49 - end
50 - %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
51 - function [dcn]=check(nelx,nely,rmin,x,dc)
52 - dcn=zeros(nely,nelx);
53 - for i = 1:nelx
54 -     for j = 1:nely
55 -         sum=0.0;
56 -         for k = max(i-floor(rmin),1):min(i+floor(rmin),nelx)
57 -             for l = max(j-floor(rmin),1):min(j+floor(rmin),nely)
58 -                 fac = rmin-sqrt((i-k)^2+(j-l)^2);
59 -                 sum = sum+max(0,fac);
60 -                 dcn(j,i) = dcn(j,i) + max(0,fac)*x(1,k)*dc(1,k);
61 -             end
62 -         end
63 -         dcn(j,i) = dcn(j,i)/(x(j,i)*sum);
64 -     end
65 - end
66 - %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
67 - function [U]=FE(nelx,nely,x,penal)
68 - [KE] = lk;
69 - K = sparse(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1));
70 - F = sparse(2*(nely+1)*(nelx+1),1); U = zeros(2*(nely+1)*(nelx+1),1);
71 - for elx = 1:nelx
72 -     for ely = 1:nely
73 -         n1 = (nely+1)*(elx-1)+ely;
74 -         n2 = (nely+1)* elx +ely;
75 -         edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1; 2*n2+2; 2*n1+1; 2*n1+2];
76 -         K(edof,edof) = K(edof,edof) + x(ely,elx)^penal*KE;
77 -     end
78 - end
79 - % DEFINE LOADS AND SUPPORTS (HALF MBB-BEAM)
80 - F(2,1) = -1;
81 - fixeddofs = union([1:2:2*(nely+1)], [2*(nelx+1)*(nely+1)]);
82 - alldofs = [1:2*(nely+1)*(nelx+1)];
83 - freedofs = setdiff(alldofs,fixeddofs);
84 - % SOLVING
85 - U(freedofs,:) = K(freedofs,freedofs) \ F(freedofs,:);
86 - U(fixeddofs,:)= 0;
87 - %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
88 - function [KE]=lk
89 - E = 1.;
90 - nu = 0.3;
91 - k=[ 1/2-nu/6    1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 ...
92 -    -1/4+nu/12 -1/8-nu/8  nu/6      1/8-3*nu/8];
93 - KE = E/(1-nu^2)*[ k(1) k(2) k(3) k(4) k(5) k(6) k(7) k(8)
94 -                  k(2) k(1) k(8) k(7) k(6) k(5) k(4) k(3)
95 -                  k(3) k(8) k(1) k(6) k(7) k(4) k(5) k(2)
96 -                  k(4) k(7) k(6) k(1) k(8) k(3) k(2) k(5)
97 -                  k(5) k(6) k(7) k(8) k(1) k(2) k(3) k(4)
98 -                  k(6) k(5) k(4) k(3) k(2) k(1) k(8) k(7)
99 -                  k(7) k(4) k(5) k(2) k(3) k(8) k(1) k(6)
100 -                  k(8) k(3) k(2) k(5) k(4) k(7) k(6) k(1)];

```

## APÊNDICE B – Algoritmo 2D SIMP Implementado

```

1      %%%% CÓDIGO 99 LINHAS BASEADO EM SIGMUND, 2001 %%%
2      %%%% CODIGO MODIFICADO PARA ESTUDAR LONGARINAS, 2017, POR PAULO LIXANDRAO %
3      function topchassi(nelx,nely,volfrac,penal,rmin)
4      % INICIALIZAÇÃO
5      x(1:nely,1:nelx) = volfrac;
6      loop = 0;
7      change = 1.;
8      % INÍCIO DAS ITERAÇÕES
9      while change > 0.1
10         loop = loop + 1;
11         xold = x;
12         % ANÁLISE FEA
13         [U]=FE(nelx,nely,x,penal);
14         % FUNÇÃO OBJETIVA E ANÁLISE DE SENSIBILIDADE
15         [KE] = lk;
16         c = 0.;
17         for ely = 1:nely
18             for elx = 1:nelx
19                 n1 = (nely+1)*(elx-1)+ely;
20                 n2 = (nely+1)* elx +ely;
21                 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);
22                 c = c + x(ely,elx)^penal*Ue'*KE*Ue;
23                 dc(ely,elx) = -penal*x(ely,elx)^(penal-1)*Ue'*KE*Ue;
24             end
25         end
26         % FILTRO DE SENSIBILIDADE
27         [dc] = check(nelx,nely,rmin,x,dc);
28         % ATUALIZAÇÃO DAS DENSIDADES
29         [x] = OC(nelx,nely,x,volfrac,dc);
30         % MOSTRA OS RESULTADOS
31         change = max(max(abs(x-xold)));
32         disp([' It.: ' sprintf('%4i',loop) ' Obj.: ' sprintf('%10.4f',c) ...
33             ' Vol.: ' sprintf('%6.3f',sum(sum(x))/(nelx*nely)) ...
34             ' ch.: ' sprintf('%6.3f',change )])
35         % PLOTA A FIGURA EM ESCALA DE CINZA
36         colormap(gray); imagesc(-x); axis equal; axis tight; axis off; pause(1e-6)
37     end
38     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
39     function [xnew]=OC(nelx,nely,x,volfrac,dc)
40     l1 = 0; l2 = 100000; move = 0.2;
41     while (l2-l1 > 1e-4)
42         lmid = 0.5*(l2+l1);
43         xnew = max(0.001,max(x-move,min(1.,min(x+move,x.*sqrt(-dc./lmid)))));
44         if sum(sum(xnew)) - volfrac*nelx*nely > 0;
45             l1 = lmid;

```



