



UNIVERSIDADE FEDERAL DE MINAS GERAIS
ESCOLA DE ENGENHARIA



**PROGRAMA DE PÓS-GRADUAÇÃO
EM ENGENHARIA DE ESTRUTURAS**

**MÉTODO DAS SOMAS PONDERADAS APLICADO À OTIMIZAÇÃO MULTIOBJETIVO
DE ESTRUTURAS LINEARES E NÃO LINEARES USANDO COMPUTAÇÃO
EVOLUCIONÁRIA**

WALLISTON DOS SANTOS FERNANDES

Orientador: Prof. Marcelo Greco

Tese apresentada ao Programa de Pós-Graduação em Engenharia de Estruturas da Escola de Engenharia da UNIVERSIDADE FEDERAL DE MINAS GERAIS, como requisito parcial para a obtenção do título de DOUTOR EM ENGENHARIA DE ESTRUTURAS.

Belo Horizonte

2016

F363m

Fernandes, Walliston dos Santos.

Método das somas ponderadas aplicado à otimização multiobjetivo de estruturas lineares e não lineares usando computação evolucionária [manuscrito] / Walliston dos Santos Fernandes. - 2016.
xix, 245 f., enc.: il.

Orientador: Marcelo Greco.

Tese (doutorado) Universidade Federal de Minas Gerais,
Escola de Engenharia.

Apêndices: f. 202-245

Bibliografia: f. 189-200.

1. Engenharia de estruturas - Teses. I. Greco, Marcelo. II. Universidade Federal de Minas Gerais. Escola de Engenharia. III. Título.

CDU: 624(043)

UNIVERSIDADE FEDERAL DE MINAS GERAIS
ESCOLA DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE ESTRUTURAS

**MÉTODO DAS SOMAS PONDERADAS APLICADO À OTIMIZAÇÃO
MULTIOBJETIVO
DE ESTRUTURAS LINEARES E NÃO LINEARES USANDO COMPUTAÇÃO
EVOLUCIONÁRIA**

Walliston dos Santos Fernandes

Tese apresentada ao Programa de Pós-Graduação em Engenharia de Estruturas da Escola de Engenharia da UNIVERSIDADE FEDERAL DE MINAS GERAIS, como requisito parcial para a obtenção do título de DOUTOR EM ENGENHARIA DE ESTRUTURAS.

Comissão Examinadora:

Prof. Dr. Marcelo Greco
DEES – UFMG (Orientador)

Prof. Dr. Felício Bruzzi Barros
DEES – UFMG

Prof. Dr. Ricardo Luiz Utsch de Freitas Pinto
DEMEC – UFMG

Prof. Dr. Rodney Rezende Saldanha
DELE – UFMG

Prof. Dr. Edson Denner Leonel
EESC – USP

Belo Horizonte, 12 de julho de 2016

*A Deus, à minha família,
a todos os amigos e, especialmente,
a todos os negros que, salvo exceções,
têm condições de chegar tão longe quanto eu.*

谢谢,大家

魏礼冬

AGRADECIMENTOS

A Deus, por todo o companheirismo, afeto e ajuda necessária para chegar até aqui e por tudo que tem feito, faz e fará para mim pelo resto de minha vida;

Ao Marcelo Greco, que foi mais que um orientador; um verdadeiro pai e amigo e sem ele era impossível eu ter chegado até aqui;

Aos meus pais, Jorge Luís Fernandes e Lourdes Aparecida dos Santos Fernandes (Lurdinha maratonista) por todo o carinho e instrução que me deram para que eu pudesse construir meu caráter e evoluir como ser humano;

Aos meus irmãos Lorena de Fátima Fernandes e William Luiz Fernandes, pela companhia e atenção nos momentos necessários, proporcionando bons momentos de descontração;

À minha namorada e melhor amiga Brenda Fernandes Limoeiro, com quem divido meus momentos de alegrias e vitórias, assim como os de angústias e incertezas, além de estar sempre pronta pra me estender sua mão e me motivar a cada vez mais a seguir em frente;

Aos meus amigos da pós-graduação com quem dividi, divido e espero sempre dividir ótimos momentos, seja com palavras amigas, seja com material de estudo ou com boas e agradáveis risadas, com certeza, fizeram parte dessa conquista;

Ao DEES e à UFMG, por terem me dado essa oportunidade de crescimento profissional e confiado em meu potencial;

A todos os professores da DEES, pelos bons e valiosos momentos de ensino e descontrações;

À CAPES, pelo auxílio financeiro por disponibilizar vários artigos que foram necessários para este estudo através de seu Portal;

E enfim, a todos que direta ou indiretamente contribuíram para que esse grande dia chegasse, meu MUITO OBRIGADO.

*Se você conhece o inimigo e conhece a si mesmo,
não precisa temer o resultado de cem batalhas.
Se você se conhece mas não conhece o inimigo,
para cada vitória ganha sofrerá também uma derrota.
Se você não conhece nem o inimigo nem a si mesmo,
perderá todas as batalhas.*

Sun Tzu

RESUMO

A Otimização Topológica de estruturas no regime estático, linear e não linear físico e geométrico, considerando problemas multiobjetivo é o foco desta Tese. Para a realização desse processo, uma formulação heurística evolucionária denominada SESO (*Smoothing Evolutionary Structural Optimization*) é utilizada em conjunto com o Método dos Elementos Finitos (MEF). Ao se considerar um problema multiobjetivo, é necessário também adotar uma metodologia para a ponderação das alternativas disponíveis a fim de se escolher a topologia ótima. Por isso, o Processo Hierárquico de Análise (AHP) foi abordado, uma vez que a teoria que envolve essa metodologia é capaz de lidar com a avaliação subjetiva de alternativas, segundo múltiplos critérios, de forma simples e hierárquica. Além disso, o Método das Somas Ponderadas (WSM) atua em conjunto com o AHP a fim de organizar numericamente, em termos de critério de decisão, o conjunto de alternativas disponíveis ao longo de toda a análise de otimização. Tudo isso foi implementado em um algoritmo desenvolvido em linguagem Python®, organizado sistematicamente para ser o arquivo de entrada do *software* livre de análise numérica no programa de análise estrutural Code-Aster® que, por sua vez, pertence ao programa de modelagem na interface Salome-Meca® o qual conta com uma interface gráfica amigável para pré e pós-processamento da análise. Finalmente, em paralelo, o trabalho complementa com o desenvolvimento de uma interface computacional no programa Lazarus®, um *software* livre que configura uma opção alternativa ao Borland Delphi®, para as adaptações que forem necessárias para o bom funcionamento do algoritmo SESO, a fim de se realizar as análises pertinentes e avaliar os resultados obtidos através da minimização do vetor gerado pelo WSM. Os nove exemplos lineares demonstram o funcionamento do código, incluindo a comparação de alguns resultados que podem ser encontrados na literatura. Os três exemplos não lineares, além disso, destacam diferenças e/ou semelhanças nas topologias ótimas entre os tipos de análise realizadas nesta Tese.

Palavras chave: *Otimização Multiobjetivo; Smoothing Evolutionary Structural Optimization (SESO); Analytic Hierarchy Process (AHP); Weighted Sum Method (WSM); Salome-Meca.*

ABSTRACT

Topological optimization of structures with static behavior, both linear and physical and geometric nonlinear, considering multi-objective problems is the main of this work. In order to do this process, an evolutionary and heuristic formulation called SESO (Smoothing Evolutionary Structural Optimization) is used together with the Finite Element Method (FEM). When considering a multi-objective problem, it is also necessary to select a methodology according to the alternatives available in order to choose the optimal topology. Therefore, the Analytic Hierarchy Process (AHP) has been used, since all the theory about this method is able to deal with evaluation of subjective alternatives, according to multiple criteria, in a simple and hierarchical manner. Furthermore, the Weighted Sum Method (WSM) works together with the AHP to organize numerically the set of alternatives available throughout the optimization analysis, in terms of decision criteria. During all this process, it was implemented in an algorithm developed in Python[®] programming language that serves as the input file to the free numerical software for multi-physics analysis Code-Aster[®], which belongs to Salome-Meca[®] modeling program, which has a friendly GUI for pre and post processing analyses. Finally, in parallel, the work complements the development of a computational interface in Lazarus, that is a free alternative software comparing with Borland Delphi, for the necessary adjustments for the proper functioning of SESO algorithm in order to perform relevant analyzes and evaluate the results obtained by minimizing the vector that was generated by the WSM. The nine linear examples show the operation code, including the comparison of results that can be found in the literature. The three nonlinear examples show differences and/or similarities between the optimal topologies among the analysis performed in this thesis.

Keywords: *Multi-objective Optimization; Smoothing Evolutionary Structural Optimization (SESO); Analytic Hierarchy Process (AHP); Weighted Sum Method (WSM); Salome-Meca.*

LISTA DE FIGURAS

Figura 1.1 – Estrutura de Michell. Fonte: Silva (2001).....	2
Figura 1.2 – (a) Otimização de Forma e (b) Otimização Topológica. Fonte: Sonmez (2008)....	3
Figura 1.3 – Processo de Otimização Estrutural. Fonte: Labanowski (2004).....	5
Figura 2.1 – Exemplo de uma viga maciça comparada a uma de suas formas otimizadas, a treliça, dado um carregamento qualquer P centrado.....	13
Figura 2.2 – Tipos de Otimização Estrutural: (a) Otimização Paramétrica; (b) Otimização de Forma; (c) Otimização Topológica. Fonte: Bendsøe e Sigmund (2003).....	15
Figura 2.3 – Otimização topológica e topográfica, destacando a espessura de reforço necessária. Fonte: Zhou <i>et al.</i> (2004).....	16
Figura 2.4 – Representação do espaço viável entre funções. Fonte: RAO (2009).....	20
Figura 2.5 – Problema de tabuleiro de xadrez (<i>checkerboard</i>). Fonte: Sigmund e Petersson (1998).....	22
Figura 2.6 – Abrangência do mapeamento de elementos vizinhos do centroide em e . Fonte: Guest e Prévost (2006).....	23
Figura 2.7 – Problema de dependência da malha em duas topologias diferentes geradas por malhas diferentes para um mesmo problema inicial. Fonte: Sigmund e Petersson (1998).....	24
Figura 2.8 – Exemplo de não unicidade de solução, no qual (A) é o domínio inicial e (B) e (C) são duas soluções viáveis. Fonte: Sigmund e Petersson (1998).....	24
Figura 2.9 – Método do Cruzamento Uniforme. Fonte: Adeli & Sarma (2006).....	30
Figura 2.10 – Técnica ESO realizando a retirada do i -ésimo elemento de uma malha triangular qualquer. Fonte: Almeida (2014).....	33
Figura 2.11 – Fluxograma da técnica ESO com Tensão de von Mises como critério de remoção de elementos.....	36
Figura 2.12 – Suavização da quantidade de elementos retirados na iteração i . Fonte: Almeida <i>et al.</i> (2011).....	38
Figura 2.13 – Fluxograma da técnica SESO com Tensão de von Mises como critério de remoção de elementos.....	40
Figura 2.14 – Elemento central i sofrendo influência dos elementos vizinhos dentro de um círculo de raio r . Fonte: Almeida (2014).....	41
Figura 2.15 – Energia de deformação U e energia de deformação complementar U^C	48
Figura 3.1 – Fluxograma de atividades para otimização topológica.....	52
Figura 3.2 – Nó associado a uma malha (2D ou 3D).....	53
Figura 3.3 – Elementos lineares (segmentos) associados a uma malha (2D ou 3D).....	54
Figura 3.4 – Sentido de mapeamento dos elementos lineares.....	54
Figura 3.5 – Elementos triangulares associados a uma malha (2D ou 3D).....	54
Figura 3.6 – Sentido de mapeamento dos elementos triangulares.....	55
Figura 3.7 – Elementos retangulares associados a uma malha (2D ou 3D).....	55
Figura 3.8 – Sentido de mapeamento dos elementos quadrangulares.....	55
Figura 3.9 – Elementos tetraédricos associados a uma malha (3D).....	56
Figura 3.10 – Elementos piramidais associados a uma malha (3D).....	56
Figura 3.11 – Elementos pentaédricos associados a uma malha (3D).....	57
Figura 3.12 – Elementos hexaédricos associados a uma malha (3D).....	58
Figura 3.13 – Método de Newton-Raphson. Fonte: Crisfield (1997).....	62
Figura 3.14 – Modelo elastoplástico perfeito.....	67
Figura 3.15 – Curvas tensão x deformação usadas para caracterizar o encruamento de um material dúctil: (a) encruamento isotrópico; (b) encruamento cinemático. Fonte: Li, Zhang e	

Tong (2014).....	67
Figura 4.1 – Pontos correspondentes ao Ótimo de Pareto e Fronteira de Pareto para duas funções objetivo f_1 e f_2 . Fonte: adaptado de Sehlström (2013).....	73
Figura 4.2 – Fluxograma de três níveis hierárquicos para encontrar a topologia ótima.....	75
Figura 4.3 – Programa desenvolvido em Python para a obtenção dos autovalores e autovetores.....	77
Figura 5.1 – Domínio inicial do problema de duas barras.....	84
Figura 5.2 – Malha de elementos finitos gerada pelo Salome-Meca®.....	84
Figura 5.3 – Topologia parcial resultante do passo de otimização 3.....	85
Figura 5.4 – Topologia parcial resultante do passo de otimização 14.....	85
Figura 5.5 – Topologia parcial resultante do passo de otimização 40 (última iteração).....	86
Figura 5.6 – Topologia final ótima resultante do passo de otimização 24.....	86
Figura 5.7 – Topologias finais ótimas para o problema de duas barras (a) Formulação SESO; (b) Xie e Steven (1993).....	87
Figura 5.8 – Campo de tensões da topologia ótima final para o problema de duas barras.....	87
Figura 5.9 – Diferentes malhas para o problema de duas barras (a) 14362 elementos; (b) 2839 elementos; (c) 1975 elementos.....	88
Figura 5.10 – Parâmetros do Vetor WSM (Volume, Deslocamentos, Tensão, Índice de Performance) x Iterações.....	90
Figura 5.11 – Vetor do WSM x Iterações.....	91
Figura 5.12 – Domínio inicial da Estrutura de Michell.....	92
Figura 5.13 – Malha de elementos finitos gerada pelo Salome-Meca®.....	92
Figura 5.14 – Topologia parcial resultante do passo de otimização 7.....	93
Figura 5.15 – Topologia parcial resultante do passo de otimização 20.....	93
Figura 5.16 – Topologia parcial resultante do passo de otimização 51 (última iteração).....	94
Figura 5.17 – Topologia final ótima resultante do passo de otimização 32.....	94
Figura 5.18 – Topologias finais ótimas para Estrutura de Michell (a) Formulação SESO; (b) Liang (2005).....	95
Figura 5.19 – Parâmetros do Vetor WSM (Volume, Deslocamentos, Tensão, Índice de Performance) x Iterações.....	97
Figura 5.20 – Vetor do WSM x Iterações.....	98
Figura 5.21 – Domínio inicial da placa fina apoiada nas quatro extremidades com carga centrada.....	99
Figura 5.22 – Malha de elementos finitos gerada pelo Salome-Meca®.....	99
Figura 5.23 – Topologia parcial resultante do passo de otimização 6.....	100
Figura 5.24 – Topologia parcial resultante do passo de otimização 20.....	100
Figura 5.25 – Topologia parcial resultante do passo de otimização 42 (última iteração).....	101
Figura 5.26 – Topologia final ótima resultante do passo de otimização 19.....	101
Figura 5.27 – Topologias finais ótimas para chapa fina presa nas quatro extremidades com carga centrada (a) Formulação SESO; (b) Wang (2003).....	102
Figura 5.28 – Parâmetros do Vetor WSM (Volume, Deslocamentos, Tensão, Índice de Performance) x Iterações.....	104
Figura 5.29 – Vetor do WSM x Iterações.....	105
Figura 5.30 – Domínio inicial da placa biapoiada com duas cavidades.....	106
Figura 5.31 – Malha de elementos finitos gerada pelo Salome-Meca®.....	106
Figura 5.32 – Topologia parcial resultante do passo de otimização 5.....	107
Figura 5.33 – Topologia parcial resultante do passo de otimização 12.....	107
Figura 5.34 – Topologia parcial resultante do passo de otimização 27 (última iteração).....	108

Figura 5.35 – Topologia final ótima resultante do passo de otimização 21.....	108
Figura 5.36 – Topologias finais ótimas para chapa biapoiada com duas cavidades (a) Formulação SESO; (b) Liang <i>et al.</i> (2003).....	109
Figura 5.37 – Parâmetros do Vetor WSM (Volume, Deslocamentos, Tensão, Índice de Performance) x Iterações.....	111
Figura 5.38 – Vetor do WSM x Iterações.....	112
Figura 5.39 – Domínio inicial do bloco apoiado nos quatro vértices inferiores.....	113
Figura 5.40 – Malha de elementos finitos gerada pelo Salome-Meca®.....	114
Figura 5.41 – Topologia parcial resultante do passo de otimização 2.....	114
Figura 5.42 – Topologia parcial resultante do passo de otimização 7.....	115
Figura 5.43 – Topologia parcial resultante do passo de otimização 20 (última iteração).....	115
Figura 5.44 – Topologia final ótima resultante do passo de otimização 13.....	116
Figura 5.45 – Parâmetros do Vetor WSM (Volume, Deslocamentos, Tensão, Índice de Performance) x Iterações.....	117
Figura 5.46 – Vetor do WSM x Iterações.....	118
Figura 5.47 – Domínio inicial da torre em formato de “T”.....	119
Figura 5.48 – Malha de elementos finitos gerada pelo Salome-Meca®.....	120
Figura 5.49 – Topologia parcial resultante do passo de otimização 2.....	120
Figura 5.50 – Topologia parcial resultante do passo de otimização 6.....	121
Figura 5.51 – Topologia parcial resultante do passo de otimização 20 (última iteração).....	121
Figura 5.52 – Topologia final ótima resultante do passo de otimização 12.....	122
Figura 5.53 – Parâmetros do Vetor WSM (Volume, Deslocamentos, Tensão, Índice de Performance) x Iterações.....	123
Figura 5.54 – Vetor do WSM x Iterações.....	124
Figura 5.55 – Domínio inicial do bloco com balanços simétricos e furo central.....	125
Figura 5.56 – Malha de elementos finitos gerada pelo Salome-Meca®.....	126
Figura 5.57 – Topologia parcial resultante do passo de otimização 3.....	126
Figura 5.58 – Topologia parcial resultante do passo de otimização 10.....	127
Figura 5.59 – Topologia parcial resultante do passo de otimização 25 (última iteração).....	127
Figura 5.60 – Topologia final ótima resultante do passo de otimização 18.....	128
Figura 5.61 – Parâmetros do Vetor WSM (Volume, Deslocamentos, Tensão, Índice de Performance) x Iterações.....	130
Figura 5.62 – Vetor do WSM x Iterações.....	131
Figura 5.63 – Domínio inicial do quadro de bicicleta. Fonte: Lin e Hsu (2008).....	131
Figura 5.64 – Malha de elementos finitos gerada pelo Salome-Meca®.....	132
Figura 5.65 – Topologia parcial resultante do passo de otimização 6.....	133
Figura 5.66 – Topologia parcial resultante do passo de otimização 11.....	133
Figura 5.67 – Topologia parcial resultante do passo de otimização 36 (última iteração).....	134
Figura 5.68 – Topologia final ótima resultante do passo de otimização 21.....	134
Figura 5.69 – Parâmetros do Vetor WSM (Volume, Deslocamentos, Tensão, Índice de Performance) x Iterações.....	136
Figura 5.70 – Vetor do WSM x Iterações.....	137
Figura 5.71 – Domínio inicial da roda de trem.....	137
Figura 5.72 – Malha de elementos finitos gerada pelo Salome-Meca®.....	138
Figura 5.73 – Topologia parcial resultante do passo de otimização 6.....	139
Figura 5.74 – Topologia parcial resultante do passo de otimização 13.....	139
Figura 5.75 – Topologia parcial resultante do passo de otimização 27 (última iteração).....	140
Figura 5.76 – Topologia final ótima resultante do passo de otimização 21.....	140