```

46 -     else
47 -         l2 = lmid;
48 -     end
49 - end
50 - %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
51 - function [dcn]=check(nelx,nely,rmin,x,dc)
52 -     dcn=zeros(nely,nelx);
53 -     for i = 1:nelx
54 -         for j = 1:nely
55 -             sum=0.0;
56 -             for k = max(i-floor(rmin),1):min(i+floor(rmin),nelx)
57 -                 for l = max(j-floor(rmin),1):min(j+floor(rmin),nely)
58 -                     fac = rmin-sqrt((i-k)^2+(j-l)^2);
59 -                     sum = sum+max(0,fac);
60 -                     dcn(j,i) = dcn(j,i) + max(0,fac)*x(l,k)*dc(l,k);
61 -                 end
62 -             end
63 -             dcn(j,i) = dcn(j,i)/(x(j,i)*sum);
64 -         end
65 -     end
66 - %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
67 - function [U]=FE(nelx,nely,x,penal)
68 -     [KE] = lk;
69 -     K = sparse(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1));
70 -     F = sparse(2*(nely+1)*(nelx+1),1); U = zeros(2*(nely+1)*(nelx+1),1);
71 -     for elx = 1:nelx+1
72 -         for ely = 1:nely
73 -             n1 = (nely+1)*(elx-1)+ely;
74 -             n2 = (nely+1)* elx +ely;
75 -             edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1; 2*n2+2; 2*n1+1; 2*n1+2];
76 -             K(edof,edof) = K(edof,edof) + x(ely,elx)^penal*KE;
77 -         end
78 -     end
79 -     % DEFINIÇÃO DE CARGAS E SUPORTES (CASO LONGARINA DO CHASSI)
80 -     for elx = 1:nelx
81 -         F(2*(nely+1)*(elx-1)+2,1)= -Ftotal/(nelx+1);
82 -     end
83 -     fixeddofs = union([2*(nely+1)*nsup1:nsup2],[2*(nely+1)*nsup3:nsup4]);
84 -     alldofs = [1:2*(nely+1)*(nelx+1)];
85 -     freedofs = setdiff(alldofs,fixeddofs);
86 -     % CONDIÇÃO DOS PONTOS DE FIXAÇÃO
87 -     U(freedofs,:) = K(freedofs,freedofs) \ F(freedofs,:);
88 -     U(fixeddofs,:)= 0;
89 - %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
90 - function [KE]=lk
91 -     E = 1.;
92 -     nu = 0.3;
93 -     k=[ 1/2-nu/6    1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 ...
94 -        -1/4+nu/12 -1/8-nu/8  nu/6      1/8-3*nu/8];
95 -     KE = E/(1-nu^2)*[ k(1) k(2) k(3) k(4) k(5) k(6) k(7) k(8)
96 -                      k(2) k(1) k(8) k(7) k(6) k(5) k(4) k(3)
97 -                      k(3) k(8) k(1) k(6) k(7) k(4) k(5) k(2)
98 -                      k(4) k(7) k(6) k(1) k(8) k(3) k(2) k(5)
99 -                      k(5) k(6) k(7) k(8) k(1) k(2) k(3) k(4)
100 -                     k(6) k(5) k(4) k(3) k(2) k(1) k(8) k(7)
101 -                     k(7) k(4) k(5) k(2) k(3) k(8) k(1) k(6)
102 -                     k(8) k(3) k(2) k(5) k(4) k(7) k(6) k(1)];

```

# APÊNDICE C – Algoritmo 2D LS de Challis, 2009

```

1  %% TOPOLOGY OPTIMIZATION USING THE LEVEL-SET METHOD, VIVIEN J. CHALLIS 2009
2  function [struc] = top_levelset(nelx,nely,volReq,stepLength,numReinit,topWeight)
3  % Initialization
4  struc = ones(nely,nelx);
5  [lsf] = reinit(struc);
6  shapeSens = zeros(nely,nelx); topSens = zeros(nely,nelx);
7  [KE,KTr,lambda,mu] = materialInfo();
8  % Main loop:
9  for iterNum = 1:200
10 % FE-analysis, calculate sensitivities
11 [U] = FE(struc,KE);
12 for ely = 1:nely
13 for elx = 1:nelx
14 n1 = (nely+1)*(elx-1)+ely;
15 n2 = (nely+1)* elx +ely;
16 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);
17 shapeSens(ely,elx) = -max(struc(ely,elx),0.0001)*Ue'*KE*Ue;
18 topSens(ely,elx) = struc(ely,elx)*pi/2*(lambda+2*mu)/mu/(lambda+mu)* ...
19 (4*mu*Ue'*KE*Ue+(lambda-mu)*Ue'*KTr*Ue);
20 end
21 end
22 % Store data, print & plot information
23 objective(iterNum) = -sum(shapeSens(:));
24 volCurr = sum(struc(:))/(nelx*nely);
25 disp([' It.: ' num2str(iterNum) ' Compl.: ' sprintf('%10.4f',objective(iterNum)) ...
26 ' Vol.: ' sprintf('%6.3f',volCurr)])
27 colormap(gray); imagesc(-struc,[-1,0]); axis equal; axis tight; axis off; drawnow;
28 % Check for convergence
29 if iterNum > 5 && ( abs(volCurr-volReq) < 0.005 ) && ...
30 all( abs(objective(end)-objective(end-5:end-1) ) < 0.01*abs(objective(end)) )
31 return;
32 end
33 % Set augmented Lagrangian parameters
34 if iterNum == 1
35 la = -0.01; La = 1000; alpha = 0.9;
36 else
37 la = la - 1/La * (volCurr - volReq); La = alpha * La;
38 end
39 % Include volume sensitivities
40 shapeSens = shapeSens - la + 1/La*(volCurr-volReq);
41 topSens = topSens + pi*(la - 1/La*(volCurr-volReq));
42 % Design update
43 [struc,lsf] = updateStep(lsf,shapeSens,topSens,stepLength,topWeight);
44 % Reinitialize level-set function
45 if ~mod(iterNum,numReinit)
46 [lsf] = reinit(struc);
47 end
48 end
49 %---- REINITIALIZATION OF LEVEL-SET FUNCTION ----
50 function [lsf] = reinit(struc)
51 strucFull = zeros(size(struc)+2); strucFull(2:end-1,2:end-1) = struc;
52 % Use "bwdist" (Image Processing Toolbox)
53 lsf = (~strucFull).*(bwdist(strucFull)-0.5) - strucFull.*(bwdist(strucFull-1)-0.5);
54 %----- DESIGN UPDATE -----
55 function [struc,lsf] = updateStep(lsf,shapeSens,topSens,stepLength,topWeight)
56 % Smooth the sensitivities

```