Figura 5.77 – Topologia final ótima completa da roda de trem.....	141
Figura 5.78 – Parâmetros do Vetor WSM (Volume, Deslocamentos, Tensão, Índice de Performance) x Iterações.....	143
Figura 5.79 – Vetor do WSM x Iterações.....	144
Figura 6.1 – Diagrama tensão x deformação do material nas análises fisicamente não lineares.....	147
Figura 6.2 – Domínio inicial da chapa apoiada pelas extremidades nos nós centrais.....	148
Figura 6.3 – Malha de elementos finitos gerada pelo Salome-Meca®.....	149
Figura 6.4 – Topologia parcial linear resultante do passo de otimização 5.....	149
Figura 6.5 – Topologia parcial linear resultante do passo de otimização 14.....	150
Figura 6.6 – Topologia parcial linear resultante do passo de otimização 27 (última iteração).....	150
Figura 6.7 – Topologia final ótima linear resultante do passo de otimização 21.....	151
Figura 6.8 – Topologia parcial não linear geométrica resultante do passo de otimização 5...151	151
Figura 6.9 – Topologia parcial não linear geométrica resultante do passo de otimização 11.152	152
Figura 6.10 – Topologia parcial não linear geométrica resultante do passo de otimização 21 (última iteração).....	152
Figura 6.11 – Topologia final ótima não linear geométrica resultante do passo de otimização 18.....	153
Figura 6.12 – Topologia parcial não linear física e geométrica resultante do passo de otimização 6.....	153
Figura 6.13 – Topologia parcial não linear física e geométrica resultante do passo de otimização 15.....	154
Figura 6.14 – Topologia parcial não linear física e geométrica resultante do passo de otimização 28 (última iteração).....	154
Figura 6.15 – Topologia final ótima não linear física e geométrica resultante do passo de otimização 24.....	155
Figura 6.16 – Topologias finais ótimas para o problema da chapa retangular presa pelas extremidades nos nós centrais (a) Formulação SESO, modelo linear; (b) Gea e Luo (2001), modelo linear; (c) Formulação SESO, modelo NLG; (d) Gea e Luo (2001), modelo NLG; (e) Formulação SESO, modelo NLF e NLG.....	155
Figura 6.17 – Parâmetros do Vetor WSM (Volume, Deslocamentos, Tensão, Índice de Performance) x Iterações.....	156
Figura 6.18 – Parâmetros do Vetor WSM (Volume, Deslocamentos, Tensão, Índice de Performance) x Iterações.....	157
Figura 6.19 – Parâmetros do Vetor WSM (Volume, Deslocamentos, Tensão, Índice de Performance) x Iterações.....	158
Figura 6.20 – Falha da convergência da análise não linear física e geométrica.....	159
Figura 6.21 – Comparação dos Vetores do WSM x Iterações.....	159
Figura 6.22 – Domínio inicial da viga em balanço.....	161
Figura 6.23 – Malha de elementos finitos gerada pelo Salome-Meca®.....	162
Figura 6.24 – Topologia parcial linear resultante do passo de otimização 3.....	162
Figura 6.25 – Topologia parcial linear resultante do passo de otimização 10.....	163
Figura 6.26 – Topologia parcial linear resultante do passo de otimização 21 (última iteração).....	163
Figura 6.27 – Topologia final ótima linear resultante do passo de otimização 17.....	164
Figura 6.28 – Topologia parcial não linear geométrica resultante do passo de otimização 3.164	164
Figura 6.29 – Topologia parcial não linear geométrica resultante do passo de otimização 8.165	165

Figura 6.30 – Topologia parcial não linear geométrica resultante do passo de otimização 22 (última iteração).....	165
Figura 6.31 – Topologia final ótima não linear geométrica resultante do passo de otimização 15.....	166
Figura 6.32 – Topologia parcial não linear física e geométrica resultante do passo de otimização 7.....	166
Figura 6.33 – Topologia parcial não linear física e geométrica resultante do passo de otimização 19.....	167
Figura 6.34 – Topologia parcial não linear física e geométrica resultante do passo de otimização 52 (última iteração).....	167
Figura 6.35 – Topologia final ótima não linear física e geométrica resultante do passo de otimização 33.....	168
Figura 6.36 – Parâmetros do Vetor WSM (Volume, Deslocamentos, Tensão, Índice de Performance) x Iterações.....	168
Figura 6.37 – Parâmetros do Vetor WSM (Volume, Deslocamentos, Tensão, Índice de Performance) x Iterações.....	169
Figura 6.38 – Parâmetros do Vetor WSM (Volume, Deslocamentos, Tensão, Índice de Performance) x Iterações.....	170
Figura 6.39 – Comparação dos Vetores do WSM x Iterações.....	171
Figura 6.40 – Domínio inicial da viga em formato de “L”.....	172
Figura 6.41 – Malha de elementos finitos gerada pelo Salome-Meca®.....	173
Figura 6.42 – Topologia parcial linear resultante do passo de otimização 4.....	173
Figura 6.43 – Topologia parcial linear resultante do passo de otimização 10.....	174
Figura 6.44 – Topologia parcial linear resultante do passo de otimização 22 (última iteração).....	174
Figura 6.45 – Topologia final ótima linear resultante do passo de otimização 14.....	175
Figura 6.46 – Topologia parcial não linear geométrica resultante do passo de otimização 4.....	175
Figura 6.47 – Topologia parcial não linear geométrica resultante do passo de otimização 10.....	176
Figura 6.48 – Topologia parcial não linear geométrica resultante do passo de otimização 22 (última iteração).....	176
Figura 6.49 – Topologia final ótima não linear geométrica resultante do passo de otimização 14.....	177
Figura 6.50 – Topologia parcial não linear física e geométrica resultante do passo de otimização 5.....	177
Figura 6.51 – Topologia parcial não linear física e geométrica resultante do passo de otimização 12.....	178
Figura 6.52 – Topologia parcial não linear física e geométrica resultante do passo de otimização 28 (última iteração).....	178
Figura 6.53 – Topologia final ótima não linear física e geométrica resultante do passo de otimização 19.....	179
Figura 6.54 – Parâmetros do Vetor WSM (Volume, Deslocamentos, Tensão, Índice de Performance) x Iterações.....	179
Figura 6.55 – Parâmetros do Vetor WSM (Volume, Deslocamentos, Tensão, Índice de Performance) x Iterações.....	180
Figura 6.56 – Parâmetros do Vetor WSM (Volume, Deslocamentos, Tensão, Índice de Performance) x Iterações.....	181
Figura 6.57 – Comparação dos Vetores do WSM x Iterações.....	182

Figura B.1 – Lazarus [®] , a IDE “open source” compatível com o programa Borland Delphi [®]	211
Figura B.2 – SPE [®] (Stani's Python Editor), um interpretador “open source” Python [®]	214
Figura B.3 – FreeCAD [®] , uma ferramenta “open source” de modelagens paramétricas 3D.....	217
Figura B.4 – Exemplo de arquivo “*.comm” do Code-Aster com as devidas macro-rotinas.....	220
Figura B.5 – Code-Aster [®] , <i>software</i> “open source” de análise estrutural.....	223
Figura B.6 – Definições dos tipos de arquivos de saída definidos pelo usuário no Code-Aster [®]	224
Figura B.7 – Eficas [®] , editor “open source” de arquivos de comando (macro-rotinas).....	227
Figura B.8 – Ambiente JDC, com as respectivas vistas, do programa Eficas [®]	229
Figura B.9 – Representação simbólica das notificações do Eficas [®]	230
Figura B.10 – Salome [®] , plataforma genérica “open source” de pré e pós-processamento.....	232
Figura B.11 – Salome-Meca [®] , programa “open source” completo de análise estrutural via MEF.....	233
Figura B.12 – Módulo de geometria do programa Salome-Meca [®] (Geometry).....	235
Figura B.13 – Módulo de geração de malha do programa Salome-Meca [®] (Mesh).....	236
Figura B.14 – Módulo de cálculo estrutural do programa Salome-Meca [®] (Aster).....	237
Figura B.15 – Módulo de visualização do programa Salome-Meca [®] (ParaViS).....	238
Figura B.16 – Programa UniOM (Units for Optimization Models).....	239
Figura B.17 – Fluxograma de otimização do SESO.....	240
Figura B.18 – Trecho inicial do código implementado em Python [®]	241
Figura B.19 – Primeira parte do código implementado em Python [®]	242
Figura B.20 – Segunda parte do código implementado em Python [®]	243
Figura B.21 – Terceira parte do código implementado em Python [®]	245

LISTA DE TABELAS

Tabela 4.1 – a) Tabela de preferência dos parâmetros (b) Tabela de pontuações relativas.....	75
Tabela 4.2 – Matriz de comparação dos parâmetros.....	76
Tabela 4.3 – Valores do Índice Aleatório (IA) para pequenos problemas. Fonte: LIANG <i>et al.</i> (2000).....	78
Tabela 5.1 – Valores dos parâmetros e do Vetor WSM.....	89
Tabela 5.2 – Valores dos parâmetros e do Vetor WSM.....	96
Tabela 5.3 – Valores dos parâmetros e do Vetor WSM.....	103
Tabela 5.4 – Valores dos parâmetros e do Vetor WSM.....	110
Tabela 5.5 – Valores dos parâmetros e do Vetor WSM.....	116
Tabela 5.6 – Valores dos parâmetros e do Vetor WSM.....	122
Tabela 5.7 – Valores dos parâmetros e do Vetor WSM.....	129
Tabela 5.8 – Valores dos parâmetros e do Vetor WSM.....	135
Tabela 5.9 – Valores dos parâmetros e do Vetor WSM.....	142
Tabela A.1 – Valores dos parâmetros e do Vetor WSM (Exemplo 6.1, linear).....	202
Tabela A.2 – Valores dos parâmetros e do Vetor WSM (Exemplo 6.1, não linear geométrico)	203
Tabela A.3 – Valores dos parâmetros e do Vetor WSM (Exemplo 6.1, não linear físico e geométrico).....	204
Tabela A.4 – Valores dos parâmetros e do Vetor WSM (Exemplo 6.2, linear).....	205
Tabela A.5 – Valores dos parâmetros e do Vetor WSM (Exemplo 6.2, não linear geométrico)	205
Tabela A.6 – Valores dos parâmetros e do Vetor WSM (Exemplo 6.2, não linear físico e	

geométrico).....	206
Tabela A.7 – Valores dos parâmetros e do Vetor WSM (Exemplo 6.3, linear).....	207
Tabela A.8 – Valores dos parâmetros e do Vetor WSM (Exemplo 6.3, não linear geométrico)	207
Tabela A.9 – Valores dos parâmetros e do Vetor WSM (Exemplo 6.3, não linear físico e geométrico).....	208

LISTA DE SÍMBOLOS

E	Módulo de elasticidade longitudinal
E_{ISO}	Módulo de encruamento isotrópico
V_i	Volume do elemento i na i -ésima iteração
V_0	Volume da estrutura na primeira iteração
σ_0^{VM}	Tensão máxima de von Mises na primeira iteração
σ_i^{VM}	Tensão máxima de von Mises do elemento i na i -ésima iteração
$\eta(j)$	Função ponderadora
ν	Coefficiente de Poisson
$\sigma_i^{ponderado}$	Tensão de von Mises ponderada do elemento i
A_i	Área do elemento i
NV	Nº de elementos cujos centróides estão no interior da circunferência
R_{max}	Raio máximo da circunferência
R_{ij}	Distância entre os centróides dos elementos i e j
D_0	Matriz constitutiva inicial do projeto
$D(j)$	Matriz constitutiva no ponto $j \in \Omega$
ρ	Densidade volumétrica
Ω	Domínio da estrutura
$f(\varepsilon)$	Função reguladora
Γ	Conjunto dos elementos que serão removidos da estrutura
$\bar{\Gamma}$	Conjunto dos elementos que não serão removidos da estrutura
$\bar{\sigma}_1, \bar{\sigma}_2, \bar{\sigma}_3$	Tensões principais
σ_x	Componente normal do tensor de tensões na direção x
σ_y	Componente normal do tensor de tensões na direção y
τ_{xy}	Componente de tensão de cisalhamento no plano xy
σ_i	Tensão do elemento i
u	Deslocamento horizontal
v	Deslocamento vertical
$\sigma_{e,MAX}^{VM}$	Tensão de von Mises máxima do elemento e e da estrutura
FRR	Fator de Razão de Rejeição
FER	Fator Evolucionário de Rejeição
V	Volume total da estrutura
VI	Volume da estrutura na iteração i
VRI	Volume retirado por iteração
∇	Gradiente de uma função qualquer
Ψ	Função custo
s^*	Ótimo global
$\tau_i(t)$	Rastro de feromônio depositado na iteração i
CE_i	Contribuição significativa do i -ésimo elemento
$[B]$	Matriz da derivada das funções de forma
$\{F\}$	Vetor de forças externas
$\{u\}$	Vetor de deslocamentos
$[K]$	Matriz de rigidez
γ_j	Número de sensibilidade do elemento j

LISTA DE ABREVIATURAS E SIGLAS

AESO	Additive Evolutionary Structural Optimization
ACO	Ant Colony Optimization
AHP	<i>Analytic Hierarchy Process</i> (Processo Hierárquico de Análise)
BESO	Bidirectional Evolutionary Structural Optimization
ESO	Evolutionary Structural Optimization
IP	Índice de Performance
MEC	Método dos Elementos de Contorno
MEF	Método de Elementos Finitos
NLF	Não Linear/Linearidade Física
NLG	Não Linear/Linearidade Geométrica
OT	Otimização Topológica
SESO	Evolutionary Structural Optimization Smoothing
SIMP	Solid Isotropic Microstructure with Penalization
TSA	Topological Sensitivity Analysis
TVM	Tensão de von Mises
UniOM	Unidades para Modelos de Otimização
WSM	<i>Weighted Sum Method</i> (Método das Somas Ponderadas)
Eq.	Equação
Fig.	Figura

SUMÁRIO

Capítulo 1 – Introdução.....	1
1.1 Linha do Tempo e Abordagem Geral.....	1
1.2 Objetivos.....	7
1.3 Justificativa e Motivação.....	8
1.4 Organização do Trabalho.....	9
Capítulo 2 – Otimização Estrutural.....	11
2.1 Conceitos Iniciais.....	11
2.2 Elementos Comuns para a Otimização Estrutural.....	12
2.3 Tipos de Otimização Estrutural.....	13
2.4 Classificações para a Otimização Estrutural.....	17
2.5 Problemas Numéricos.....	21
2.5.1 Tabuleiro de Xadrez (<i>checkerboard</i>).....	21
2.5.2 Dependência da Malha.....	23
2.5.3 Mínimos Locais.....	25
2.6 Técnicas Usuais de Otimização Estrutural.....	26
2.7 ESO (<i>Evolucionary Structural Optimization</i>).....	32
2.8 SESO (<i>Smoothing Evolucionary Structural Optimization</i>).....	37
2.8.1 Índice de Performance.....	42
2.8.2 Número de Sensibilidade.....	43
2.8.3 <i>Compliance</i>	46
Capítulo 3 – O Método dos Elementos Finitos Aplicado à Otimização Topológica.....	49
3.1 O Método dos Elementos Finitos e a Otimização Estrutural Evolucionária.....	49
3.1.1 Elementos Finitos principais do Code-Aster®.....	53
3.2 Análise não Linear Geométrica e a Otimização Estrutural.....	59

3.3 Análise Não Linear Física e a Otimização Estrutural.....	64
<hr/>	
Capítulo 4 – Metodologia Adotada para a Otimização Multiobjetivo.....	71
4.1 Otimização Multiobjetivo.....	71
4.2 Processo Hierárquico de Análise (AHP) e Método das Somas Ponderadas (WSM).....	74
4.3 Parâmetros de Otimização, Calibração e Metodologia.....	79
<hr/>	
Capítulo 5 – Resultados Lineares.....	83
5.1 Problema de Duas Barras.....	83
5.2 Estrutura de Michell.....	91
5.3 Placa Fina Apoiada nas Quatro Extremidades com Carga Centrada.....	98
5.4 Placa Biapoiada com Duas Cavidades.....	105
5.5 Bloco Apoiado nos Quatro Vértices Inferiores.....	112
5.6 Torre em Formato de “T”.....	118
5.7 Bloco com Balanços Simétricos e Furo Central.....	124
5.8 Quadro de Bicicleta.....	131
5.9 Roda de Trem.....	137
5.10 Discussão dos Resultados.....	144
<hr/>	
Capítulo 6 – Resultados Não-Lineares.....	147
6.1 Chapa Apoiada pelas Extremidades nos Nós Centrais.....	148
6.2 Viga em Balanço.....	160
6.3 Viga em Formato de “L”.....	172
6.4 Discussão dos Resultados.....	183
<hr/>	
Capítulo 7 – Considerações Finais.....	185
<hr/>	
BIBLIOGRAFIA.....	189
<hr/>	
APÊNDICE A.....	201

APÊNDICE B.....	209
------------------------	------------

1

INTRODUÇÃO

Este capítulo se destina a fazer uma síntese dos assuntos abordados ao longo deste trabalho, que envolve otimização em conjunto com o Método dos Elementos Finitos, tanto em análises estáticas lineares quanto não lineares, visando a obtenção da topologia final ótima dos modelos propostos à luz da otimização multiobjetivo.

1.1 Linha do Tempo e Abordagem Geral

Inicialmente, é importante buscar uma definição sobre otimização. Otimizar seria uma maneira de encontrar uma solução adequada, de acordo com um ou mais critérios de custo, para que se possa minimizar ou maximizar determinado critério de projeto. Esse critério pode ser uma melhor relação custo-benefício para se produzir uma peça estrutural, uma significativa redução de peso da estrutura para, por exemplo, minimizar os efeitos causados por ele, uma melhor distribuição de tensões para que membros internos da estrutura não sofram possíveis esforços elevados, minimizar deslocamentos, entre outros.

Seguindo-se a linha do tempo do tema, há relatos de que os primeiros passos da otimização estrutural foram introduzidos no final do século XIX, com o precursor trabalho de Maxwell, em 1872, no qual buscava-se o menor volume para estruturas uniaxiais submetidas a carregamentos (ROZVANY *et al.*, 1995). Em 1904, Michell deu continuidade ao trabalho de Maxwell ao projetar estruturas treliçadas nas quais buscava o menor peso próprio, observando as tensões nas barras para cada caso de carga (BOUCHITTÉ *et al.* 2008). Somente no final dos anos 50, com o trabalho de Cox (1958), seu trabalho foi reconhecido. Suas famosas estruturas são citadas na teoria moderna de Otimização Topológica até hoje. Uma solução

otimizada apresentada no seu trabalho pode ser vista na Fig. 1.1.

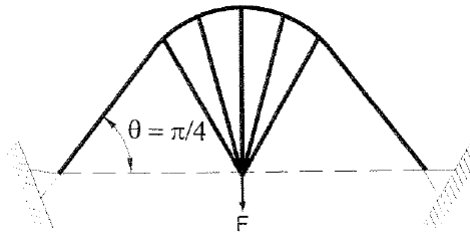


Figura 1.1 – Estrutura de Michell. Fonte: Silva (2001)

Segundo Zhou e Rozvany (1991), na década de 60, os estudos sobre otimização de leiaute avançaram e conseguiram determinar tanto a localização dos membros estruturais como a disposição e seção transversal ótima. Porém, apenas nos anos 70, alguns problemas específicos puderam ser também resolvidos, como os de Hemp (1973) e o de Prager (1974), ambos para uma classe muito restrita de estruturas, vistos como uma extensão do conceito de otimização de estruturas de treliças, desenvolvido por Michell, em 1904.

Na década de 1980, várias publicações foram feitas no que diz respeito à maximização da rigidez de placas delgadas considerando a espessura como variável de projeto (CHENG e OLHOFF, 1981; KHON e STRANG, 1986; ROZVANY *et al.*, 1982), com a utilização do Método dos Elementos Finitos (MEF). As conclusões referentes a esses trabalhos eram que várias soluções ótimas locais podem ser obtidas em problemas de otimização.

Ainda em meados da mesma década de 1980, vários questionamentos sobre os resultados das otimizações de forma e paramétricas surgiram, pois alguns deles apresentavam problemas quando se alterava o modelo. Por isso, com a necessidade de se aprimorar a otimização de forma, surge no final da década a Otimização Topológica (OT), com uma metodologia de domínio fixo estendido, inicialmente no método da Homogeneização (BENDSØE e KIKUCHI, 1988). Essa metodologia foi inspirada nos trabalhos de Cheng e Olhoff (1981) e de Cheng e Olhoff (1982) que tratavam da otimização de espessuras de chapas e placas, Lurie *et al.* (1982), Goodman *et al.* (1986), Kohn e Strang (1986) que estudaram a otimização para projetos de barras de torção construídas com dois materiais com diferentes proporções volumétricas e Rozvany *et al.* (1982), que investigaram a formulação matemática para o problema de maximização da rigidez (com restrição de volume) de placas delgadas, onde a

variável de projeto é a espessura e concluíram que para este problema de otimização existem vários ótimos locais.

As diferenças entre a otimização de forma e a topológica são mostradas na Fig. 1.2.