```

57 - [shapeSens] = conv2(padarray(shapeSens,[1,1],'replicate'),1/6*[0 1 0; 1 2 1;0 1 0],'valid');
58 - [topSens] = conv2(padarray(topSens,[1,1],'replicate'),1/6*[0 1 0; 1 2 1; 0 1 0],'valid');
59 - % Load bearing pixels must remain solid - Bridge:
60 - shapeSens(end,[1,round(end/2):round(end/2+1),end]) = 0;
61 - topSens(end,[1,round(end/2):round(end/2+1),end]) = 0;
62 - % Design update via evolution
63 - [struc,lsf] = evolve(-shapeSens,topSens.*(lsf(2:end-1,2:end-1)<0),lsf,stepLength,topWeight);
64 - %%---- EVOLUTION OF LEVEL-SET FUNCTION ----
65 - function [struc,lsf] = evolve(v,g,lsf,stepLength,w)
66 - % Extend sensitivities using a zero border
67 - vFull = zeros(size(v)+2); vFull(2:end-1,2:end-1) = v;
68 - gFull = zeros(size(g)+2); gFull(2:end-1,2:end-1) = g;
69 - % Choose time step for evolution based on CFL value
70 - dt = 0.1/max(abs(v(:)));
71 - % Evolve for total time stepLength * CFL value:
72 - for i = 1:(10*stepLength)
73 - % Calculate derivatives on the grid
74 - dpx = circshift(lsf,[0,-1])-lsf;
75 - dmx = lsf - circshift(lsf,[0,1]);
76 - dpy = circshift(lsf,[-1,0]) - lsf;
77 - dmy = lsf - circshift(lsf,[1,0]);
78 - % Update level set function using an upwind scheme
79 - lsf = lsf - dt * min(vFull,0).* ...
80 - sqrt( min(dmx,0).^2+max(dpx,0).^2+min(dmy,0).^2+max(dpy,0).^2 ) ...
81 - dt * max(vFull,0) .*...
82 - sqrt( max(dmx,0).^2+min(dpx,0).^2+max(dmy,0).^2+min(dpy,0).^2 )...
83 - w*dt*gFull;
84 - end
85 - % New structure obtained from lsf
86 - strucFull = (lsf<0); struc = strucFull(2:end-1,2:end-1);
87 - %%---- FINITE ELEMENT ANALYSIS ----
88 - function [U] = FE(struc,KE)
89 - [nely,nelx] = size(struc);
90 - K = sparse(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1));
91 - F = sparse(2*(nely+1)*(nelx+1),1); U = zeros(2*(nely+1)*(nelx+1),1);
92 - for elx = 1:nelx
93 - for ely = 1:nely
94 - n1 = (nely+1)*(elx-1)+ely;
95 - n2 = (nely+1)* elx +ely;
96 - edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1; 2*n2+2; 2*n1+1; 2*n1+2];
97 - K(edof,edof) = K(edof,edof) + max(struc(ely,elx),0.0001)*KE;
98 - end
99 - end
100 - % Define loads and supports - Half MBB-beam:
101 - F = sparse(2,1,-1,2*(nely+1)*(nelx+1),1);
102 - fixeddofs = union([1:2:2*(nely+1)], [2*(nelx+1)*(nely+1)]);
103 - alldofs = [1:2*(nely+1)*(nelx+1)];
104 - freedofs = setdiff(alldofs,fixeddofs);
105 - U(freedofs,:) = K(freedofs,freedofs) \ F(freedofs,:);
106 - %%---- MATERIAL INFORMATION ----
107 - function [KE,KTr,lambda,mu] = materialInfo()
108 - % Set material parameters, find Lamé values
109 - E = 1.; nu = 0.3;
110 - lambda = E*nu/((1+nu)*(1-nu)); mu = E/(2*(1+nu));
111 - % Find stiffness matrix "KE"
112 - k = [ 1/2-nu/6 1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 ...
113 - -1/4+nu/12 -1/8-nu/8 nu/6 1/8-3*nu/8];
114 - KE = E/(1-nu^2)*stiffnessMatrix(k);
115 - % Find "trace" matrix "KTr"
116 - k = [1/3 1/4 -1/3 1/4 -1/6 -1/4 1/6 -1/4];
117 - KTr = E/(1-nu)*stiffnessMatrix(k);
118 - %%---- ELEMENT STIFFNESS MATRIX ----
119 - function [K] = stiffnessMatrix(k)
120 - % Forms stiffness matrix from first row
121 - K=[k(1) k(2) k(3) k(4) k(5) k(6) k(7) k(8)
122 - k(2) k(1) k(8) k(7) k(6) k(5) k(4) k(3)
123 - k(3) k(8) k(1) k(6) k(7) k(4) k(5) k(2)
124 - k(4) k(7) k(6) k(1) k(8) k(3) k(2) k(5)
125 - k(5) k(6) k(7) k(8) k(1) k(2) k(3) k(4)
126 - k(6) k(5) k(4) k(3) k(2) k(1) k(8) k(7)
127 - k(7) k(4) k(5) k(2) k(3) k(8) k(1) k(6)
128 - k(8) k(3) k(2) k(5) k(4) k(7) k(6) k(1)];

```

# APÊNDICE D – Algoritmo 2D LS Implementado

```

1  %% OTIMIZAÇÃO TOPOLÓGICA UTILIZANDO O MÉTODO LEVEL SET, VIVIEN J. CHALLIS 2009
2  %%% CODIGO MODIFICADO PARA ESTUDAR LONGARINAS, outubro 2017, POR PAULO LIXANDRAO %%%
3  function [struc] = top_levelset(nelx,nely,volReq,stepLength,numReinit,topWeight)
4  % Inicialização
5  struc = ones(nely,nelx);
6  [lsf] = reinit(struc);
7  shapeSens = zeros(nely,nelx); topSens = zeros(nely,nelx);
8  [KE,KTr,lambda,mu] = materialInfo();
9  % Loop Principal:
10 for iterNum = 1:66
11 % Análise FEA, Cálculo da Sensibilidade
12 [U] = FE(struc,KE);
13 for ely = 1:nely
14 for elx = 1:nelx
15 n1 = (nely+1)*(elx-1)+ely;
16 n2 = (nely+1)* elx +ely;
17 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);
18 shapeSens(ely,elx) = min(struc(ely,elx),0.0001)*Ue'*KE*Ue;
19 topSens(ely,elx) = struc(ely,elx)*pi/2*(lambda+2*mu)/mu/(lambda+mu)* ...
20 (4*mu*Ue'*KE*Ue+(lambda-mu)*Ue'*KTr*Ue);
21 end
22 end
23 % Dados de registro e impressão
24 objective(iterNum) = sum(shapeSens(:));
25 volCurr = sum(struc(:))/(nelx*nely);
26 disp([' It.: ' num2str(iterNum) ' Compl.: ' sprintf('%10.4f',objective(iterNum)) ...
27 ' Vol.: ' sprintf('%6.3f',volCurr)])
28 colormap(gray); imagesc(-struc,[-1,0]); axis equal; axis tight; axis on; drawnow;pause(1e-6);
29 xlabel('nelx'); ylabel('nely'); grid off;
30 % Checagem para Convergência
31 if iterNum > 5 && ( abs(volCurr-volReq) < 0.005 ) && ...
32 all( abs(objective(end)-objective(end-5:end-1) ) < 0.01*abs(objective(end)) )
33 return;
34 end
35 % Dados para os parâmetros do Lagrangeano Aumentado
36 if iterNum == 1
37 la = -0.01; La = 1000; alpha = 0.9;
38 else
39 la = la - 1/La * (volCurr - volReq); La = alpha * La;
40 end
41 % Inclusão de Sensibilidae do Volume
42 shapeSens = shapeSens - la + 1/La*(volCurr-volReq);
43 topSens = topSens + pi*(la - 1/La*(volCurr-volReq));
44 % Atualização do Dominio
45 [struc,lsf] = updateStep(lsf,shapeSens,topSens,stepLength,topWeight);
46 % Reinicialização da Função de Level Set
47 if ~mod(iterNum,numReinit)
48 [lsf] = reinit(struc);
49 end
50 end
51 %----- REINICIALIZAÇÃO DA FUNÇÃO LEVEL SET -----
52 function [lsf] = reinit(struc)
53 strucFull = zeros(size(struc)+2); strucFull(2:end-1,2:end-1) = struc;
54 % Use "bwdist" (Image Processing Toolbox)
55 lsf = (~strucFull).*(bwdist(strucFull)-0.5) - strucFull.*(bwdist(strucFull)-0.5);
56 %----- ATUALIZAÇÃO DO DOMÍNIO -----
57 function [struc,lsf] = updateStep(lsf,shapeSens,topSens,stepLength,topWeight)
58 % Suavidade da sensibilidade
59 [shapeSens] = conv2(padarray(shapeSens,[1,1],'replicate'),1/6*[0 1 0; 1 2 1;0 1 0],'valid');
60 [topSens] = conv2(padarray(topSens,[1,1],'replicate'),1/6*[0 1 0; 1 2 1; 0 1 0],'valid');

```