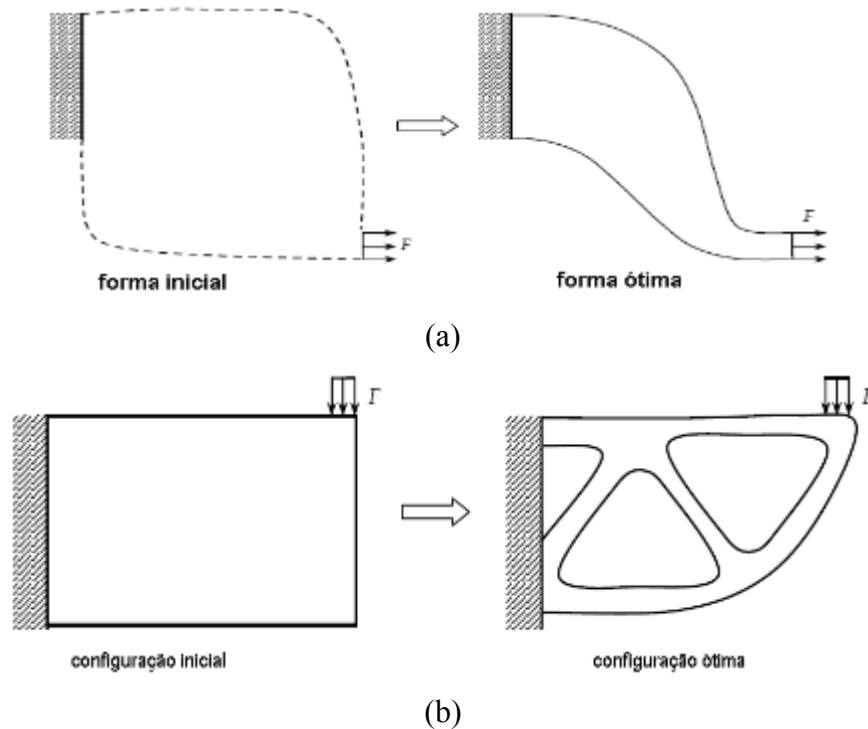


Figura 1.2 – (a) Otimização de Forma e (b) Otimização Topológica. Fonte: Sonmez (2008)

A principal diferença entre estes tipos de otimização está no fato de que na otimização de forma não há a formação de cavidades, ou seja, há apenas uma adaptação do domínio para o novo modelo. Tendo em vista essa diferença, nasce então uma classificação da otimização quanto ao tipo: Otimização de Forma e Otimização Topológica. Além destes dois tipos, há ainda a Otimização Paramétrica e a Otimização Topográfica.

A partir dessas informações, vários trabalhos sobre otimização foram feitos e melhorados de várias formas, seguindo os conceitos básicos que dão suporte teórico que foram estabelecidos há mais de um século (ROZVANY *et al.*, 1995), porém, em sua maioria, seguindo uma análise linear dos modelos propostos. Apenas após o ano 2000, alguns trabalhos sobre não linearidade e otimização começaram a aparecer com mais frequência, como visto em Gea e Luo (2001), Pereira (2006), Lahuerta *et al.* (2010) e Chang *et al.* (2012).

A grande vantagem da Otimização Topológica em contraste com os métodos tradicionais de otimização, como a Otimização de Forma ou a Otimização Paramétrica, é que esses métodos não são capazes de alterar o leiaute da estrutura original, fazendo com que esses não auxiliem o projeto conceitual da estrutura.

Pode-se dizer que na Otimização Paramétrica, a estrutura apresenta a forma e a topologia fixa, ou seja, variam-se as dimensões, tais como o diâmetro da seção transversal de seus componentes, espessura e altura, por exemplo. Alguns trabalhos sobre este tipo de otimização podem ser listados: linear (ANDERSON, 1984), parcialmente linear (HECKMAN, 1986), não-linear por partes (ROBINSON, 1988), aditivo (HASTIE e TIBSHIRANI, 1986), expansão unidirecional (HÄRDLE e STOKER, 1989), redução de dimensionalidade (LI, 1991), expansão unidirecional aditiva (FRIEDMAN e STUETZLE, 1981).

Na otimização de forma, um pouco mais genérica, a estrutura apresenta forma variável e a topologia fixa, não ocorrendo a inserção de cavidades. Nesta abordagem, as variáveis de projeto definem o contorno do domínio, cuja forma é alterada durante o processo de otimização sobre esse procedimento (HAFTKA e GRANDHI, 1986). Em outras palavras, esse tipo de otimização pode ser descrito como aquele onde a meta é maximizar (ou minimizar) a função objetivo, valendo-se do contorno dos segmentos e a posição de furos, já fixos. A escolha das variáveis de projeto adequadas para a parametrização das formas é um dos problemas deste método (PARKINSON *et al.* 1997).

A otimização topográfica é uma particularidade da otimização de forma, onde enrijecedores são projetados para contribuir para aumentar a rigidez de estruturas de placas e cascas, principalmente. Assim, esse tipo de otimização preocupa-se em obter o melhor formato para reforçadores de acordo com as condições inicialmente impostas (SCHNEIDER e ERNEY, 2000; ZHOU *et al.* 2004).

Diferentemente da otimização de forma, na qual as variáveis de projeto que definem o contorno são alteradas em cada iteração durante o processo de otimização, a Otimização Topológica apresenta como principais características a inserção de buracos e domínio fixo

estendido, com as dimensões do projeto mantidas fixas durante todo o processo iterativo (BENDSØE e KIKUCHI, 1988). Na procura pela solução ótima, a Otimização Topológica distribui o material por todo o domínio de tal maneira que se possa otimizar um determinado critério, como o de tensão máxima da estrutura. Esse tipo de otimização pode ser dividido em duas classes distintas: a partir de um meio discreto e a partir de um meio contínuo. A primeira é subdividida em dois grupos: otimização de estruturas de malha contínua (“gridlike continua”) que utiliza um número infinito de barras rígidas separadas por um espaço infinitesimal cuja solução é obtida analiticamente e a otimização de estruturas discretas (“ground structural”) que considera o domínio de projeto com vários pontos distribuídos que podem ser uma possível ligação de treliças, cuja solução é obtida numericamente.

De modo geral, as etapas envolvidas em um processo de otimização estrutural podem ser descritas de acordo com a Fig. 1.3.

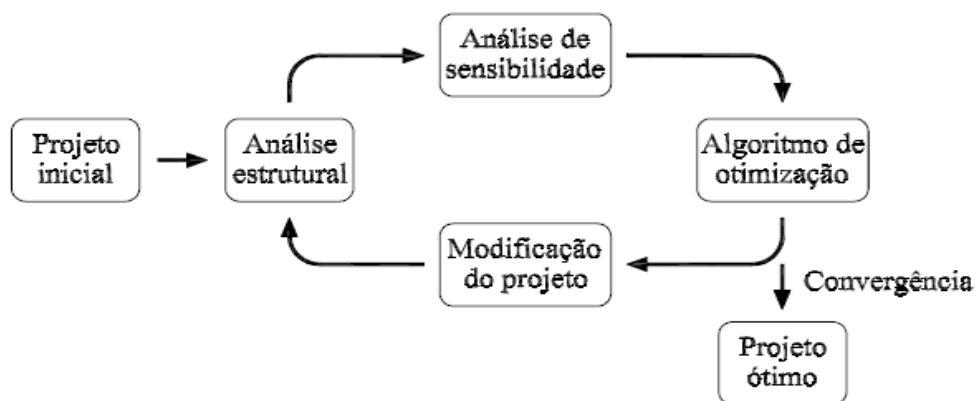


Figura 1.3 – Processo de Otimização Estrutural. Fonte: Labanowski (2004)

Assim, dado um modelo inicial de projeto, faz-se uma análise estrutural, de acordo com as condições de contorno e verifica-se sua sensibilidade, que representa o comportamento da estrutura face às condições impostas inicialmente. Depois, o algoritmo de otimização é aplicado e, caso haja a convergência, tem-se a topologia ótima final; do contrário, o projeto é modificado e é então refeita a análise estrutural. A ocorrência dessa convergência pode depender de vários fatores, o que, na prática, aumenta as incertezas quanto à segurança da resposta obtida pelo processo de otimização. Segundo Zhao *et al.* (2015), como as incertezas são inevitavelmente observadas em aplicações do mundo real devido a erros de fabricação e

ambiente variável, por exemplo, topologias inviáveis podem ser obtidas dessa maneira. Portanto, uma minuciosa análise deve ser levada em consideração para a avaliação das topologias ótimas antes de serem usadas em projetos estruturais.

Mas a otimização por si só não é suficiente para analisar um modelo, ou seja, ela trabalha em conjunto com métodos de análise estrutural para garantir os resultados pertinentes. Naturalmente, além dos vários trabalhos sobre Otimização Topológica envolvendo o Método dos Elementos Finitos que são descritos ao longo deste capítulo, pesquisas envolvendo o Método dos Elementos de Contorno (CERVERA, 2003; CERVERA e TREVELYAN, 2005; MARCZAK, 2006) também podem ser encontradas na literatura.

A grande vantagem da utilização do Método dos Elementos Finitos em conjunto com otimização estrutural é a fácil implementação computacional de algoritmos que permitam realizar as análises, além de vários *softwares* disponíveis (comerciais e livres) para vários sistemas operacionais. Um desses programas de análise estrutural que merece destaque é o Code-Aster[®], que possibilita a realização de vários tipos de análise via Método dos Elementos Finitos, como estática linear e não linear, dinâmica (com ou sem amortecimento, sendo linear ou não linear), fratura, dano, fadiga e interações solo-estrutura, fluido-estrutura e solo-fluido-estrutura. Além disso, o programa é flexível e de código aberto e gratuito, permitindo modificações e/ou implementações diretas em sua biblioteca de comandos. Uma outra vantagem é a possibilidade de se utilizar *scripts* implementados na linguagem Python[®] para realizar alterações na análise sem a necessidade de modificar diretamente o código do *software*.

Este trabalho propõe a implementação de um algoritmo de otimização em forma de *script* Python[®] para a análise de estruturas estáticas, de forma linear e não linear física e geométrica. A técnica implementada será a oriunda de uma formulação baseada no método heurístico clássico evolucionário ESO (*evolutionary structural optimization*), denominada SESO – Suavização ou Smoothing ESO, trabalhando em conjunto com o Método dos Elementos Finitos presente no *software* Code-Aster[®] para a apuração dos resultados.

1.2 Objetivos

O objetivo principal deste trabalho é obter e analisar topologias ótimas a partir de análises estáticas lineares e não lineares considerando os conceitos de otimização multiobjetivo para a técnica heurística de otimização SESO (Smoothing Evolutionary Structural Optimization). Os resultados lineares obtidos serão comparados com a literatura a fim de validar todo o procedimento de otimização. Os resultados não lineares serão comparados entre si, investigando as diferenças e/ou semelhanças entre os diferentes tipos de análise (linear, não linear física e não linear geométrica). Para soluções não lineares disponíveis na literatura, os resultados também serão comparados com as mesmas. Essas comparações feitas com os resultados da literatura envolvem otimização topológica e elementos finitos, tanto em questão das técnicas de otimização utilizadas quanto em relação aos modelos discretizados, a fim de mostrar a utilidade da técnica de otimização SESO (*Smoothing Evolutionary Structural Optimization*) tanto quanto sua precisão em obter os resultados, proporcionalmente à sua facilidade em relação a implementação numérica.

Os objetivos específicos deste trabalho são:

→ Desenvolver um *script* na linguagem Python[®] que, associado ao programa Salome-Meca[®] (versão 2014.1 e que inclui o programa de elementos finitos Code-Aster[®]), seja capaz de otimizar problemas estruturais estáticos, modificando sua topologia inicial de acordo com um critério de remoção de elementos. Esse *script*, na verdade, será a implementação de um algoritmo adaptável a vários tipos de análise (estática linear e não linear física e geométrica), que é diretamente inserido no arquivo de entrada do *software* Code-Aster[®] para a realização das análises.

→ Desenvolver um programa em Lazarus[®] capaz de gerar um arquivo em formato “.py” (Python[®]) com a finalidade de se obter os pesos necessários para a otimização multiobjetivo. Além disso, o programa deve ser capaz de realizar as modificações necessárias no arquivo de entrada do Code-Aster[®], de acordo com o tipo de análise definida pelo usuário. O programa possui uma interface gráfica simples, gerando apenas o arquivo de entrada necessário já pronto para ser usado na análise do modelo e/ou o arquivo de entrada para um interpretador Python[®].

1.3 Justificativa e Motivação

Produzir uma peça estrutural econômica, funcional e segura nos dias atuais têm sido um grande desafio em várias áreas da engenharia. Por isso, a otimização estrutural torna-se um assunto bastante relevante a ser considerado e explorado, ressaltando a importância da realização deste trabalho que envolve este assunto bastante pertinente.

Além disso, ainda é muito importante citar a pesquisa e utilização de novas plataformas de código livre, amplamente validados e gratuitos para análise de estruturas, uma vez que as licenças de *softwares* comerciais possuem custo elevado, mesmo em âmbito acadêmico. Assim, uma grande vantagem deste trabalho é a utilização de programas pouco conhecidos (ou até mesmo totalmente desconhecidos) no meio acadêmico e/ou empresarial, abrindo portas para novas opções de análise e pesquisa.

Dada a relevância da otimização estrutural de forma geral, tanto do ponto de vista prático quanto no meio acadêmico, a importância desta Tese baseia-se na obtenção e avaliação da topologia ótima face a múltiplos objetivos e diferentes condições iniciais. Ao se considerar técnicas heurísticas de otimização, principalmente aquelas em que são necessários passos de otimização ou iterações (como ESO, SESO, BESO, entre outras), a garantia da topologia ótima obtida pela técnica passa a ser uma incógnita, uma vez que critérios de parada são necessários, assim como a calibração adequada dos parâmetros de remoção. Por essa razão, a partir de n passos de otimização e, conseqüentemente, n topologias parciais obtidas, o ótimo pode ser obtido a partir do leque de respostas parciais sem necessariamente ser a iteração com o menor volume entre todas as opções (considerando a minimização de volume o fator principal para a otimização das técnicas mencionadas acima), tornando o estudo, de acordo com o autor, relevante e necessário.

Em relação a não linearidade, é proposto nesta Tese uma alternativa ao Compliance para a obtenção da topologia ótima a partir de técnicas evolucionárias de otimização. A formulação, além de possuir uma implementação mais simples (quando comparada com formulações baseadas em gradientes), possibilita a obtenção de soluções nos domínios onde o Compliance não é definido.

1.4 Organização do Trabalho

O Capítulo 2, a seguir, destina-se a tratar especificamente da otimização estrutural, envolvendo conceitos, técnicas e métodos usados e descritos pela literatura na análise de um modelo. Uma síntese sobre as técnicas de otimização mais usadas pela literatura, incluindo as respectivas particularidades, serão brevemente descritas, com um foco maior na técnica usada nesta Tese.

O Capítulo 3 destina-se a tratar de uma abordagem do Método dos Elementos Finitos, utilizado na resolução dos problemas propostos, em conjunto com métodos clássicos para análises não lineares de forma física e geométrica. Todos os assuntos são diretamente relacionados com a otimização estrutural, enfatizando suas utilizações na literatura.

O Capítulo 4 destina-se a tratar da metodologia aqui adotada para a obtenção da topologia ótima. Uma abordagem sobre Otimização Multiobjetivo, do Processo Hierárquico de Análise (AHP) e do Método das Somas Ponderadas (WSM) preenche o capítulo, além de uma abordagem sobre os parâmetros necessários para a execução do código implementado para as análises desta Tese.

O Capítulo 5 destina-se a tratar especificamente de sete resultados obtidos pela implementação do código de otimização, sendo estes analisados em regime estático linear. Os modelos retirados da literatura foram comparados para comprovar a eficiência da metodologia aqui desenvolvida, assim como incluídos exemplos novos para enriquecer o conteúdo da Tese. Ao fim do capítulo, uma breve discussão dos resultados é apresentada.

O Capítulo 6 destina-se a tratar especificamente de três resultados obtidos pela implementação do código de otimização, sendo estes analisados em regime estático não linear físico e geométrico. Assim como no capítulo 5, os modelos retirados da literatura foram comparados para comprovar a eficiência da metodologia aqui desenvolvida, assim como incluídos exemplos novos. As topologias ótimas para os regimes linear, não linear geométrico e não linear físico e geométrico foram comparadas entre si para cada exemplo, a fim de se verificar as diferenças e/ou semelhanças entre as análises, ressaltando a importância de uma

abordagem adequada para os problemas propostos. Uma breve discussão dos resultados encerra o capítulo.

O Capítulo 7 apresenta as considerações finais a respeito dos resultados obtidos nos capítulos 5 e 6, além da conclusão sobre a utilização da otimização multiobjetivo em conjunto com o Método dos Elementos Finitos.

A bibliografia encerra a Tese.

2

OTIMIZAÇÃO ESTRUTURAL

Este capítulo se destina a tratar especificamente da otimização estrutural, envolvendo conceitos, técnicas e métodos usualmente utilizados e descritos pela literatura na análise de um modelo.

2.1 Conceitos Iniciais

Em termos gerais, otimizar pode ser definido como o processo de obtenção da melhor alternativa, de acordo com o problema considerado, que é pertencente a um espaço de soluções viáveis. Um espaço viável é definido pelo conjunto de todas as possíveis soluções para um problema de otimização que satisfazem a todas as restrições impostas, quer sejam de igualdade e/ou desigualdade. Assim, um problema de otimização consiste em minimizar ou maximizar uma função, denominada objetivo, a fim de se obter para esta mesma função a melhor solução, dado um conjunto de restrições quaisquer.

Na área da engenharia, a otimização é um campo de pesquisa que está se expandindo rapidamente e que desempenha um papel cada vez mais amplo na maioria dos ramos relacionados com a tecnologia, como o aeroespacial, da engenharia mecânica, naval, civil e de materiais. Possui alta exigência na área de métodos computacionais por estar diretamente envolvida em teorias complexas da matemática, lidando com alguns dos problemas mais difíceis com relação a área de ciências mecânicas. Entretanto é de considerável interesse, sendo realizada na maioria dos países desenvolvidos do mundo (ROZVANY e OLHOFF, 2000).

2.2 Elementos Comuns para a Otimização Estrutural

Segundo Arora (2012) e Haftka e Gürdal (1992), na otimização estrutural, as variáveis de projeto são parâmetros que expressam a possibilidade de mudança ou alteração de um modelo estrutural inicial, podendo controlar as dimensões da seção transversal da peça, as propriedades do material, a geometria da estrutura, entre outros e são geralmente expressos na forma de vetor $x = (x_1, x_2, \dots, x_n)$. Além disso, podem ser de natureza discreta (assumem valores isolados dentro de um conjunto na forma $\{x \in \mathbb{R} \mid x = (k_1, k_2, \dots, k_n)\}$) ou contínua (assumem qualquer valor dentro de um conjunto na forma $\{x \in \mathbb{R} \mid k_{\text{inf}} \leq x \leq k_{\text{sup}}\}$). É importante ressaltar que a escolha das variáveis de projeto pode ser decisiva para o sucesso do processo de otimização, devendo ser consistentes com o modelo em análise.

Por outro lado, têm-se as funções objetivo, que servem para medir a eficácia do projeto. São expressas por $f(x)$, para o caso de uma função objetivo, ou $F(x) = [f_1(x), f_2(x), \dots, f_p(x)]$, para o caso de mais de uma função objetivo. Peso, deslocamentos, tensão, frequência de vibração, custo ou qualquer combinação destes eventos podem ser utilizados como função objetivo na otimização estrutural.

Por último, e não menos importante, há as restrições. São elas que impõem limites e/ou valores às variáveis de projeto. São divididas em: restrições laterais, que impõem limites superiores e inferiores dentro de um intervalo (valores máximos e mínimos admissíveis, por exemplo) e restrições de igualdade, que determinam valores precisos (equações de equilíbrio da estática, por exemplo) e são representados por $h_j(x) = h_j(x_1, x_2, \dots, x_n) = 0$, com j variando de 1 até o número de total de restrições de igualdade; e restrições de desigualdade, que impõem limites superiores ou inferiores de forma quantitativa (a estrutura não pode atingir o estado limite de utilização, ou seja, a tensão em todos os pontos da estrutura deve ficar abaixo da máxima tensão admissível, por exemplo). Diversos problemas de otimização de natureza não linear não aceitam restrições de igualdade, limitando-se apenas às restrições de desigualdade (ARORA, 2012; HAFTKA e GÜRDAL, 1992).

É fundamental que haja dependência explícita ou implícita das funções objetivo para com as variáveis de projeto. As funções que não dependerem de nenhuma variável não possuem

relação com o problema e podem ser descartadas. Além disso, o número de restrições de igualdade deve ser menor ou igual ao número de variáveis de projeto, para que haja um sistema indeterminado de equações, causando ou uma dependência linear entre as restrições ou uma inconsistência. Já para as restrições de desigualdade, o número de restrições ativas simultaneamente, isto é, que tiverem seu critério atendido, precisa ser também menor ou igual ao número de variáveis de projeto. Podem existir problemas sem restrições.

2.3 Tipos de Otimização Estrutural

Na engenharia, alguns elementos estruturais são clássicos e indispensáveis nas construções que existem no mundo. Pilares, vigas e treliças são exemplos de elementos estruturais para se construir um edifício. Fazendo assim um destaque especial às treliças, é importante ressaltar que, dentre várias particularidades, para determinados tipos de carregamentos elas se comportam cinematicamente de maneira consistente com a concepção do elemento, além de serem econômicas do ponto de vista do peso próprio total, se comparadas a uma estrutura similar maciça (Fig. 2.1).