```

61 % Pixels do Carregamento precisam ser sólidos:
62 shapeSens(end,[1,round(end/2):round(end/2+1),end]) = 0;
63 topSens(end,[1,round(end/2):round(end/2+1),end]) = 0;
64 % Atualização do Domínio via evolução
65 [struc,lsf] = evolve(-shapeSens,topSens.*(lsf(2:end-1,2:end-1)<0),lsf,stepLength,topWeight);
66 %%---- EVOLUÇÃO DA FUNÇÃO LEVEL SET ----
67 function [struc,lsf] = evolve(v,g,lsf,stepLength,w)
68 % Ampliando as sensibilidades utilizando um contorno nulo
69 vFull = zeros(size(v)+2); vFull(2:end-1,2:end-1) = v;
70 gFull = zeros(size(g)+2); gFull(2:end-1,2:end-1) = g;
71 % Escolha do tempo por etapa para evolução baseado no valor CFL
72 dt = 0.1/max(abs(v(:)));
73 % Evolução para um tempo Total do Compimento da Etapa * Valor de CFL:
74 for i = 1:(10*stepLength)
75 % Cálculo das derivadas na tela
76 dpx = circshift(lsf,[0,-1])-lsf;
77 dmx = lsf - circshift(lsf,[0,1]);
78 dpy = circshift(lsf,[-1,0]) - lsf;
79 dmy = lsf - circshift(lsf,[1,0]);
80 % Atualização da função level Set utilizando um esquema Upwind
81 lsf = lsf - dt * min(vFull,0) .* ...
82 sqrt( min(dmx,0).^2+max(dpx,0).^2+min(dmy,0).^2+max(dpy,0).^2 ) ...
83 - dt * max(vFull,0) .*...
84 sqrt( max(dmx,0).^2+min(dpx,0).^2+max(dmy,0).^2+min(dpy,0).^2 ) ...
85 - w*dt*gFull;
86 end
87 % Nova estrutura obtida de LSF
88 strucFull = (lsf<0); struc = strucFull(2:end-1,2:end-1);
89 %%---- ANÁLISE DOS ELEMENTOS FINITOS ----
90 function [U] = FE(struc,KE)
91 [nely,nelx] = size(struc);
92 K = sparse(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1));
93 F = sparse(2*(nely+1)*(nelx+1),1); U = zeros(2*(nely+1)*(nelx+1),1);
94 for elx = 1:nelx
95 for ely = 1:nely
96 n1 = (nely+1)*(elx-1)+ely;
97 n2 = (nely+1)* elx +ely;
98 edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1; 2*n2+2; 2*n1+1; 2*n1+2];
99 K(edof,edof) = K(edof,edof) + max(struc(ely,elx),0.0001)*KE;
100 end
101 end
102 % Definição de Carregamentos e Suportes - Longarina do Chassi:
103 F = sparse(2*(nely+1)*(nelx+1),1);
104 fixeddofs = union(2*(nely+1)*(nsup1:nsup2),2*(nely+1)*(nsup3:nsup4));
105 for elx = 1:nelx+1
106 F(2*(nely+1)*(elx-1)+2,1) = -Ftotal/(nelx+1);
107 end
108 alldofs = [1:2*(nely+1)*(nelx+1)];
109 freedofs = setdiff(alldofs,fixeddofs);
110 U(freedofs,:) = K(freedofs,freedofs) \ F(freedofs,:);
111 %%---- INFORMAÇÃO DO MATERIAL ----
112 function [KE,KTr,lambda,mu] = materialInfo()
113 % Dado dos parâmetros do material.
114 E = 1.; nu = 0.3;
115 lambda = E*nu/((1+nu)*(1-nu)); mu = E/(2*(1+nu));
116 % Encontrando a matriz de rigidez "KE"
117 k = [ 1/2-nu/6 1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 ...
118 -1/4+nu/12 -1/8-nu/8 nu/6 1/8-3*nu/8];
119 KE = E/(1-nu^2)*stiffnessMatrix(k);
120 % Encontrando a matriz traço "KTr"
121 k = [1/3 1/4 -1/3 1/4 -1/6 -1/4 1/6 -1/4];
122 KTr = E/(1-nu)*stiffnessMatrix(k);
123 %%---- MATRIZ DE RIGIDEZ DOS ELEMENTOS----
124 function [K] = stiffnessMatrix(k)
125 % Formato da matriz de rigidez
126 K=[k(1) k(2) k(3) k(4) k(5) k(6) k(7) k(8)
127 k(2) k(1) k(8) k(7) k(6) k(5) k(4) k(3)
128 k(3) k(8) k(1) k(6) k(7) k(4) k(5) k(2)
129 k(4) k(7) k(6) k(1) k(8) k(3) k(2) k(5)
130 k(5) k(6) k(7) k(8) k(1) k(2) k(3) k(4)
131 k(6) k(5) k(4) k(3) k(2) k(1) k(8) k(7)
132 k(7) k(4) k(5) k(2) k(3) k(8) k(1) k(6)
133 k(8) k(3) k(2) k(5) k(4) k(7) k(6) k(1)];

```

# APÊNDICE E – Exemplo de Algoritmo 3D de Liu e Tovar, 2013