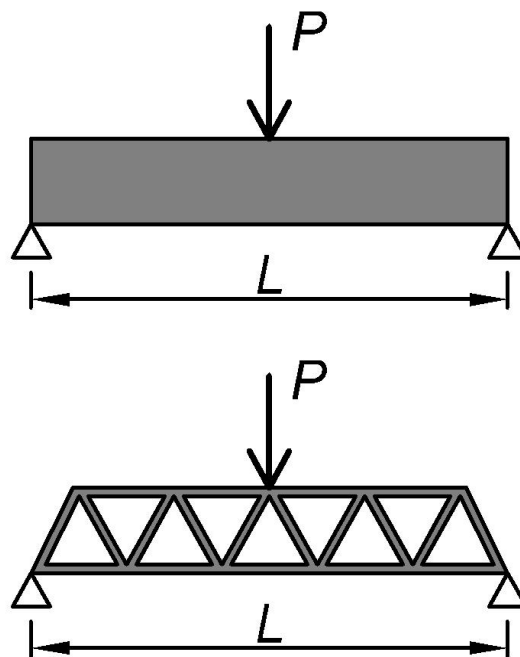


Figura 2.1 – Exemplo de uma viga maciça comparada a uma de suas formas otimizadas, a treliça, dado um carregamento qualquer P centrado

Na Fig 2.1 acima, ao se comparar ambas estruturas supostamente feitas com um mesmo material qualquer M , possuindo um mesmo comprimento L , uma mesma altura h , uma mesma espessura t e submetidas a uma mesma carga P , percebe-se que a treliça possui menor volume final do que a viga maciça, o que torna a treliça um modelo otimizado em relação à viga maciça. Assim, para este mesmo exemplo, um dos critérios de otimização pode ser definido pela busca pelo menor peso, que é obtido no modelo final (treliça) em relação ao modelo inicial (viga maciça). Outros exemplos de critérios de otimização estrutural para diferentes problemas seriam a busca por menores deslocamentos e/ou frequência do projeto sem ferir as condições de equilíbrio e compatibilidade da peça, ou mesmo uma melhor distribuição de tensões ao longo da estrutura, descartando-se regiões do domínio onde não haja fluxo de tensões ou este seja de muito baixo valor, ou ainda a relação custo-benefício para a produção em série daquela estrutura.

Ainda se considerando o mesmo exemplo apresentado na Fig. 2.1, verifica-se que, do ponto de vista da otimização estrutural, cavidades foram formadas no interior da viga maciça até o ponto em que a estrutura de treliça fosse determinada. Este atributo é destacado em um tipo de otimização denominada topológica, cujas principais características são a inserção de cavidades e domínio fixo estendido, ou seja, as dimensões do projeto são mantidas fixas durante todo o processo iterativo (BENDSØE e KIKUCHI, 1988). Mais dois outros tipos clássicos de otimização são encontrados na literatura, sendo a otimização paramétrica, onde a estrutura apresenta forma e topologia ambas fixas, e variam-se apenas as características constitutivas do material e/ou as dimensões da peça estrutural (VANDERPLAATS, 1984) e a otimização de forma, onde a estrutura apresenta a topologia fixa, variando-se a forma e não ocorrendo a inserção de cavidades. Na otimização de forma, as variáveis de projeto definem o contorno do domínio, cuja forma é alterada durante o processo de otimização (HAFTKA e GRANDHI, 1986). Em outras palavras, tanto a otimização de forma quanto a paramétrica e a topológica abordam, cada uma, diferentes aspectos do problema estrutural em si, de acordo com a Fig. 2.2.

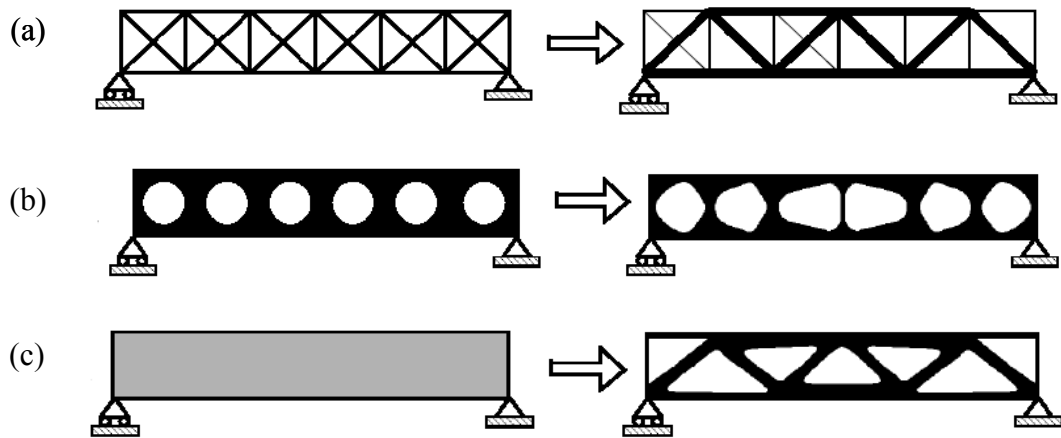


Figura 2.2 – Tipos de Otimização Estrutural: (a) Otimização Paramétrica; (b) Otimização de Forma; (c) Otimização Topológica. Fonte: Bendsøe e Sigmund (2003)

Em um típico problema de otimização paramétrica, por exemplo, o objetivo pode ser o de encontrar a espessura ótima de uma placa no regime elástico linear as áreas ótimas da estrutura de treliça. Assim, a espessura ideal minimiza ou maximiza uma função quantitativa, relativa à tensão máxima ou à deflexão em um intervalo qualquer, por exemplo, enquanto o equilíbrio e outras restrições sobre as variáveis de estado e projeto estão satisfeitos. Logo, a principal característica de um problema de otimização paramétrica é que o domínio do projeto e as variáveis de estado são conhecidos e fixos ao longo do processo de obtenção da estrutura final, ou de otimização, podendo ser alterados os parâmetros do material em análise (BENDSØE e SIGMUND, 2003).

Por outro lado, em um problema de otimização de forma, o objetivo é encontrar a configuração ótima do domínio. A forma é definida em um domínio que agora é a variável de projeto, ou seja, de forma geral, a ideia é procurar as formas de geometria que minimizam (ou maximizam) determinada função objetivo, como a dissipação de energia ou massa e/ou as restrições geométricas. Este tipo de problema geralmente implica grande custo computacional, que engloba aproximações numéricas (e otimização) e exige uma abordagem adequada para a representação da forma ótima do modelo, assim como o cálculo do gradiente da função objetivo a ser minimizado (ou maximizado). Assim, a variável de controle é a forma do domínio e o objetivo é minimizar (ou maximizar) o custo da função objetivo funcional para finalmente a forma ideal ser obtida (YAN e GAO, 2013). Em paralelo, uma

vertente da otimização de forma seria a otimização topográfica, cujo objetivo é definir padrões de reforço estrutural em uma determinada região do modelo considerado a partir de uma variação de sua forma. É utilizada especialmente em estruturas de placas e cascas, com a inclusão de padrões de reforço estrutural. Sua abordagem é semelhante à abordagem utilizada na otimização topológica (Fig. 2.3).

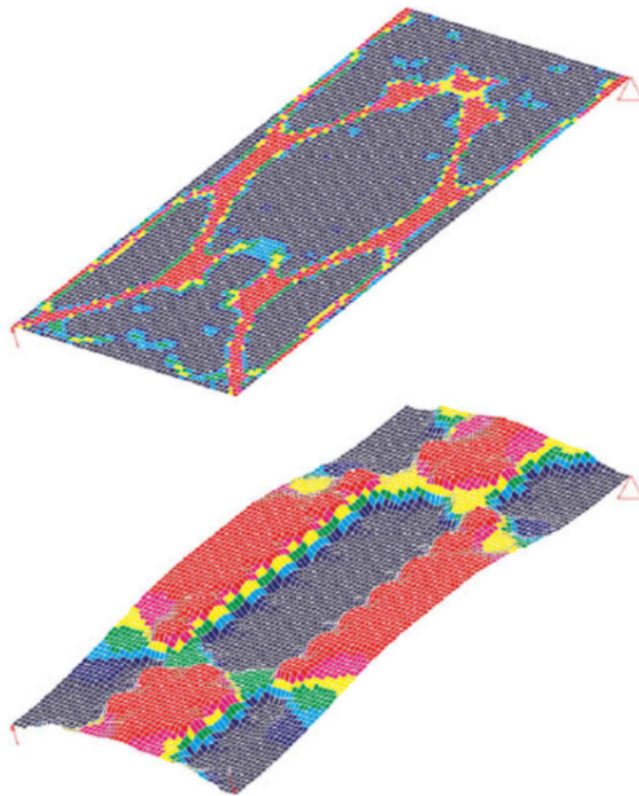


Figura 2.3 – Otimização topológica e topográfica, destacando a espessura de reforço necessária. Fonte: Zhou *et al.* (2004)

Já a otimização topológica busca determinar características como a quantidade, o local e a forma dos vazios formados, assim como a conectividade do domínio. A importância deste tipo de otimização está no fato de que a escolha da topologia adequada de uma estrutura é geralmente o fator mais importante para uma boa relação custo-benefício. Além disso, a otimização paramétrica e de forma não podem alterar a topologia da estrutura durante o processo de obtenção da solução, de modo que uma possível resposta obtida por um destes tipos de otimização possuirá a mesma topologia em relação à concepção inicial. Isso torna a otimização topológica mais valiosa para a obtenção de concepções prévias em relação aos outros dois tipos (KROG e OLHOFF, 1999).

De fato, as respostas intermediárias obtidas pela otimização topológica são de grande valia para se ter uma ideia de como será a concepção estrutural ótima. Muitas vezes a convergência do problema não ocorre de forma rápida e simples, demandando grande tempo de processamento para a obtenção da topologia final. Juntamente a isso, deve-se considerar o fato de que algumas vezes uma determinada topologia resultante de um processo de otimização não possa ser considerada como ótima, levando-se em conta certas condições preestabelecidas no projeto inicial como, por exemplo, o surgimento de um furo em um local inadequado do domínio, ao longo do processo de otimização, de acordo com a finalidade da estrutura ou o aparecimento de algum membro interno, em um determinado instante, não concebível do ponto de vista estrutural. Por isso, ao se avaliar um modelo verificando cada resposta obtida em passo de otimização é possível estimar como será a topologia ótima final e, assim, fazer ou não alterações na metodologia de análise de acordo com as especificações iniciais de projeto, tornando a otimização topológica muito vantajosa neste aspecto.

2.4 Classificações para a Otimização Topológica

Ao se tratar de Otimização Estrutural, é importante classificar a busca pela solução ótima de acordo com a técnica utilizada. Essa classificação ocorre de acordo com o comportamento da análise em busca da melhor topologia.

Ao se considerar os problemas de otimização que possuem uma função objetivo, nem sempre essa função possuirá uma solução única, podendo assim haver múltiplas respostas e dificultando a verificação de qual seria a melhor delas, ou de até mesmo não existir nenhuma solução adequada para o modelo proposto. Nesses casos, para se obter uma resposta válida, algumas técnicas de otimização se preocupam em lidar com esses problemas matemáticos de forma simplificada. Esses artificios caracterizam métodos heurísticos de otimização, os quais se baseiam em regras, simplificações ou aproximações que reduzem ou limitam a busca por soluções em domínios que são difíceis e/ou pouco compreendidos. Ou seja, os métodos heurísticos são métodos aproximados e empíricos que podem ou não apresentar resultados satisfatórios, dependendo do nível de consistência matemática encontrada no cálculo da função objetivo, caso ela exista explicitamente. Nesses casos, a derivada da função que rege o

problema é nula, sendo esses métodos heurísticos considerados como métodos de ordem zero. Mas uma grande vantagem desses métodos reside no fato de tanto a função objetivo quanto as restrições não necessitarem ter uma representação matemática explícita e, tampouco, que a função objetivo seja contínua ou diferenciável. A otimização evolucionária, descrita a seguir neste capítulo, engloba técnicas que utilizam métodos heurísticos para a busca pela solução ótima, por exemplo.

Quando a busca pela solução ótima envolve aleatoriedade em alguma lei constitutiva, métodos probabilísticos devem ser utilizados. Os métodos evolutivos se baseiam em análises probabilísticas de possíveis soluções para um determinado problema. Porém, a relação entre causa e efeito não é determinada. Em outras palavras, os métodos probabilísticos, como o próprio nome sugere, utilizam um processo de busca randômicas guiadas por decisões probabilísticas para obter o mínimo global, além de serem também ferramentas poderosas para problemas com variáveis discretas. Os algoritmos genéticos, descrito a seguir neste capítulo, constituem um exemplo de uma técnica que utiliza métodos probabilísticos para a busca pela solução ótima.

Mas a utilização de algoritmos numéricos que garantem respostas mais precisas para a solução de problemas caracteriza métodos de programação matemática ou métodos determinísticos. Os métodos determinísticos se baseiam em um conjunto de teoremas e ações sistemáticas para determinar um resultado otimizado ou muito próximo dele, com certo grau de precisão. Em alguns casos dependem de uma função objetivo e, quando isso acontece, utilizam de, pelo menos, a sua primeira derivada em relação às variáveis de projeto. Por isso, é imprescindível e importante que a função, se houver, seja diferenciável e contínua. Os principais métodos determinísticos envolvem programação linear ou não linear, sendo alguns importantes descritos a seguir.

A programação linear se destina a solucionar problemas lineares de otimização, ou seja, problemas em que a função objetivo e as restrições são funções lineares em relação às

variáveis de projeto. A função a ser minimizada deve ser do tipo $f(x) = \sum_{i=1}^n a_i x_i$, sendo os parâmetros a_i constantes. Após o cálculo das derivadas, a solução do problema se encontra

nos extremos de um domínio limitado por um contorno, denominado polítopo, uma vez que os gradientes não são necessariamente nulos. O método Simplex é um exemplo de um método eficiente de programação linear, já que resolve problemas do tipo $Ax = b$ (com $x \geq 0$) e minimiza funções do tipo $f = c^T x$, onde c é um vetor do tipo $n \times 1$, A é uma matriz do tipo $m \times n$ e b é um vetor do tipo $m \times 1$. As variáveis devem ser contínuas. É geralmente utilizado para resolver problemas complexos baseados em desigualdades.

Já a programação não linear caracteriza os métodos de solução de problemas em que a função objetivo e restrições não são lineares em relação às variáveis de projeto. Além disso, possui métodos com ou sem restrição, mesmo que a maior parte dos problemas de otimização apresente restrições. Apesar disso, é importante destacar que os métodos para a solução de problemas com restrições baseiam-se nos métodos para a solução de problemas sem restrições e que, ainda, existem maneiras de transformar métodos com restrição em métodos sem restrição para resolvê-los em seguida. Finalmente, os métodos sem restrição são mais usados.

Inicialmente, com relação aos métodos com restrição, Haftka e Gürdal (1992) e Arora (2012) os definem da seguinte forma, de acordo com a Eq. (2.1):

$$\begin{aligned} \min \quad & f(x) \\ \text{sujeito a} \quad & g_j(x) \leq 0, \quad j=1, \dots, n_g \\ & h_k(x) = 0, \quad k=1, \dots, n_e \end{aligned} \quad (2.1)$$

onde x é o vetor das variáveis de projeto, g_j e h_k são respectivamente vetores que contém as restrições de desigualdade e igualdade, e n_g e n_e referem-se a números de restrições de desigualdade e igualdade, respectivamente.

Nem todos os problemas com restrição possuem ambas restrições de igualdade e desigualdade, simultaneamente. Independentemente disso, a consideração das restrições é fundamental para a identificação do espaço viável. A Fig. 2.4 representa o espaço viável entre três funções, mostrando um ponto que seria o limite desse espaço de acordo com restrições impostas ao problema, delimitando o espaço viável e o inviável.

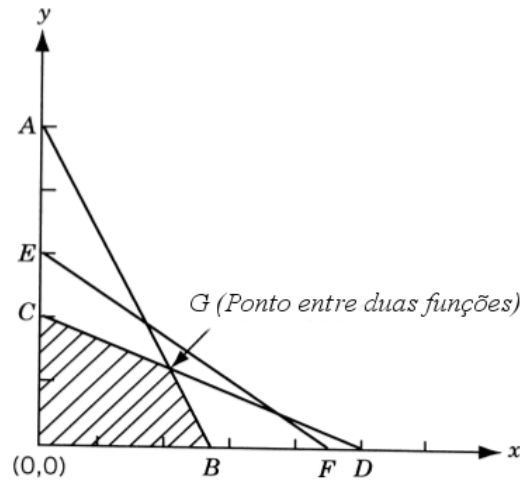


Figura 2.4 – Representação do espaço viável entre funções. Fonte: RAO (2009)

O espaço hachurado apresentado na Fig. 2.4 representa o espaço viável, que é o conjunto de soluções possíveis que satisfazem as funções lineares apresentadas. O ponto G, que pertence ao espaço viável, é também o ponto de interseção entre as duas funções representadas pelos segmentos AB e CD.

Com relação aos métodos sem restrição, eles são classificados em três tipos de acordo com o aspecto da função objetivo: métodos de ordem zero, de primeira ordem e de segunda ordem.

Nos métodos de ordem zero somente o valor da função objetivo é utilizado. São usados quando a função não é diferenciável, sendo difícil obter as derivadas de forma precisa. Entre um dos mais importantes métodos temos o método das Direções Conjugadas de Powell, que sugere uma aproximação quadrática da função a ser minimizada. A grande vantagem aqui é a rapidez de convergência em relação aos métodos de ordens mais elevadas.

Os métodos de primeira ordem utilizam a informação das derivadas da função objetivo para encontrar o ponto ótimo. Esses métodos apresentam uma taxa de convergência linear. O *Steepest Descent* é um dos métodos mais antigos que busca encontrar a direção paralela ao

gradiente (com sinal oposto) ao minimizar $\nabla f^T s = \sum_{i=1}^n \frac{\partial f}{\partial x_i} s_i$, com $s^T s = 1$ e s sendo a direção a ser seguida.

Já os métodos de segunda ordem utilizam os valores da função objetivo, suas derivadas e sua matriz Hessiana. Um dos métodos mais antigos e clássicos é o método de Newton que se

assemelha ao *Steepest Descent* ao minimizar $\nabla f^T s = \sum_{i=1}^n \frac{\partial f}{\partial x_i} s_i$. Porém, a norma Euclideana é substituída pela norma $s^T Q s = 1$, sendo Q a matriz Hessiana da função objetivo.

Vale a pena lembrar aqui que esses métodos são muito usados para a solução de sistemas de equações lineares e não-lineares, uma vez que esses problemas derivam intrinsecamente de problemas de otimização. Por isso são utilizados na maior parte dos programas comerciais atuais de análise estrutural e mecânica que utilizam o Método dos Elementos Finitos.

2.5 Problemas Numéricos

Ao lidar com otimização topológica em conjunto com o Método dos Elementos Finitos, alguns problemas podem ser encontrados e devem ser contornados para que o resultado desejado não seja comprometido. Esses problemas geram instabilidade na solução do processo de otimização estrutural podendo, inclusive, levar a uma topologia final inadequada para as condições definidas. Os três problemas mais comuns são o *checkerboard* (ou tabuleiro de xadrez), a dependência da malha e os mínimos locais (SIGMUND e PETERSSON, 1998). Estes problemas podem ocorrer em conjunto ou isoladamente em uma análise, gerando resultados imprecisos. Uma breve descrição destes problemas será dada a seguir.

2.5.1 Tabuleiro de Xadrez (*checkerboard*)

O nome *checkerboard* refere-se ao problema de formação de regiões alternadas entre sólidos e vazios (cavidades), assemelhando-se a um tabuleiro de xadrez, obtido em resposta à análise de otimização topológica. De acordo com Díaz e Sigmund (1995) e Jog e Haber (1996), este problema ocorre devido ao mal condicionamento da matriz de rigidez do modelo, o que significa uma não convergência adequada da análise. O aspecto desse problema é mostrado na Fig. 2.5.



Figura 2.5 – Problema de tabuleiro de xadrez (*checkerboard*). Fonte: Sigmund e Petersson (1998)

Para sanar esse problema, Díaz e Sigmund (1995) e Jog e Haber (1996) sugerem a utilização de elementos de alta ordem, com 8 e 9 nós, devida a sua maior precisão na obtenção dos valores de deslocamentos e tensões ao custo de um aumento significativo em tempo de processamento.

De acordo com Kikuchi *et al.*, (1984), uma outra opção seria criar uma malha formada por “superelementos”, constituídos de, por exemplo, 4 elementos bilineares isoparamétricos agrupados, todos com os mesmos valores de densidade. Essa estratégia é, de certa forma, um controle de gradiente nas densidades, pois grupos de elementos vizinhos não podem sofrer variações bruscas em suas densidades. Apesar desse “superelemento” ser formado apenas por quadriláteros de baixa ordem, o “superelemento” se comporta como se tivesse graus de liberdade extras, aumentando o custo computacional para solução do problema.

Além disso, Sigmund (1994) ainda sugere a implementação de filtros que verificam a importância daquele elemento na malha, assim como seus vizinhos, evitando remoções indesejadas de elementos ou até mesmo removendo elementos que poderiam causar esta instabilidade. Basicamente, o filtro proposto por Sigmund consiste em aplicar uma modificação diretamente na análise de sensibilidade de malha de elementos finitos em cada iteração, fazendo com que a sensibilidade de cada elemento individualmente dependa de uma média ponderada de 8 elementos vizinhos. Os elementos vizinhos são facilmente identificados, pois Sigmund utilizou malhas regulares em suas discretizações.