```

1  % AN 169 LINE 3D TOPOLOGY OPTIMIZATION CODE BY LIU AND TOVAR (JUL 2013)
2  function top3d(nelx,nely,nelz,volfrac,penal,rmin)
3  % USER-DEFINED LOOP PARAMETERS
4  maxloop = 200; % Maximum number of iterations
5  tolx = 0.01; % Termination criterion
6  displayflag = 0; % Display structure flag
7  % USER-DEFINED MATERIAL PROPERTIES
8  E0 = 1; % Young's modulus of solid material
9  Emin = 1e-9; % Young's modulus of void-like material
10 nu = 0.3; % Poisson's ratio
11 % USER-DEFINED LOAD DOFs
12 il = nelx/2; j1 = 0; k1 = nelz/2;
13 loadnid = k1*(nelx+1)*(nely+1)+il*(nely+1)+(nely+1-j1);
14 load dof = 3*loadnid(:) - 1; % DOFs
15 % USER-DEFINED SUPPORT FIXED DOFs
16 iif = [0 0 nelx nelx]; jf = [0 0 0 0]; kf = [0 nelz 0 nelz];
17 fixednid = kf*(nelx+1)*(nely+1)+iif*(nely+1)+(nely+1-jf);
18 fixeddof = [3*fixednid(:); 3*fixednid(:)-1; 3*fixednid(:)-2]; % DOFs
19 % PREPARE FINITE ELEMENT ANALYSIS
20 nele = nelx*nely*nelz;
21 ndof = 3*(nelx+1)*(nely+1)*(nelz+1);
22 F = sparse(load dof,1,-1,ndof,1);
23 U = zeros(ndof,1);
24 freedofs = setdiff(1:ndof,fixeddof);
25 KE = lk_H8(nu);
26 nodegrd = reshape(1:(nely+1)*(nelx+1),nely+1,nelx+1);
27 nodeids = reshape(nodegrd(1:end-1,1:end-1),nely*nelx,1);
28 nodeidz = 0:(nely+1)*(nelx+1):(nelz-1)*(nely+1)*(nelx+1);
29 nodeids = repmat(nodeids,size(nodeidz))+repmat(nodeidz,size(nodeids));
30 edofVec = 3*nodeids(:)+1;
31 edofMat = repmat(edofVec,1,24)+ ...
32 repmat([0 1 2 3*nely + [3 4 5 0 1 2] -3 -2 -1 ...
33 3*(nely+1)*(nelx+1)+[0 1 2 3*nely + [3 4 5 0 1 2] -3 -2 -1]],nele,1);
34 iK = reshape(kron(edofMat,ones(24,1))',24*24*nele,1);
35 jK = reshape(kron(edofMat,ones(1,24))',24*24*nele,1);
36 % PREPARE FILTER
37 iH = ones(nele*(2*(ceil(rmin)-1)+1)^2,1);
38 jH = ones(size(iH));
39 sH = zeros(size(iH));
40 k = 0;
41 for k1 = 1:nelz
42 for i1 = 1:nelx
43 for j1 = 1:nely
44 e1 = (k1-1)*nelx*nely + (i1-1)*nely+j1;
45 for k2 = max(k1-(ceil(rmin)-1),1):min(k1+(ceil(rmin)-1),nelz)
46 for i2 = max(i1-(ceil(rmin)-1),1):min(i1+(ceil(rmin)-1),nelx)
47 for j2 = max(j1-(ceil(rmin)-1),1):min(j1+(ceil(rmin)-1),nely)
48 e2 = (k2-1)*nelx*nely + (i2-1)*nely+j2;
49 k = k+1;
50 iH(k) = e1;
51 jH(k) = e2;
52 sH(k) = max(0,rmin-sqrt((i1-i2)^2+(j1-j2)^2+(k1-k2)^2));
53 end
54 end

```

```

55 -         end
56 -     end
57 - end
58 - end
59 - H = sparse(iH,jH,sH);
60 - Hs = sum(H,2);
61 - % INITIALIZE ITERATION
62 - x = repmat(volfrac,[nely,nelx,nelz]);
63 - xPhys = x;
64 - loop = 0;
65 - change = 1;
66 - % START ITERATION
67 - while change > tolx && loop < maxloop
68 -     loop = loop+1;
69 -     % FE-ANALYSIS
70 -     sK = reshape(KE(:)*(Emin+xPhys(:)'.^penal*(E0-Emin)),24*24*nele,1);
71 -     K = sparse(iK,jK,sK); K = (K+K')/2;
72 -     U(freedofs,:) = K(freedofs,freedofs)\F(freedofs,:);
73 -     % OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
74 -     ce = reshape(sum((U(edofMat)*KE).*U(edofMat),2),[nely,nelx,nelz]);
75 -     c = sum(sum(sum((Emin+xPhys.^penal*(E0-Emin)).*ce)));
76 -     dc = -penal*(E0-Emin)*xPhys.^(penal-1).*ce;
77 -     dv = ones(nely,nelx,nelz);
78 -     % FILTERING AND MODIFICATION OF SENSITIVITIES
79 -     dc(:) = H*(dc(:)./Hs);
80 -     dv(:) = H*(dv(:)./Hs);
81 -     % OPTIMALITY CRITERIA UPDATE
82 -     l1 = 0; l2 = 1e9; move = 0.2;
83 -     while (l2-l1)/(l1+l2) > 1e-3
84 -         lm1d = 0.5*(l2+l1);
85 -         xnew = max(0,max(x-move,min(1,min(x+move,x.*sqrt(-dc./dv/lm1d)))));
86 -         xPhys(:) = (H*xnew(:))./Hs;
87 -         if sum(xPhys(:)) > volfrac*nele, l1 = lm1d; else l2 = lm1d; end
88 -     end
89 -     change = max(abs(xnew(:)-x(:)));
90 -     x = xnew;
91 -     % PRINT RESULTS
92 -     fprintf(' It.:%5i Obj.:%11.4f Vol.:%7.3f ch.:%7.3f\n',loop,c,mean(xPhys(:)),change);
93 -     % PLOT DENSITIES
94 -     if displayflag, clf; display_3D(xPhys); end %#ok<UNRCH>
95 - end
96 - clf; display_3D(xPhys);
97 - end
98 -
99 -
100 - % === GENERATE ELEMENT STIFFNESS MATRIX ===
101 - function [KE] = lk_H8(nu)
102 - A = [32 6 -8 6 -6 4 3 -6 -10 3 -3 -3 -4 -8;
103 -     -48 0 0 -24 24 0 0 0 12 -12 0 12 12 12];
104 - k = 1/144*A'*[1; nu];
105 -
106 - K1 = [k(1) k(2) k(2) k(3) k(5) k(5);
107 -     k(2) k(1) k(2) k(4) k(6) k(7);
108 -     k(2) k(2) k(1) k(4) k(7) k(6);
109 -     k(3) k(4) k(4) k(1) k(8) k(8);
110 -     k(5) k(6) k(7) k(8) k(1) k(2);
111 -     k(5) k(7) k(6) k(8) k(2) k(1)];
112 - K2 = [k(9) k(8) k(12) k(6) k(4) k(7);
113 -     k(8) k(9) k(12) k(5) k(3) k(5);
114 -     k(10) k(10) k(13) k(7) k(4) k(6);
115 -     k(6) k(5) k(11) k(9) k(2) k(10);
116 -     k(4) k(3) k(5) k(2) k(9) k(12);
117 -     k(11) k(4) k(6) k(12) k(10) k(13)];

```



```

118 - K3 = [k(6) k(7) k(4) k(9) k(12) k(8);
119       k(7) k(6) k(4) k(10) k(13) k(10);
120       k(5) k(5) k(3) k(8) k(12) k(9);
121       k(9) k(10) k(2) k(6) k(11) k(5);
122       k(12) k(13) k(10) k(11) k(6) k(4);
123       k(2) k(12) k(9) k(4) k(5) k(3)];
124 - K4 = [k(14) k(11) k(11) k(13) k(10) k(10);
125       k(11) k(14) k(11) k(12) k(9) k(8);
126       k(11) k(11) k(14) k(12) k(8) k(9);
127       k(13) k(12) k(12) k(14) k(7) k(7);
128       k(10) k(9) k(8) k(7) k(14) k(11);
129       k(10) k(8) k(9) k(7) k(11) k(14)];
130 - K5 = [k(1) k(2) k(8) k(3) k(5) k(4);
131       k(2) k(1) k(8) k(4) k(6) k(11);
132       k(8) k(8) k(1) k(5) k(11) k(6);
133       k(3) k(4) k(5) k(1) k(8) k(2);
134       k(5) k(6) k(11) k(8) k(1) k(8);
135       k(4) k(11) k(6) k(2) k(8) k(1)];
136 - K6 = [k(14) k(11) k(7) k(13) k(10) k(12);
137       k(11) k(14) k(7) k(12) k(9) k(2);
138       k(7) k(7) k(14) k(10) k(2) k(9);
139       k(13) k(12) k(10) k(14) k(7) k(11);
140       k(10) k(9) k(2) k(7) k(14) k(7);
141       k(12) k(2) k(9) k(11) k(7) k(14)];
142 - KE = 1/((nu+1)*(1-2*nu))*...
143       [ K1 K2 K3 K4;
144         K2' K5 K6 K3';
145         K3' K6 K5' K2';
146         K4 K3 K2 K1'];
147 - end
148 % === DISPLAY 3D TOPOLOGY (ISO-VIEW) ===
149 function display_3D(rho)
150 [nely,nelx,nelz] = size(rho);
151 hx = 1; hy = 1; hz = 1; % User-defined unit element size
152 face = [1 2 3 4; 2 6 7 3; 4 3 7 8; 1 5 8 4; 1 2 6 5; 5 6 7 8];
153 set(gcf,'Name','ISO display','NumberTitle','off');
154 for k = 1:nelz
155     z = (k-1)*hz;
156     for i = 1:nelx
157         x = (i-1)*hx;
158         for j = 1:nely
159             y = nely*hy - (j-1)*hy;
160             if (rho(j,i,k) > 0.5) % User-defined display density threshold
161                 vert = [x y z; x y-hx z; x+hx y-hx z; x+hx y z; x y z+hx; x y-hx z+hx;
162                        x+hx y-hx z+hx; x+hx y z+hx];
163                 vert(:,[2 3]) = vert(:,[3 2]); vert(:,2,:) = -vert(:,2,:);
164                 patch('Faces',face,'Vertices',vert,'FaceColor',[0.2+0.8*(1-rho(j,i,k)),
165                        0.2+0.8*(1-rho(j,i,k)),0.2+0.8*(1-rho(j,i,k))]);
166                 hold on;
167             end
168         end
169     end
170 end
171 axis equal; axis tight; axis off; box on; view([30,30]); pause(1e-6);
172 end

```