Um exemplo de filtro pode ser mapear os elementos vizinhos ao elemento considerado, que são obtidos por um raio de varredura (fixo ou variável) em torno de um elemento central. Se o centroide de um determinado elemento estiver dentro do círculo de raio r que delimita uma região de domínio Ω (Fig. 2.6), esse elemento terá sua densidade considerada no cálculo da

densidade do elemento central. Assim, além da possibilidade de se utilizar malhas genéricas, o resultado final fica menos dependente da discretização.

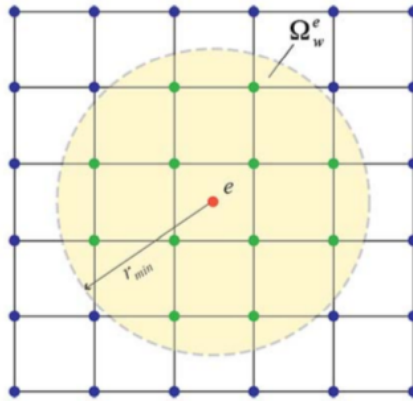


Figura 2.6 – Abrangência do mapeamento de elementos vizinhos do centroide em e . Fonte: Guest e Prévost (2006)

Logo, esses filtros podem ser verificações simples da eficiência de cada elemento ou, até mesmo, funções que regulam a retirada ou permanência dos elementos na malha.

2.5.2 Dependência da Malha

A dependência da malha é um efeito de divergência da topologia ótima final para diferentes discretizações realizadas. Ou seja, para diferentes malhas geradas a fim de realizar a análise de um mesmo modelo predefinido. Ao se refinar uma malha, o efeito esperado seria o de uma melhora da topologia ótima final, ou seja, quanto mais elementos finitos maior deveria ser a precisão da topologia. Porém, quanto maior a quantidade de elementos, maior a tendência da estrutura a se alterar (em comparação com topologias geradas com os mesmos tipos de elementos finitos em menor quantidade), aumentando a quantidade de cavidades (vazios) e alterando qualitativamente a resposta do modelo estrutural, de acordo com a Fig. 2.7.



Figura 2.7 – Problema de dependência da malha em duas topologias diferentes geradas por malhas diferentes para um mesmo problema inicial. Fonte: Sigmund e Petersson (1998).

De acordo com Sigmund e Petersson (1998), o problema da dependência da malha pode ser dividido em duas categorias:

- Obtenção de topologias finais cada vez mais complexas à medida que a malha de elementos finitos vai sendo refinada. Nesse caso, malhas mais refinadas produzem estruturas com maior número de cavidades e membros internos quando comparadas com estruturas geradas a partir de malhas menos refinadas;
- Obtenção de diversas soluções ótimas com mesmo valor da função objetivo. Este problema aparece devido à não unicidade da solução do problema discretizado. Ou seja, várias respostas podem satisfazer a solução do problema e, conseqüentemente, gerando várias topologias diferentes para um mesmo modelo inicial. Na Fig. 2.8, tanto a estrutura (B) quanto a estrutura (C) funcionariam perfeitamente sob o dado caso de carregamento (BENDSØE, 1995; SIGMUND e PETERSSON, 1998).

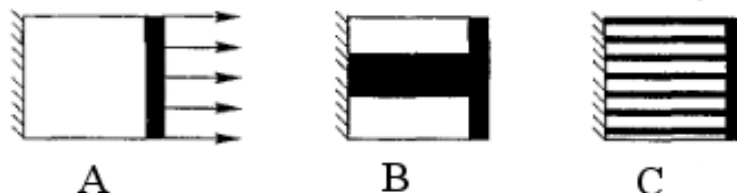


Figura 2.8 – Exemplo de não unicidade de solução, no qual (A) é o domínio inicial e (B) e (C) são duas soluções viáveis. Fonte: Sigmund e Petersson (1998).

Apesar de não existirem maneiras de resolver diretamente o problema da não existência da

solução única, alguns métodos de restrição podem ser aplicados para minimizar o surgimento desse tipo de problema. Haber *et al.* (1996) sugerem uma implementação numérica de controle de perímetro, que representa um somatório das circunferências de todas as cavidades e contornos externos do modelo. Ainda, de acordo com Sigmund e Petersson (1998), restrições globais ou locais da variação de densidade (restrições globais ou locais de gradiente) têm-se mostrado eficientes para a eliminação de possíveis inconsistências numéricas, incluindo o *checkerboard*, ou, no mínimo, suficientes para o enfraquecimento desses tipos de problema.

Sigmund (1997) ainda propõe a implementação de um filtro (inclusive sendo uma extensão de um método proposto por ele próprio para resolver o *checkerboard*) que modifica diretamente o cálculo das sensibilidades através de uma média ponderada sobre elementos vizinhos, baseada em critérios heurísticos.

2.5.3 Mínimos Locais

Os mínimos locais representam a obtenção de diferentes soluções para o mesmo problema ao se escolher diferentes parâmetros de otimização para o algoritmo. Em outras palavras, a não-convexidade do problema leva a possibilidade de encontrar vários mínimos locais e diferentes soluções para a mesma discretização quando são utilizados diferentes parâmetros iniciais no algoritmo de otimização. Isto ocorre porque as provas de convergências dos algoritmos funcionam para programação convexa, enquanto que para programação não convexa apenas se garante a convergência para pontos estacionários, que não são necessariamente mínimos globais. Os algoritmos de otimização global disponíveis são, em sua maioria, incapazes de lidar com uma grande quantidade de variáveis de projetos, o que é o caso da otimização topológica (SIMONETTI, 2016).

Os métodos descritos anteriormente para controlar o aparecimento de instabilidades como tabuleiro de xadrez e dependência da malha tendem a tornar o problema convexo, fazendo com que, dessa forma, não sendo alterados os parâmetros, a topologia final possa ser reproduzida (SIGMUND e PETERSSON, 1998).

As funções envolvidas no problema de otimização topológica, geralmente, são não-convexas, devido às não-linearidades associadas aos problemas reais de engenharia, podendo dessa forma possuir vários mínimos locais.

Para resolver esse problema, Sigmund e Petersson (1998) sugerem um método que consiste em gradualmente modificar o problema de otimização de uma forma artificial e convexa (o qual permite regiões cinzas) para o problema original não-convexo, praticamente composto apenas por regiões com ou sem material. O problema a ser otimizado passa a ser analisado em um processo iterativo (número de passos) e a convergência ocorre em cada passo, gerando uma topologia que pode não ser a ideal mas que pode ser melhorada no próximo passo.

Guedes e Taylor (1997) sugerem ainda uma aproximação de continuação onde os elementos de densidade intermediária são gradualmente incrementados através do ajuste de uma função

peso ω na restrição $\int_V \omega \rho dx \leq V$, sendo V o volume total e ρ a densidade.

2.6 Técnicas Usuais de Otimização Estrutural

A técnica SIMP (*Solid Isotropic Micro-structures with Penalization*) possui sua formulação baseada na Teoria da Homogeneização para calcular a rigidez efetiva do material composto formado pela combinação dos diferentes constituintes. Essa teoria visa descrever as propriedades efetivas de um material composto formado a partir de um material isotrópico de base, além de fornecer um método de parametrização consistente para resolver o problema da melhor distribuição de material em um domínio relaxado. A homogeneização também permite uma descrição contínua de meios não contínuos, como descrito acima. Na aplicação deste tipo de técnica, as propriedades físicas dos materiais, como peso específico e módulo de elasticidade são descontínuos e oscilam muito rapidamente entre os valores associados a cada componente do material composto. Este tipo de característica da técnica pode representar o problema físico de maneira inadequada.

A solução proposta pela Teoria da Homogeneização consiste na criação de “células” de

material que possuem comportamento constante entre si. As microestruturas são formadas pela mistura de materiais homogêneos, criando então um meio não homogêneo. Assume-se que o material é formado pelo agrupamento periódico dessas células unitárias, de modo que as propriedades efetivas do meio homogeneizado sejam parametrizadas apenas por características geométricas dessas microestruturas artificiais. Assim, na aproximação são consideradas ainda dois tipos de variáveis: a densidade do material e os parâmetros que descrevem a geometria da célula unitária. Isto implica um problema de otimização muito caro computacionalmente, pois, dependendo da geometria da célula unitária, muitos parâmetros devem ser otimizados. Por isso, uma tática utilizada por Bendsøe (1989) consiste em utilizar microestruturas artificiais para que a única variável de projeto seja a densidade de material $\rho(x)$, garantindo a existência entre os dois materiais isotrópicos de base e simplificando a análise do projeto, sendo matematicamente expressa da seguinte forma:

$$\begin{aligned} x &\in \Omega \\ E_{ijkl} &= \rho^n E_{ijkl}^0 \\ \int_{\Omega} \rho(x) d\Omega &= V \\ 0 &\leq \rho(x) \leq 1 \end{aligned} \quad (2.2)$$

onde n é um parâmetro adimensional que relaciona a proporção entre os tensores constitutivos isotrópicos E_{ijkl} e E_{ijkl}^0 e a densidade ρ .

Assim, a aproximação apresentada pela Eq. (2.2) resume a técnica SIMP e basicamente propõe que a microestrutura é desconhecida (daí a denominação artificial), mas sua rigidez é conhecida (PEREIRA, 2006). Além disso, a formulação está intimamente relacionada com a Teoria da Homogeneização, visto que, de acordo com a Eq. (2.2), após a aplicação do método da homogeneização sobre uma célula com densidade de material ρ , o tensor constitutivo isotrópico homogeneizado, obtido pelas equações do método da homogeneização, será igual a E_{ijkl} . Existe claramente uma proporcionalidade entre o tensor constitutivo homogeneizado E_{ijkl} e o tensor constitutivo isotrópico “poroso” do material de base E_{ijkl}^0 . É uma das formulações mais utilizadas na literatura no que diz respeito à penalização da densidade.

A técnica TSA (*Topological Sensitivity Method*) consiste na avaliação da sensibilidade de uma função custo para a criação de uma nova cavidade. Nos pontos do domínio onde essa

sensibilidade for baixa o suficiente, o material pode ser progressivamente eliminado. Matematicamente, o conceito original da técnica, que também é denominada de Derivada Topológica, está relacionada com a sensibilidade de uma determinada função custo ψ quando a topologia do domínio avaliado Ω é alterada pela inserção de um pequeno orifício de raio ϵ centrado em \hat{x} (MARCZAK, 2008). Isoladamente, esta técnica não é muito utilizada pela literatura em relação às outras aqui descritas devido à dificuldade de obtenção do resultado da derivada topológica. Porém, costuma ser usada em conjunto com outras técnicas para melhores resultados.

Há uma técnica de otimização baseada na função *Level Set* que descreve a evolução da interface entre dois domínios. Foi proposta inicialmente por Osher e Sethian (1988) como um método versátil para representar implicitamente as interfaces submetidas a um processo de evolução em um sistema de coordenadas Euleriano. A técnica está intimamente relacionada com a otimização de forma, uma vez que alterações na topologia envolvendo incrementos no número de cavidades em relação ao domínio de projeto inicial não são permitidas; contudo, uma diminuição em seu número é possível. Porém, para resolver esse problema, outras técnicas, como a derivada topológica, podem ser incorporadas para a formação de novas cavidades no interior do domínio. Como exemplo, os trabalhos de Yulin e Xiaoming (2004) e Allaire *et al.* (2005) tratam desse assunto.

Ao longo do processo de otimização, formas complexas e mudanças topológicas podem ser obtidas através de uma função $\varphi(x)$ implícita, de alta dimensão, denominada função *level set*, representada por um domínio composto por material Ω , um domínio composto por cavidades $D \setminus \Omega$ e uma interface entre ambos domínios Γ , todos estes contidos em um domínio de projeto D , de tal forma que:

$$\left\{ \begin{array}{ll} \varphi(\mathbf{X}) > c, \forall \mathbf{X} \in \Omega & (\textit{material}) \\ \varphi(\mathbf{X}) = c, \forall \mathbf{X} \in \Gamma & (\textit{interface}) \\ \varphi(\mathbf{X}) < c, \forall \mathbf{X} \in D \setminus \Omega & (\textit{cavidade}) \end{array} \right\} \quad (2.3)$$

onde c é uma constante de valor geralmente nulo ($c = 0$) e \mathbf{X} é um ponto qualquer do domínio de projeto D (SETHIAN e WIEGMANN 2000; WANG *et al.* 2003).

Neste contexto, a função *level set* serve para distinguir e separar os dois domínios presentes no domínio de projeto. Os parâmetros relacionados à parametrização da função *level set* representam as variáveis de otimização. O modelo mecânico inclui o mapeamento da geometria e a discretização do modelo estrutural, e ambos influenciam no comportamento do projeto resultante. O desempenho e os requisitos de um problema inicial são formulados através da função objetivo e das restrições, sendo analisadas através do modelo mecânico. A escolha mais comum para parametrizar a função *level set* é a utilização do método dos elementos finitos, parametrizando funções bases de elementos finitos em malha fixa (BELYTCHKO *et al.*, 2003; WANG *et al.*, 2003; ALLAIRE *et al.*, 2004).

Em relação aos métodos aproximados, os Algoritmos Genéticos (AG) são técnicas heurísticas inspiradas na Teoria da Evolução de Darwin e na lei natural da sobrevivência do mais forte, o Algoritmo Genético (ou AG) é um procedimento de pesquisa global que visa melhorar gradualmente a solução pela evolução de populações através de operadores que imitam a evolução natural, como reprodução, cruzamento e mutação e executa uma troca de informação aleatória para criar descendentes superiores (GOLDBERG, 1989). Possui as seguintes características:

- As operações são realizadas através de *strings* binários) de comprimento finito (denominados cromossomos) em vez de parâmetros reais;
- Consideram um grupo de pontos na busca da solução em cada iteração (população de pontos);
- Utilizam uma busca aleatória baseada em informações prévias ao invés do cálculo de gradientes;
- Necessita de um espaço (domínio) delimitado para codificar os parâmetros.

As operações são realizadas através de reprodução, cruzamento e mutação. A reprodução consiste simplesmente em um processo de decidir quais *strings* devem sobreviver e quantas cópias delas devem ser produzidas ao longo do processo. A decisão é feita comparando-se a aptidão de cada *string* com a afinidade média da população. Essa aptidão representa exatamente a função objetivo do problema ou uma combinação da função objetivo com as respectivas restrições, sendo que, de forma geral, em problemas de maximização uma *string*

com uma aptidão maior receberá proporcionalmente mais cópias na nova população e em problemas de minimização uma *string* com uma aptidão menor receberá menos cópias.

O cruzamento é a operação pela qual duas *strings* com alta aptidão (pais) realizam a produção de dois descendentes, misturando e combinando suas qualidades desejáveis através de um processo aleatório. Após a reprodução, o cruzamento prossegue em dois passos. Primeiro, duas *strings* recém-reproduzidas são selecionadas aleatoriamente para a reprodução. A seguir, alguns *bits* (códigos binários) escolhidos ao acaso são trocados entre essas duas *strings*. Vários métodos podem ser utilizados para escolher as partes a serem permutadas como, por exemplo, o cruzamento uniforme, que cria uma *string* binária de forma aleatória denominada máscara (SYSWERDA, 1989). Essa máscara age como uma peneira e obriga uma troca de posição dos *bits* quando sua posição correspondente for nula. Do contrário, nada é alterado (Fig. 2.9).

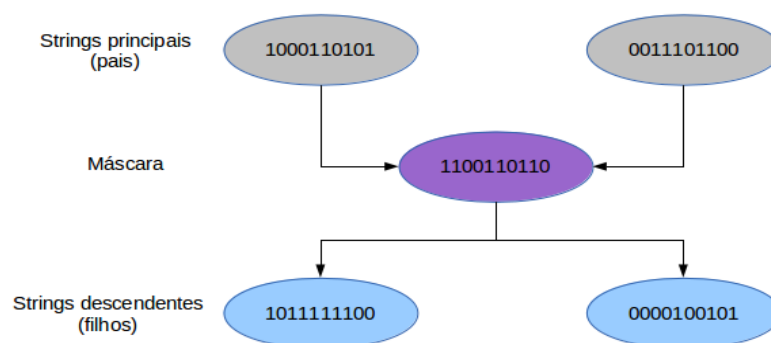


Figura 2.9 – Método do Cruzamento Uniforme. Fonte: Adeli e Sarma (2006)

Logo, dadas *strings* principais, a *string* aleatória máscara gera *strings* descendentes modificadas de acordo com os *bits* nulos que possui.

Reprodução e cruzamento realizam simples operações, gerando *strings* aleatórias, além de cópias e trocas de *bits*. Ainda, o cruzamento é um meio bastante eficaz em troca de informações e combinação de *strings* de alta aptidão. Assim, reprodução e cruzamento, juntos, potencializam a confiabilidade da técnica de algoritmos genéticos.

Por fim, a mutação funciona como uma garantia de variabilidade, sendo responsável pela

inserção de pequenas mudanças aleatórias nas *strings* descendentes. Alguns dígitos em uma posição particular das *strings* podem ser eliminados durante o processo de reprodução e cruzamento, sendo impossível apenas para ambos recuperá-los e possivelmente criando restrições adicionais no domínio de busca e, assim, soluções indesejadas, a mutação permuta *bits* de todas as *strings* de acordo com uma variação imposta.

A eficiência desta técnica de otimização é dependente de uma eficiente representação do modelo a ser avaliado a partir das operações realizadas através dos cromossomos que regem o problema. É importante ressaltar que este método não impede a formação de estruturas inadequadas ou de problemas clássicos da otimização estrutural (via Método dos Elementos Finitos), como o *checkerboard* (ou tabuleiro de xadrez), sendo este uma alternância entre presença de material e de cavidades em uma dada região do domínio, podendo ocasionar instabilidade numérica e dependência da malha.

Uma outra técnica de otimização são os Algoritmos de Colônia de Formigas (ACO, do inglês *Ant Colony Optimization*) que, como o próprio nome indica, foram inspirados nas formigas não apenas no comportamento que elas apresentam na busca por alimento, mas também no que diz respeito à organização do trabalho e cooperação entre si. Uma colônia de insetos é muito bem organizada e as atividades dos insetos são realizadas coletivamente de forma organizada. Assim como os Algoritmos Genéticos, é também uma técnica probabilística e, portanto, de ordem zero.

É uma técnica cuja ideia central é a simulação de uma metáfora natural do real comportamento de uma colônia de formigas. As formigas (reais) são capazes de encontrar o caminho mais curto a partir de uma fonte qualquer de alimento para o seu ninho sem o uso de pistas visuais, mas explorando uma substância química denominada feromônio. Enquanto caminham, as formigas depositam feromônio no chão criando, assim, um rastro. Outras formigas adicionam seus feromônios ao que foi depositado anteriormente por outras formigas. Quanto mais feromônio em uma rota, maior a probabilidade daquela rota ser escolhida pelas formigas. Este comportamento serviu de inspiração para o desenvolvimento da técnica ACO, onde um conjunto de formigas artificiais coopera para resolver um problema, trocando informações indiretamente através de um rastro de feromônio artificial. Este rastro é

acumulado em tempo de execução através de um mecanismo de aprendizagem que garante “recompensas” a boas soluções de problemas. Assim, a técnica ACO é baseada em um conjunto de formigas artificiais paralelas que, de forma iterativa, constroem e atualizam soluções de problemas definidos por uma sequência de estados e construídos através de uma pesquisa probabilística. As formigas artificiais começam a partir de um estado aleatório e cada uma escolhe um novo estado para “experimental”, usando uma função probabilística baseada principalmente na intensidade de feromônio. Ao final de cada ciclo, a quantidade de feromônio remetentes às melhores soluções é aumentado de acordo com uma regra de aprendizagem (KAVEH *et al.*, 2008).

2.7 ESO (*Evolutionary Structural Optimization*)

A técnica ESO foi desenvolvida a partir de observações de estruturas na natureza, tais como ossos, árvores e conchas, em uma tentativa de avaliar como essas estruturas adquiriram forma ao longo do processo de evolução natural (XIE e STEVEN, 1993). É uma técnica baseada na Computação Evolucionária que é relativamente simples, quando comparada a outras técnicas presentes na literatura, pois modifica a topologia da estrutura mediante heurística de remoção gradual de elementos finitos da malha, correspondentes as regiões que não contribuem efetivamente para a boa performance da estrutura. Além disso, essa técnica foi primeiramente aplicada no processo de otimização topológica para a minimização das tensões em componentes mecânicos (SIMONETTI, 2016).

Em outras palavras, o ESO consiste em uma remoção gradual de elementos finitos da malha gerada em regiões que não contribuem efetivamente para a estrutura, através de um critério de rejeição. É um procedimento simples de retirada, considerado, portanto, “*hard-kill*”, ou seja, há uma remoção única (por iteração) e definitiva dos elementos que não atendem ao critério de rejeição estabelecido. Se diferencia de outras técnicas como o SIMP, pelo fato da remoção ser gradual. Ainda, uma grande vantagem está no fato da topologia e da forma serem determinados simultaneamente (XIE e STEVEN, 1997). A Fig. 2.10 mostra o processo de retirada de elementos da técnica ESO.

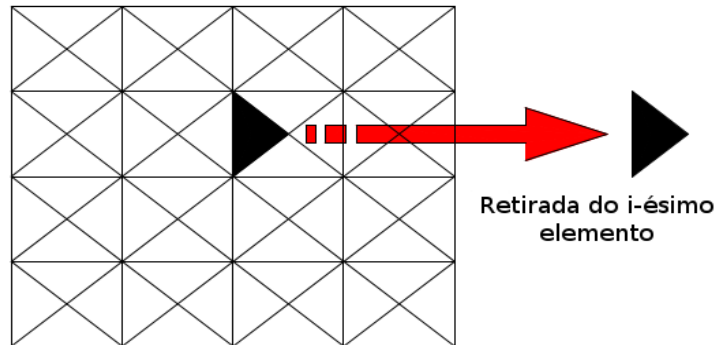


Figura 2.10 – Técnica ESO realizando a retirada do i -ésimo elemento de uma malha triangular qualquer. Fonte: Almeida (2014)

De acordo com Proos (2006), a determinação dos elementos a serem removidos foi originalmente feita comparando as Tensões de von Mises de cada elemento em relação à Tensão Máxima de von Mises de toda a estrutura. A tensão de von Mises pode ser determinada pela seguinte relação que envolve as tensões principais $\bar{\sigma}_1$, $\bar{\sigma}_2$ e $\bar{\sigma}_3$:

$$\sigma_{max}^{VM} = \frac{\sqrt{2}}{2} [(\bar{\sigma}_1 - \bar{\sigma}_2)^2 + (\bar{\sigma}_2 - \bar{\sigma}_3)^2 + (\bar{\sigma}_1 - \bar{\sigma}_3)^2]^{\frac{1}{2}} \quad (2.4)$$

Para cada elemento, a tensão de von Mises σ_e^{VM} é obtida através da média das tensões avaliadas em cada nó do elemento finito considerado. Assim, determinada a tensão de von Mises do elemento, σ_e^{VM} , a tensão máxima de von Mises atuante na estrutura, σ_{max}^{VM} e, preestabelecido um coeficiente de remoção denominado FRR ou fator de raio de rejeição, tem-se a seguinte inequação:

$$\sigma_e^{VM} < FRR \cdot \sigma_{MAX}^{VM} \quad (2.5)$$

Desta forma, os elementos que satisfazem a inequação são removidos da estrutura a cada iteração, sendo estes considerados ineficientes. Após a remoção dos elementos na iteração i , é necessário atualizar o FRR , uma vez que há alteração no campo de tensões nos elementos remanescentes. Essa atualização é feita através de um coeficiente denominado FER, ou fator evolucionário de remoção, e acontece segundo a Eq. (2.6).

$$FRR_{i+1} = FRR_i + FER, \quad i=0,1,2,3 \dots \quad (2.6)$$

Em um determinado momento, não há mais remoção de elementos quando a Eq. (2.5) não é mais satisfeita. Nesse instante, é necessária uma atualização do FRR conforme a Eq. (2.6), acompanhando o aumento gradual do campo de tensão dos elementos remanescentes. Essa correção é feita ao longo do processo iterativo até que a topologia ótima final seja obtida, de acordo com a escolha do usuário ou segundo um critério de parada preestabelecido, geralmente o de volume final em comparação com o volume inicial da estrutura. De acordo com Querin *et al.* (1998), para garantir uma melhor convergência, devem ser adotados pequenos valores de FER (da ordem de 1%).

A grande antagem da utilização do critério de remoção de tensão de von Mises está no fato da simples implementação e da boa precisão dos resultados, comparados na literatura. Porém, outros tipos de critério podem ser implementados (incluindo outros tipos de tensão, como tensões principais, de compressão ou de tração) para a remoção de elementos, tais como deformação (XIE e STEVEN, 1997), rigidez (CHU *et al.*, 1996), frequência natural (ZHAO, STEVEN e XIE, 1996), carga crítica de flambagem (MANICKARAJAH, 1998) e fluxo de calor (LI *et al.*, 1997). A escolha do critério é feita em função do objetivo da otimização (analiticamente, é a definição da função objetivo), ou seja, o que se deseja minimizar ou maximizar.

Matematicamente, o processo de remoção de elementos da técnica ESO pode ser descrita da seguinte forma:

$$D(j) = \begin{cases} D_0, & \text{sej} \in \Gamma \\ 0, & \text{sej} \in \bar{\Gamma} \end{cases} \quad (2.7)$$

onde $D(j)$ é a matriz constitutiva do ponto $j \in \Omega$; D_0 é a matriz constitutiva inicial do projeto, Ω é o domínio da estrutura, dado por $\Omega = \Gamma + \bar{\Gamma}$; Γ é o conjunto dos elementos

que não serão removidos da estrutura (sólido), dado por $\Gamma = \left\{ \Omega / \left(\frac{\sigma_e^{JM}}{\sigma_{MAX}^{JM}(\Omega)} \right) \geq FRR \right\}$ e $\bar{\Gamma}$ é o

conjunto dos elementos que serão removidos da estrutura (vazio), dado por

$$\bar{\Gamma} = \Omega - \Gamma = \left\{ \Omega / \left(\frac{\sigma_e^{VM}}{\sigma_{MAX}^{VM}(\Omega)} \right) < FRR \right\}, \text{ considerando } \sigma_e^{VM} \text{ a tensão de von Mises do elemento;}$$

σ_{max}^{VM} a tensão máxima de von Mises atuante na estrutura e FRR o fator de raio de remoção que é definido previamente.

O processo evolucionário, cujo critério de remoção seja o de tensão, pode ser definido nas seguintes etapas:

- 1) Discretizar o domínio de análise com uma malha de elementos finitos, assim como definir todas as condições de contorno e forças aplicadas;
- 2) Calcular as tensões principais e de von Mises em cada elemento (σ_e^{VM});
- 3) Remover (penalizar) os elementos que satisfazem a Eq. (2.5) de forma integral ou de acordo com um limite prescrito de elementos a ser removido por iteração (volume, por exemplo);
- 4) Repetir os passos 2 e 3 até que seja atingido o equilíbrio e/ou que não haja mais elementos a serem removidos;
- 5) Atualizar FRR , conforme a Eq. (2.6), e iniciar uma nova retirada de elementos, repetindo os passos 2, 3 e 4.

Os passos descritos acima ainda podem ser definidos conforme Fig. 2.11.

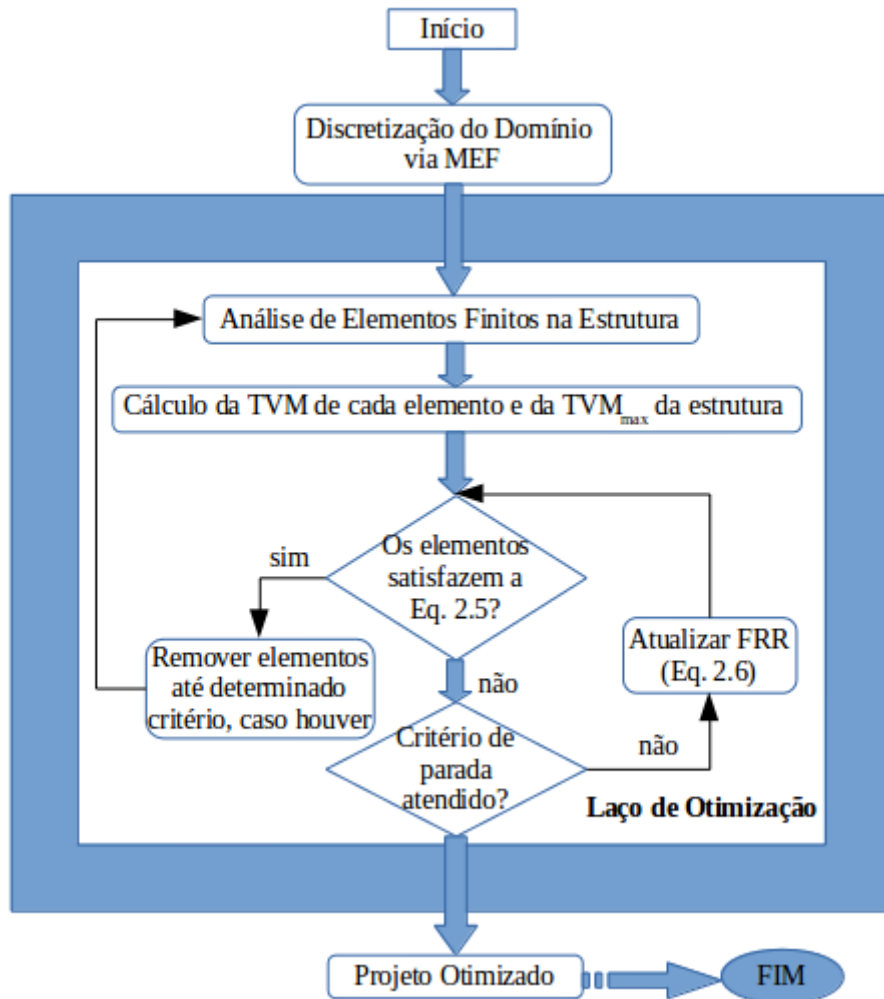


Figura 2.11 – Fluxograma da técnica ESO com Tensão de von Mises como critério de remoção de elementos

A técnica ESO consiste em um processo simples e direto de remoção de elementos. O critério que limita a retirada de elementos da malha pode ser uma condição que verifica a quantidade de elementos a serem removidos por iteração ou até mesmo nem existir nenhuma condição especial. Ou seja, todos os elementos que atenderem a Eq. (2.5) serão retirados de uma única vez, dependendo apenas de uma implementação adequada do algoritmo. É importante lembrar que a quantidade de elementos removidos pode influenciar os valores do campo de tensões, o que pode provocar topologias finais ótimas diferentes.

Por ser uma técnica de simples implementação, ela é muito utilizada pela literatura. Consequentemente, algumas melhorias foram sugeridas e isso acabou originando outras

versões dessa técnica, como é o caso do AESO (*Additive Evolutionary Structural Optimization*), BESO (*Bidirectional Evolutionary Structural Optimization*) e o SESO (*Smoothing Evolutionary Structural Optimization*). De forma sucinta, o AESO, a partir de uma decisão adequada implementada no algoritmo, realiza a otimização de forma a minimizar tensão elevada em pontos específicos da malha (QUERIN *et al.*, 2000). Já no BESO, elementos podem ser removidos e incluídos na malha, sendo basicamente uma modificação da combinação das técnicas ESO e AESO (QUERIN *et al.*, 1998). Quanto ao SESO, técnica que será usada neste trabalho, será descrito a seguir.

2.8 SESO (*Smoothing Evolutionary Structural Optimization*)

A forma simplificada de obtenção da topologia ótima do ESO proporcionou estudos mais aprofundados. Para incrementar e melhorar o princípio de remoção de elementos da técnica ESO, alguns procedimentos foram sugeridos. Inicialmente, notou-se que a remoção dos elementos em cada iteração que satisfizesse a Eq. (2.5) levava à retirada, muitas vezes, precipitada ou precoce de um elemento que não deveria ser eliminado. Isto se dava, muitas vezes, pelo fato de que no processo evolutivo certo elemento que não deveria sair acabava sendo retirado para atender a inequação de tensões máximas, gerando uma solução forçada, não-ótima e com grandes possibilidades de desenvolvimento de uma região instável de “tabuleiro de xadrez” (checkerboard), o qual é um dos maiores problemas do ESO. Para resolver esse problema, o SESO propõe uma organização dos elementos que atendem a Eq. (2.5) tal que $p\%$ desses elementos sejam removidos e $1-p\%$ sejam devolvidos à estrutura. Esta devolução é realizada por uma função reguladora, que realiza uma suavização do procedimento ou, em outras palavras, pondera os elementos com maiores tensões, assim como em sua vizinhança. É considerada uma técnica do tipo “*soft-kill*”, onde uma parcela dos elementos removidos retornará à estrutura. Esse procedimento reduz o surgimento do efeito o “tabuleiro de xadrez”, pois os vazios de possíveis regiões instáveis são devolvidos graças aos seus vizinhos. Tal procedimento pode ser interpretado da seguinte forma:

$$D_i(j) = \begin{cases} D_0, & \text{se } j \in \Gamma_i \\ D_0 \cdot \eta_j(\bar{\Gamma}), & \text{se } j \in \Gamma_{GS} \\ 0, & \text{se } j \in \Gamma_{LS} \end{cases} \quad (2.8)$$

em que $\bar{\Gamma} = \Gamma_{LS_i} + \Gamma_{GS_i}$ representa o domínio dos elementos do problema, Γ_{LS_i} representa o domínio dos elementos que devem ser excluídos efetivamente, Γ_{GS_i} representa o domínio dos elementos que são devolvidos a estrutura, $0 \leq \eta(\bar{\Gamma}) \leq 1$ é a função reguladora que pondera o valor da razão $\sigma_e^{VM} / \sigma_{MAX}^{VM}$ dentro do domínio $\bar{\Gamma}$ e pode eliminar o problema do “tabuleiro de xadrez” e i é a iteração atual de otimização (ALMEIDA *et al.*, 2011). A suavização proposta pode ser realizada através de $\eta(\bar{\Gamma})$ usando uma função linear do tipo $\eta(\bar{\Gamma}) = \alpha j + \beta$ ou uma função trigonométrica do tipo $\eta(\bar{\Gamma}) = \text{sen}(\alpha j)$. A Fig. 2.12 ilustra a ideia utilizada na suavização.

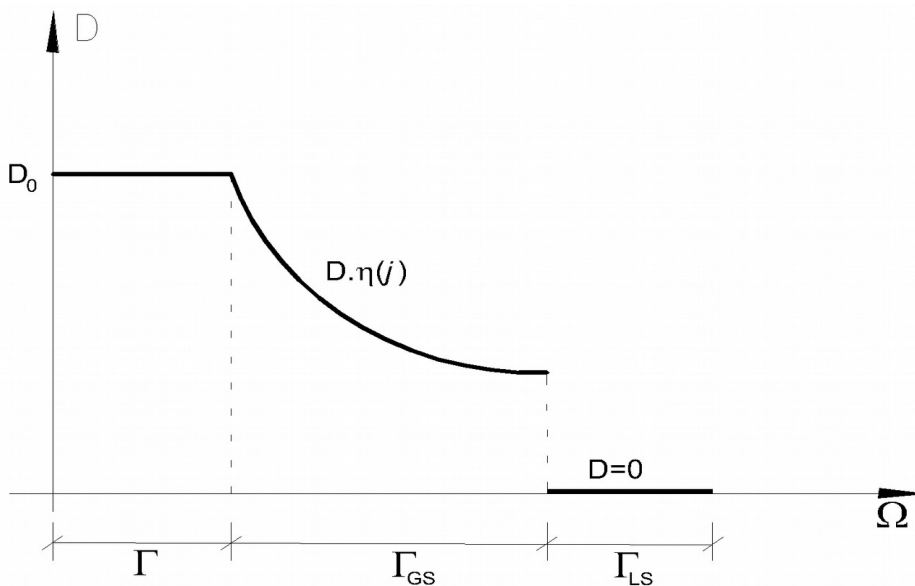


Figura 2.12 – Suavização da quantidade de elementos retirados na iteração i . Fonte: ALMEIDA *et al.* (2011)

Assim, se um elemento satisfaz a Eq. (2.5) referente à técnica ESO, aquele elemento só poderá ser removido se ao menos uma de suas bordas ou lados não estiverem conectadas a nenhum outro elemento da estrutura (QUERIN, 1998).

Uma grande vantagem da técnica de otimização SESO em relação às detradadas na literatura

reside no fato de ter uma formulação simplificada e de fácil implementação computacional, dispensando a necessidade do cálculo de gradientes e derivadas para se obter a topologia ótima. Como desvantagem, tem-se que a topologia ótima não é obtida em uma única etapa, fazendo com que seja necessário um processo iterativo com respostas intermediárias até se chegar ao ótimo. Além disso, não há uma garantia de se obter o ótimo global, uma vez que os parâmetros de remoção podem variar. Entretanto, uma solução cujo ótimo é local pode ser bastante eficiente, uma vez que essa solução pode atender às solicitações impostas inicialmente, além de ser obtida mais rapidamente que o ótimo global.

É importante salientar que, embora nem sempre seja possível o cálculo do gradiente da função objetivo que rege o problema, no caso de funções não diferenciáveis, há pesquisas na literatura acerca de estratégias de resolução de problemas de otimização não diferenciáveis e irrestritos definidos no \mathcal{R}^n , como é o caso dos subgradientes. Segundo Venceslau (2013), a pesquisa de soluções para a resolução de problemas de otimização não diferenciáveis é importante porque o gradiente é uma rica fonte de informações sobre o comportamento da função objetivo nas vizinhanças de um ponto considerado, viabilizando assim a busca pelo ótimo, ainda que restrita a um âmbito local. Quando a função objetivo for não diferenciável, os métodos de otimização tradicionais, que visam o cálculo de gradientes, não se aplicam e novos métodos devem ser especialmente desenvolvidos para que a busca pelo ótimo possa ser realizada de uma forma eficaz (obtenção efetiva do ótimo) e eficiente (relativo à complexidade computacional). Assim, o método do subgradiente, o método do ε -subgradiente, o método dos planos cortantes e o método dos feixes são exemplos de adaptações de métodos tradicionais utilizados para o cálculo de gradientes de problemas de otimização não diferenciáveis (IZMAILOV e SOLODOV, 2007).

O processo evolucionário, cujo critério de remoção seja o de tensão, definido para a técnica SESO pode ser então reescrito nas seguintes etapas:

- 1) Discretizar o domínio de análise com uma malha de elementos finitos, assim como definir todas as condições de contorno e forças aplicadas;
- 2) Calcular as tensões principais e de von Mises em cada elemento (σ_e^{VM});
- 3) Remover (penalizar) p% dos elementos que satisfazem a Eq. (2.5) e devolver 1-p%

dos elementos removidos de acordo com funções reguladoras implementadas no algoritmo;

- 4) Repetir os passos 2 e 3 até que seja atingido o equilíbrio e/ou que não haja mais elementos a serem removidos;
- 5) Atualizar FRR , conforme a Eq. (2.6), e iniciar uma nova retirada de elementos, repetindo os passos 2, 3 e 4.

Os passos anteriormente descritos podem ser definidos conforme Fig. 2.13.

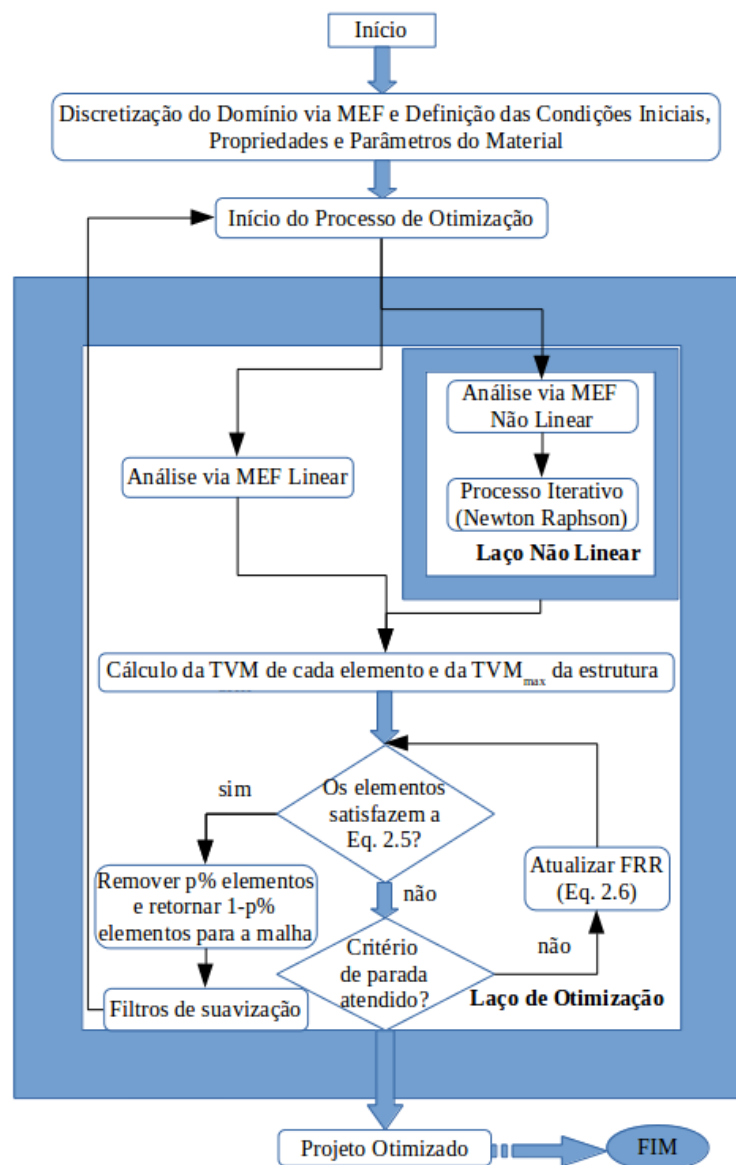


Figura 2.13 – Fluxograma da técnica SESO com Tensão de von Mises como critério de remoção de elementos

É um processo um pouco diferente do descrito na técnica ESO, onde elementos podem retornar para a estrutura caso seja necessário. Todo o processo ainda é o mesmo, exceto pela presença de funções que ajudam a verificar os elementos mais importantes para o modelo estrutural.

Quanto à dependência da malha, por exemplo, o SESO propõe a inserção de um filtro que consiste em aplicar um operador matemático sobre uma função não regular capaz de obter uma função nova, mais suave. Assim, usa-se como parâmetro de controle desse filtro um raio de abrangência. Quanto maior for o raio maior será a suavização da distribuição das tensões de von Mises do elemento central no interior do domínio inicial (Fig. 2.14).

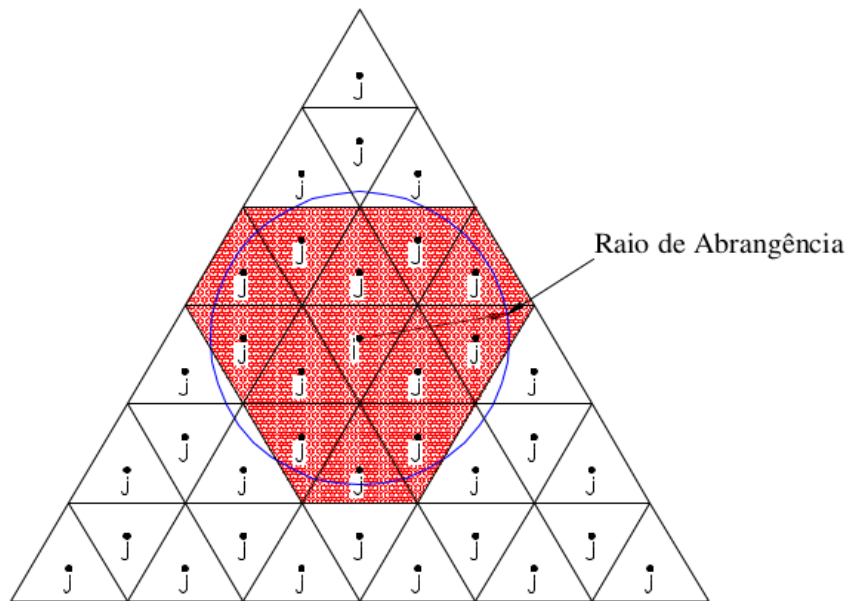


Figura 2.14 – Elemento central i sofrendo influência dos elementos vizinhos dentro de um círculo de raio r . Fonte: Almeida (2014)

A maneira pela qual a tensão do elemento i sofre influência das tensões dos elementos j dentro de uma circunferência de raio r é dada pela equação:

$$\sigma_i^{\text{ponderado}} = \frac{\sigma_i^{vm} \cdot A_i + \bar{w} \cdot \sum_j^{NV} \sigma_j^{vm} \cdot A_j}{A_i + \bar{w} \cdot \sum_j^{NV} A_j} \quad (2.9)$$

em que o termo $\sigma_i^{\text{ponderado}}$ é a tensão de von Mises ponderada do elemento i , que, inicialmente, apresenta tensão de von Mises σ_i^{vm} , A_i é a área do elemento i , A_j é a área do elemento j e σ_j^{vm} a tensão de von Mises do elemento j . O peso \bar{w} é escrito através da equação:

$$\bar{w} = \frac{\sum_{j=1}^{NV} w_j}{NV} \quad (2.10)$$

e

$$w_j = \frac{R_{\max} - R_{ij}}{R_{\max}} \quad (2.11)$$

em que NV representa o número de elementos finitos cujos centroides estão no interior da circunferência de raio de abrangência R_{\max} e R_{ij} é a distância entre os centroides dos elementos i e j . Com o aumento do raio, os gradientes ficam restritos a valores menores, logo, a transição entre sólido (material) e vazio (sem material) torna-se mais suave, gerando mais elementos com tensões intermediárias.

Portanto, o SESO nada mais é do que uma melhora do ESO, com a inclusão de uma ou mais funções matemáticas que melhoram o processo de discretização da estrutura na obtenção da topologia ótima, sendo esta a técnica que será adotada neste trabalho para a resolução dos problemas de otimização.

2.8.1 Índice de Performance

O Índice de Performance é um parâmetro adimensional que mede a eficiência do desempenho estrutural, podendo auxiliar como critério de parada do processo de otimização.

Liang *et al.* (2000) propuseram em seu trabalho a seguinte equação para o cálculo do IP:

$$IP = \left(\frac{\sigma_{0,\max}^{\text{VM}}}{\sigma_{i,\max}^{\text{VM}}} \right) \frac{\rho_0 V_0}{\rho_i V_i} = \left(\frac{\sigma_{0,\max}^{\text{VM}}}{\sigma_{i,\max}^{\text{VM}}} \right) \frac{V_0}{V_i} \quad (2.12)$$

em que V_0 e V_i representa o volume inicial e da i -ésima iteração, $\sigma_{0,max}^{VM}$ e $\sigma_{i,max}^{VM}$ são as Tensões Máximas de von Mises inicial e da i -ésima iteração e ρ_0 e ρ_i representa a densidade inicial e da i -ésima iteração, que são iguais para materiais incompressíveis, respectivamente.

Neste contexto, o índice de performance expresso pela Eq. (2.12), pode ser escrita como:

$$IP = \left(\frac{\sigma_{0,max}^{VM}}{\sigma_{i,max}^{VM}} \right) \frac{A_0 t}{\sum_{j=1}^{NE} A_j t_j} = \left(\frac{\sigma_{0,max}^{VM}}{\sigma_{i,max}^{VM}} \right) \frac{A_0 t_0}{\sum_{j=1}^{NE} A_j t_j \eta(j)} \quad (2.13)$$

em que t_0 e t_j são, respectivamente, a espessura inicial e a espessura do j -ésimo elemento na i -ésima iteração e A_0 é a área inicial do elemento. Assim, a definição da topologia ótima é verificada por esse índice de performance, sendo este um fator de monitoramento da estrutura. Se esse índice cai de forma acentuada, é uma forte indicação de que passou por uma configuração local ideal ou estacionário. Entretanto, não há garantia de que a resposta obtida seja considerada, de fato, uma topologia ótima final pronta para ser desenvolvida, mas sim uma configuração ideal de estudos para um futuro projeto de engenharia.

2.8.2 Número de Sensibilidade

Uma boa característica de algumas técnicas de otimização é o fato de possuírem certa flexibilidade na implementação de critérios de remoção, como é o caso das técnicas ESO e SESO. Além do critério de remoção por tensão (podendo ser von Mises ou tensões principais, por exemplo), critérios em relação aos deslocamentos também merecem destaque. De acordo com Chu *et al.* (1996), a ideia de remoção é a mesma descrita para tensão, ou seja, os elementos com menores números de sensibilidade são removidos da estrutura. Logo, os elementos removidos são os que apresentam menor energia de deformação, ou seja, contribuem menos para a rigidez da estrutura (LANES e GRECO, 2013).

Em termos matemáticos, se o sistema for conservativo (o que implica que existe igualdade entre a energia de deformação da estrutura e o trabalho das forças externas), a energia de deformação U de uma considerada estrutura pode ser definida por:

$$U = \frac{1}{2} \{f\}^T \{u\} \quad (2.14)$$

em que $\{f\}$ o vetor de forças nodais e $\{u\}$ o vetor de deslocamentos nodais. Aqui é importante perceber a relação entre a energia de deformação e a rigidez da estrutura. De acordo com o método dos elementos finitos, há uma correlação entre a matriz de rigidez, o vetor de forças nodais e o vetor de deslocamentos, ou $[k]\{u\} = \{f\}$, que será descrito no próximo capítulo.

Se o i -ésimo elemento for removido, a matriz de rigidez da estrutura é alterada. Considerando que a energia de deformação é também alterada dessa maneira e o vetor de forças não sofre modificações, além de ignorar os efeitos de alta ordem, essa alteração provoca uma variação de deslocamentos Δu que pode ser expresso por:

$$\{\Delta u\} = -[\bar{K}]^{-1} [\Delta \bar{K}] \{u\} \quad (2.15)$$

em que $[\bar{K}]$ a matriz de rigidez global. Combinando as Eqs. (2.14) e (2.15), tem-se:

$$\Delta U = \frac{1}{2} \{f\}^T \{\Delta u\} = -\frac{1}{2} \{f\}^T [\bar{K}]^{-1} [\Delta \bar{K}] \{u\} = \frac{1}{2} \{u_i\}^T [\bar{K}_i] \{u_i\} \quad (2.16)$$

em que $\{u_i\}$ o vetor de deslocamentos do i -ésimo elemento. Há também uma mudança da matriz de rigidez global $[\bar{K}]$ que passa a ser local, $[\bar{K}_i]$, de acordo com o produto matemático. A variação de energia obtida na Eq. (2.16) é denominada número de sensibilidade, denotado por α_i , ou:

$$\alpha_i = \frac{1}{2} \{u_i\}^T [\bar{K}_i] \{u_i\}, \text{ com } i=1, n \quad (2.17)$$

Assim, a Eq. (2.17) indica a variação da energia de deformação devido à remoção do i -ésimo elemento ou, em outras palavras, ao se modificar a rigidez da estrutura. O valor dessa variação, ou do número de sensibilidade, é sempre positivo (CHU *et al.* 1996).

É importante perceber que a Eq. (2.17) se refere às restrições impostas à rigidez global da estrutura. Caso as restrições sejam impostas aos deslocamentos, a variação de deslocamentos Δu deverá ser reescrita. Faz-se necessário a introdução de um vetor unitário de forças $\{P_j\}$ a fim de encontrar a mudança em u_j devido à remoção do elemento, sendo u_j a restrição imposta ao j -ésimo componente do vetor de deslocamentos. Ao aplicar esse carregamento unitário, apenas o j -ésimo componente do vetor de forças possui valor unitário, sendo os outros valores nulos.

Multiplicando-se a Eq. (2.15) pelo vetor unitário $\{P_j\}^T$, tem-se:

$$\Delta u_j = \{P_j\}^T \{\Delta u\} = -\{P_j\}^T [\bar{K}]^{-1} [\Delta \bar{K}] \{u\} = -\{u_j\}^T [\Delta \bar{K}] \{u\} \quad (2.18)$$

Logo,

$$\Delta u_j = \{u_{ij}\}^T [\bar{K}_i] \{u_i\} \quad (2.19)$$

em que $\{u_j\}$ o vetor de deslocamentos, resultante da aplicação do vetor de forças unitárias; $\{u_i\}$ e $\{u_{ij}\}$ são os vetores de deslocamentos relacionados ao i -ésimo elemento e, sendo o último, combinado com o vetor de deslocamentos $\{u_j\}$. Assim, a variação de deslocamento obtida na Eq. (2.19) é denominada número de sensibilidade, denotado por α_i , ou:

$$\alpha_i = |\alpha_{ij}| = \left| \{u_{ij}\}^T [\bar{K}_i] \{u_i\} \right|, \text{ com } i=1, n \quad (2.20)$$

Finalmente, a Eq. (2.20) indica a variação dos deslocamentos devido à remoção do i -ésimo elemento. Ao contrário da Eq. (2.17), o valor de α_{ij} pode ser positivo ou negativo, o que significa que u_j pode variar sua direção (CHU *et al.* 1996). Ainda, é importante ressaltar que $[\bar{K}]$ pode ser substituído por uma matriz de rigidez equivalente que inclua os efeitos inerciais em problemas dinâmicos. Isso faz com que o valor do número de sensibilidade seja diferente do valor do *compliance*, descrito a seguir. Para o caso elástico, esses valores são iguais.

2.8.3 Compliance

Como mencionado no t3pico anterior, a flexibilidade de algumas t3cnicas de otimiza33o, como a ESO e a SESO, permite a ado33o de diferentes crit3rios de remo33o. Um terceiro crit3rio que vale a pena ser citado 3 o *compliance* que, de acordo com Gournay, Allaire e Jouve (2008), 3 definido como o trabalho total realizado pela for3a aplicada ou, de modo equivalente, a soma da energia de deforma33o e da energia de deforma33o complementar de um dado modelo estrutural. 3 uma medida conveniente de rigidez global de uma estrutura.

A rigidez da estrutura est3 ligada ao deslocamento sofrido pela estrutura sob a a33o de um carregamento qualquer. Quanto maior a rigidez menor ser3 o deslocamento da estrutura e, portanto, o *compliance* ser3 m3nimo (GEA e LUO, 2001). Por defini33o, seu valor 3 expresso por:

$$W = \int_{^0V} f_i^B u_i d^0V + \int_{^0S_f} f_i^S u_i^S d^0S \quad (2.21)$$

em que f^B s3o as for3as de corpo, u s3o os deslocamentos, 0V o volume inicial e f^S e u^S s3o as for3as atuantes e os deslocamentos avaliados na superf3cie 0S_f , respectivamente.

Como W representa tamb3m o trabalho realizado pelas for3as externas, ao se considerar as for3as permanentes aplicadas na estrutura (peso pr3prio, por exemplo), esse trabalho W pode ser expresso por:

$$W = U + U^C \quad (2.22)$$

em que U e U^C a energia de deforma33o e sua energia de deforma33o complementar, respectivamente. A energia de deforma33o pode ser definida por:

$$U = \int_{^0V} \int S_{ij} d\varepsilon_{ij} d^0V \rightarrow U = \frac{1}{2} \int_{^0V} S_{ij} \varepsilon_{ij} d^0V, \text{ com } S_{ij} = C_{ijkl} \varepsilon_{kl} \quad (2.23)$$

onde S_{ij} s3o as componentes cartesianas do segundo tensor de tens3o de Piola-Kirchhoff¹, C_{ijkl}

¹ O tensor de tens3o de Piola-Kirchhoff s3o tensores usados na teoria da elasticidade com deforma33es finitas

são componentes do tensor constante de elasticidade e ε_{ij} são componentes do tensor de deformação de Green-Lagrange², sendo o último definido por:

$$\varepsilon_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial^0 x_j} + \frac{\partial u_j}{\partial^0 x_i} + \frac{\partial u_k}{\partial^0 x_i} \frac{\partial u_k}{\partial^0 x_j} \right) \quad (2.24)$$

É importante observar que a relação elástica constitutiva de S_{ij} é válida para deformações suficientemente pequenas (até 0,2%, por exemplo).

Para derivar a energia complementar U^c acima mencionada é necessário reescrever W em função das componentes de tensão e deformação. Ao se considerar o estado de equilíbrio e utilizando a Eq. (2.21) além da Eq. (2.24), tem-se:

$$W = \int_{^0V} S_{ij} \left(\varepsilon_{ij} + \frac{1}{2} \frac{\partial u_k}{\partial^0 x_i} \frac{\partial u_k}{\partial^0 x_j} \right) d^0V \quad (2.25)$$

Finalmente, ao subtrair a Eq. (2.23) da Eq. (2.25), a energia U^c pode ser expressa por:

$$U^c = U + \frac{1}{2} \int_{^0V} S_{ij} \frac{\partial u_k}{\partial^0 x_i} \frac{\partial u_k}{\partial^0 x_j} d^0V \quad (2.26)$$

Portanto, de acordo com Gea e Luo (2001), a energia de deformação U não é igual à sua energia complementar U^c em estruturas com comportamento não linear mesmo se o material for elástico. Nas análises onde o comportamento é linear, a energia de deformação é igual à sua energia complementar e, por isso, o *compliance* possui valor igual a duas vezes a energia de deformação, tornando inadequado escolher o *compliance* ou energia de deformação como função objetivo no caso de projetos de otimização de rigidez. É importante destacar que o número de sensibilidade pode ser aplicado apenas para problemas com comportamento linear, enquanto a aplicação do *compliance* é mais geral.

Uma desvantagem do *compliance* reside no fato da energia complementar, U^c , nem sempre

para representar a tensão referente à configuração inicial não deformada, ao contrário dos tensores de tensão de Cauchy, usualmente usados para representar as tensões da configuração deformada.

2 O tensor de deformação de Green-Lagrange são usados em mecânica de meios contínuos para caracterizar a alteração de forma e volume de um corpo. Ambos tensores constituem pares energéticos conjugados.

poder ser determinada, dependendo do tipo de material. Considerando o comportamento do material da Fig. 2.15³ e a definição de energia complementar ao longo deste tópico, o trecho após a tensão última não permite o cálculo de U^c de forma direta.

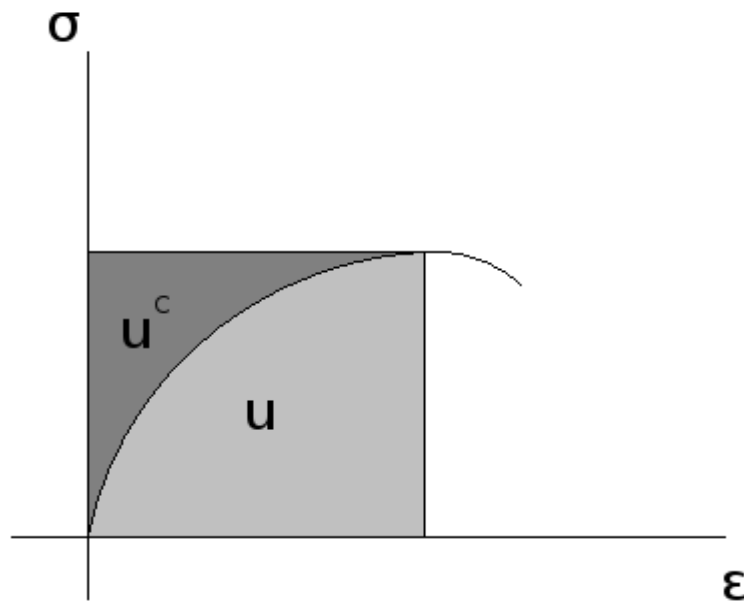


Figura 2.15 – Energia de deformação U e energia de deformação complementar U^c

Por isso, a utilização do *compliance* como critério de remoção para a técnica heurística de otimização SESO torna-se limitada até o ponto onde a energia complementar pode ser determinada, motivo o qual para esta Tese foi adotado o critério de remoção de tensão de von Mises associado ao SESO.

3 Os parâmetros u e u^c do gráfico referem-se, respectivamente, às energias de deformação e energia de deformação complementar por unidade de volume.

3

O MÉTODO DOS ELEMENTOS FINITOS APLICADO À OTIMIZAÇÃO TOPOLÓGICA

Este capítulo se destina a tratar de uma abordagem do Método dos Elementos Finitos, assim como de Análise Não Linear Física e Geométrica, especificamente aplicado à otimização topológica evolucionária, enfatizando as aplicações encontradas na literatura. Além disso, os elementos utilizados nas análises aqui realizadas são brevemente citados e explicados.

3.1 O Método dos Elementos Finitos e a Otimização Estrutural Evolucionária

Vários problemas de engenharia são regidos por equações diferenciais. De forma geral, os problemas de engenharia são baseados em modelos matemáticos que descrevem, de maneira precisa ou aproximada, situações físicas. Dessa maneira, estes modelos matemáticos são baseados em equações diferenciais cujos parâmetros prévios e condições iniciais e de contorno são preestabelecidos. Estas equações são provenientes das leis fundamentais e princípios da natureza, ou seja, representadas por grandezas relacionadas com massa, força e energia. Quando possível, a solução exata dessas equações fornece detalhes de um comportamento do sistema, levando-se em conta as condições de contorno e os parâmetros iniciais (MOAVENI, 2003). O Método dos Elementos Finitos (MEF) é um procedimento numérico para a obtenção de respostas para os vários problemas encontrados em análises da engenharia. Combina vários conceitos matemáticos para produzir um sistema linear ou não linear de equações. Mas para resolvê-los são necessários métodos ou técnicas que sejam eficazes e viáveis computacionalmente, uma vez que o número de equações geralmente é

muito elevado, o que inviabiliza ou torna pouco prático para resoluções não computacionais (SEGERLIND, 1984).

De forma sucinta, a ideia do método é discretizar um domínio qualquer em uma malha de elementos e calcular a influência de cada um desses elementos nessa malha, de acordo com a contribuição deles. O comportamento de cada elemento é estabelecido usando-se funções aproximadoras que descrevem os deslocamentos desses elementos. Em geral, são usados polinômios para essa representação. A partir destas aproximações iniciais, calculam-se outras grandezas, como deformações, tensões, energia e forças. Cada tipo de elemento, seja linear, plano ou tridimensional, possui funções de forma que são utilizadas na aproximação cinemática.

A formulação clássica do Método dos Elementos Finitos pode ser deduzida a partir do princípio dos trabalhos virtuais (PTV), exigindo que, para o estado de equilíbrio, o trabalho da força aplicada é nulo para qualquer variação infinitesimal da configuração do sistema, que está em harmonia com as dadas restrições cinemáticas. Ainda, no equilíbrio, a energia potencial deve ser estacionária em relação a todas as variações cinematicamente admissíveis (LANCZOS, 1952). Assim, pode-se afirmar que o trabalho virtual externo é igual ao trabalho virtual interno (energia de deformação interna virtual), ou matematicamente falando:

$$\sum_{i=1}^n F_R \cdot \delta u = \int_V (\sigma \cdot \delta \epsilon) dV \quad (3.4)$$

em que F_R representa as forças reais, δu os deslocamentos virtuais, σ as tensões e $\delta \epsilon$ as deformações virtuais.

Para o caso da medida de deformação de engenharia, a componente de deformação normal no sentido do eixo da barra pode ser expressa por $\epsilon = \frac{du}{dx}$. Se u representa os deslocamentos, ao se considerar a Eq. (3.3) (derivando-se a matriz que representa as funções de forma), tem-se:

$$\varepsilon = \frac{du}{dx} = \frac{d\phi}{dx} \{u^e\} = \begin{bmatrix} -\frac{1}{L} & \frac{1}{L} \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} = [B] \{u^e\} \quad (3.5)$$

onde $[B]$ relaciona os deslocamentos e deformações do elemento, sendo diferente para cada mapeamento e/ou elemento finito considerado, e $\{u^e\}$ são os deslocamentos do elemento no sistema local. Usando a relação de tensão por deformação dada pela Lei de Hooke, tem-se:

$$\sigma = E \varepsilon = E [B] \{u^e\} \quad (3.6)$$

Aqui é necessário mudar as coordenadas dos vetores e matrizes pertinentes para o sistema global caso não coincidam com o sistema local. Para isso, matrizes de transformação devem ser aplicadas, se necessário. A aplicação do PTV significa impor uma variação de deformações e deslocamentos, ou seja, o trabalho externo é gerado por uma variação de deslocamentos δu^e e o trabalho interno é gerado por uma variação de deformação $\delta \varepsilon$ (BATHE, 1996). Portanto, reescrevendo a Eq. (3.4) ao substituir o valor de σ e ε descrito na Eq. (3.5) (se $\varepsilon = [B] \{u^e\}$, $\delta \varepsilon = [B] \{\delta u^e\}$ ou $\delta \varepsilon^T = \{\delta u^e\}^T [B]^T$), tem-se:

$$\{F^e\} \{\delta u^e\} = \int_V (\delta \varepsilon \cdot \sigma) dV \rightarrow \{\delta u^e\}^T \{F^e\} = \int_V \{\delta u^e\}^T [B]^T E [B] \{u^e\} dV \quad (3.7)$$

Como $\{\delta u^e\}^T$ é uma constante por serem valores de deslocamentos impostos nas extremidades, a Eq. (3.7) pode ser reescrita da seguinte forma:

$$\{F^e\} = \left(\int_V [B]^T E [B] dV \right) \{u^e\} \quad \text{ou} \quad \{F^e\} = [k^e] \{u^e\} \quad (3.8)$$

sendo $\{F^e\}$ o vetor que representa as forças nodais aplicadas, $[k^e]$ a matriz de rigidez e $\{u^e\}$ os deslocamentos do elemento finito considerado. Esta é a formulação clássica do Método dos Elementos Finitos para elementos estáticos e lineares e, para um melhor aprofundamento, esse assunto é muito bem abordado por Bathe (1996).

Com base nas informações descritas e traçando-se um paralelo com o contexto deste trabalho, o Método dos Elementos Finitos é um método numérico natural para o uso na otimização

topológica em função da característica do método, baseado em parâmetros de domínio, o que facilita o desenvolvimento de técnicas de otimização associados aos valores de densidade de material de cada ponto deste domínio (ALMEIDA, 2009). Por isso, é amplamente utilizado na otimização de estruturas. Pelo fato do Método dos Elementos Finitos ser de fácil implementação frente a outros métodos e devido à existência de vários *softwares* disponíveis atualmente (comerciais e de código livre), a sua utilização se torna amplamente perceptível nos vários trabalhos existentes na literatura.

O processo de otimização em conjunto com elementos finitos é abordado de forma iterativa, onde a cada iteração uma porcentagem de elementos é removido, de acordo com a técnica de otimização utilizada, até que não haja mais a remoção de elementos e a topologia ótima seja finalmente alcançada. Sendo a análise linear ou não linear, o processo iterativo de otimização ocorre de similar forma, de acordo com os resultados parciais das malhas geradas. Tal processo pode ser resumido em um fluxograma de atividades de acordo com a Fig. 3.1:

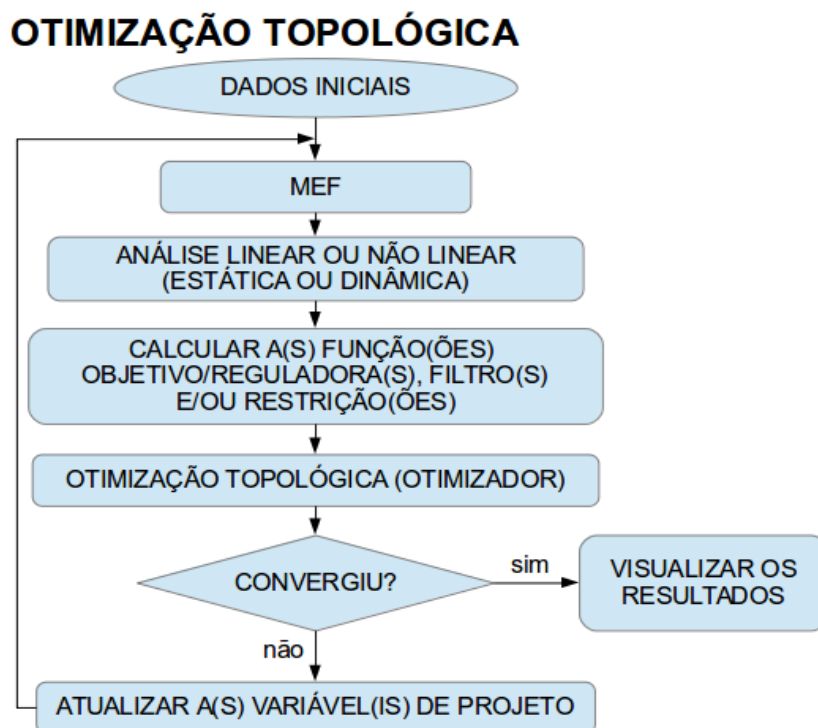


Figura 3.1 – Fluxograma de atividades para otimização topológica

O fluxograma descreve que, partindo dos dados iniciais, a malha é gerada para que a análise

linear ou não linear seja processada. É importante aqui ressaltar que como não há convergência em uma única iteração de otimização, uma vez que a análise é sensível aos parâmetros de otimização definidos inicialmente e que uma brusca remoção provoca um grande aumento da tensão máxima da estrutura que pode gerar instabilidade numérica, as funções que regem o problema devem ser calculadas antes da remoção de elementos, com as devidas restrições, se existirem. O processo de remoção de elementos pelo otimizador é, então, iniciado. Enquanto não houver a convergência, os parâmetros de entrada relacionados à otimização são atualizados e o processo iterativo é recommençado até a obtenção da esperada topologia ótima final. Para o caso não linear, pode haver aplicação de vários passos de força ou deslocamentos e em cada um destes passos pode haver várias iterações. O mesmo ocorre para o caso dinâmico, mas em termos de aplicação de passos de tempo para a análise transiente.

3.1.1 Elementos Finitos principais do Code Aster®

O *software* livre Code-Aster®, programa de elementos finitos utilizado para o cálculo neste trabalho e descrito no próximo capítulo, possui uma vasta biblioteca de elementos que podem ser utilizados. Serão descritos aqui os principais elementos finitos do programa, alguns dos quais serão usados neste trabalho.

O programa reconhece os nós da malha através de um identificador de nome POI1, de acordo com a Fig. 3.2.

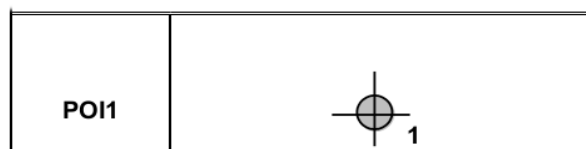


Figura 3.2 – Nó associado a uma malha (2D ou 3D)

Os elementos finitos lineares pertencentes à biblioteca contêm 2, 3 ou 4 nós, sendo chamados de SEG2, SEG3 e SEG4, respectivamente, conforme Fig. 3.3.

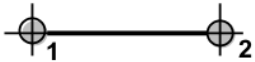
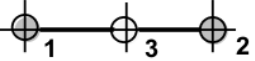
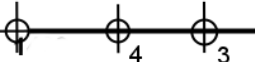
SEG2	
SEG3	
SEG4	

Figura 3.3 – Elementos lineares (segmentos) associados a uma malha (2D ou 3D)

O sentido de mapeamento dos nós dos elementos lineares (segmentos) é representado como na Fig. 3.4.

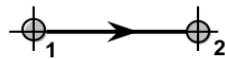


Figura 3.4 – Sentido de mapeamento dos elementos lineares

Com relação aos elementos planos triangulares de 3, 6 e 7 nós, cujos nomes são TRIA3, TRIA6 e TRIA7, respectivamente, os nós são definidos conforme Fig. 3.5.

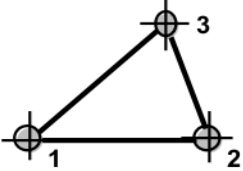
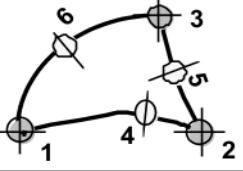
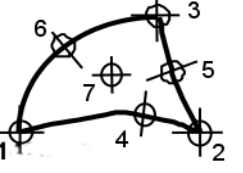
TRIA3	
TRIA6	
TRIA7	

Figura 3.5 – Elementos triangulares associados a uma malha (2D ou 3D)

O sentido de mapeamento dos nós dos elementos triangulares é anti-horário, conforme Fig. 3.6.

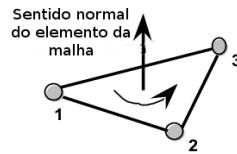


Figura 3.6 – Sentido de mapeamento dos elementos triangulares

Com relação aos elementos planos quadrilaterais de 4, 8 e 9 nós, cujos nomes são QUAD4, QUAD8 e QUAD9, respectivamente, os nós são definidos conforme Fig. 3.7.

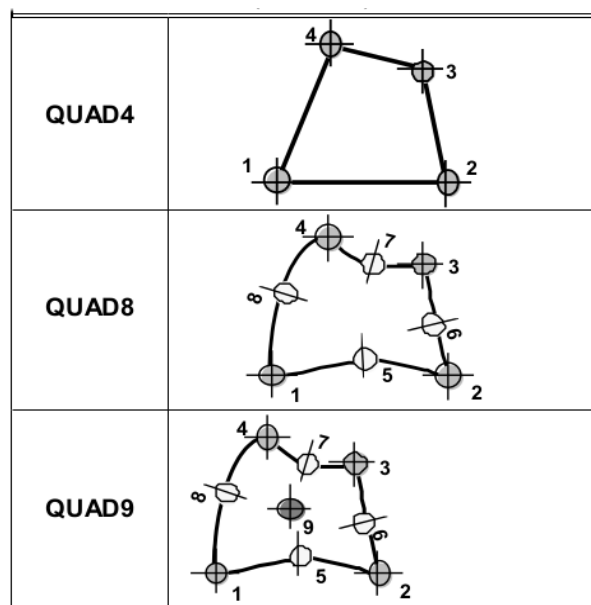


Figura 3.7 – Elementos quadrilaterais associados a uma malha (2D ou 3D)

O sentido de mapeamento dos nós dos elementos quadrilaterais é anti-horário, conforme Fig. 3.8.

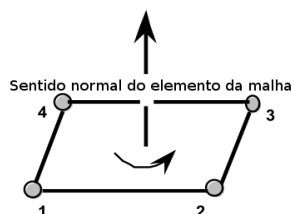


Figura 3.8 – Sentido de mapeamento dos elementos quadrilaterais

Com relação aos elementos sólidos tetraédricos de 4 e 10 nós, cujos nomes são TETRA4 e TETRA10, respectivamente, os nós são definidos conforme Fig. 3.9.

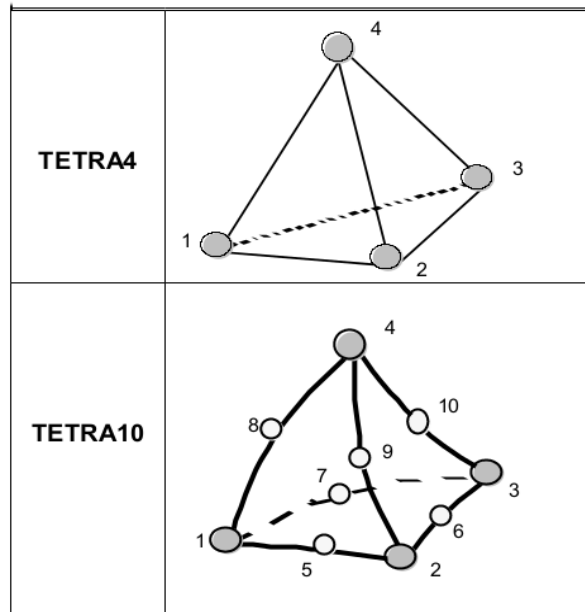


Figura 3.9 – Elementos tetraédricos associados a uma malha (3D)

Com relação aos elementos sólidos piramidais de 5 e 13 nós, cujos nomes são PYRAM5 e PYRAM13, respectivamente, os nós são definidos conforme Fig. 3.10.

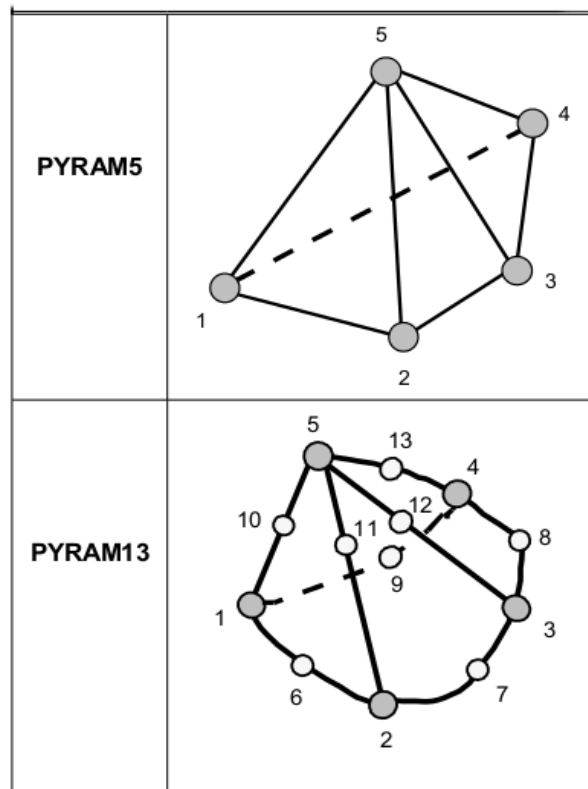


Figura 3.10 – Elementos piramidais associados a uma malha (3D)

Com relação aos elementos sólidos “pentaédricos” de 6, 15 e 18 nós, cujos nomes são PENTA6, PENTA15 e PENTA18, respectivamente, os nós são definidos conforme Fig. 3.11. Este tipo de elemento é usado com frequência em elementos sólidos curvos tridimensionais para fechar a geometria da malha.

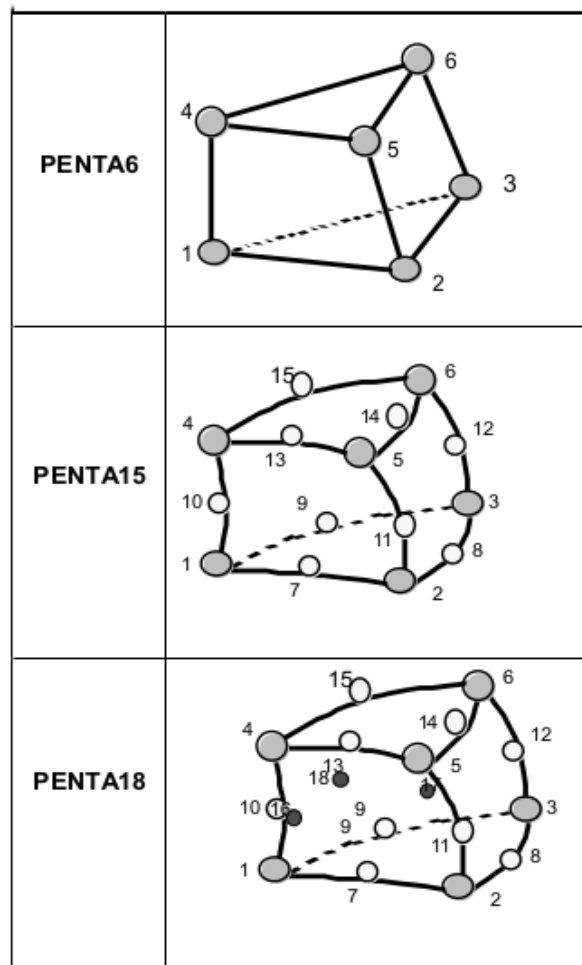


Figura 3.11 – Elementos pentaédricos associados a uma malha (3D)

Por fim, o programa disponibiliza ainda aos elementos sólidos hexaédricos de 8, 20 e 27 nós, cujos nomes são HEXA8, HEXA20 e HEXA27, cujos nós são definidos conforme Fig. 3.12.

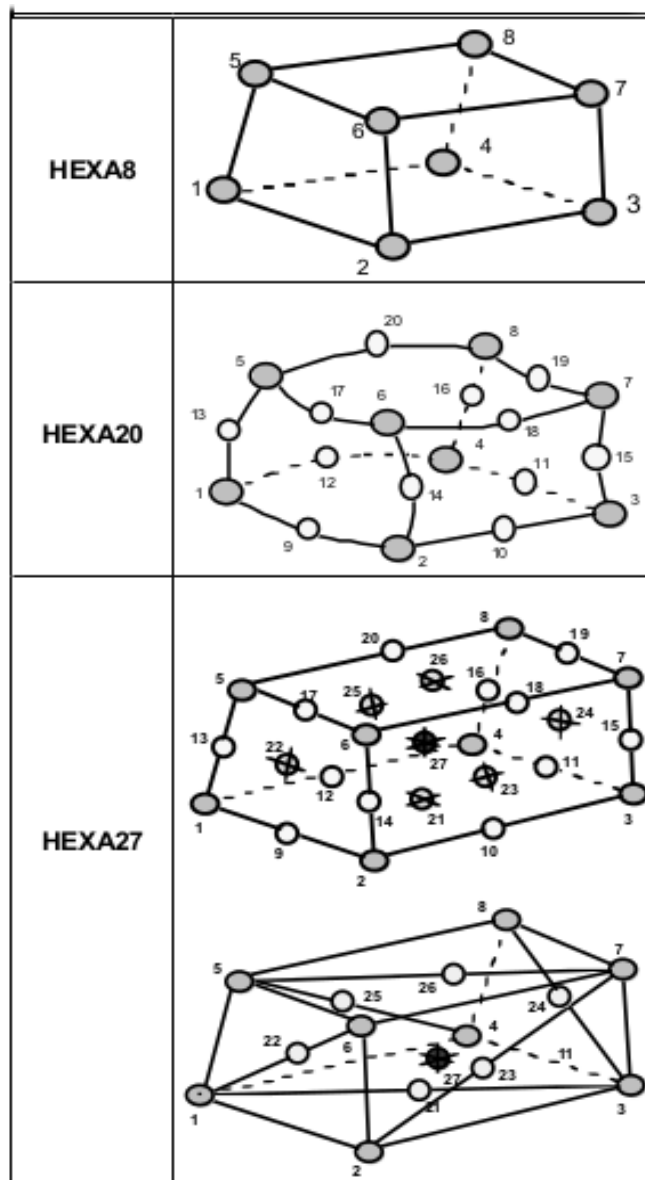


Figura 3.12 – Elementos hexaédricos associados a uma malha (3D)

É importante mencionar que se utiliza ainda um outro *software* livre, Salome-Meca[®], que conta com um gerador automático de malha, utilizando uma combinação de alguns desses elementos. Ambas malhas, bidimensionais e tridimensionais, podem ser feitas a partir desse gerador, facilitando a criação de malhas rápidas para as análises que serão realizadas. Nesta tese encontram-se informações no apêndice B sobre o *software* livre Salome-Meca[®] e dos demais programas utilizados.

3.2 Análise não Linear Geométrica e a Otimização Estrutural

Ao se considerar imperfeições ou alterações geométricas que influenciam no equilíbrio nas análises estruturais, já não existe mais proporcionalidade entre deslocamentos e forças. Essa alteração da geometria, comparada a uma configuração inicial preestabelecida, pode ser devida a um erro de confecção da peça estrutural, por um erro de fixação do material de acordo com sua finalidade ou devido a deslocamentos e rotações da configuração original, provocados por um carregamento qualquer. Também, ao se considerar que a estrutura sofre grandes deslocamentos (ou deslocamentos significativos em relação à configuração inicial) de acordo com a carga aplicada, podem surgir problemas de instabilidade da estrutura e, dessa forma, os resultados oriundos de uma análise linear não serão adequados para descrever o modelo analisado. No que tange à otimização, durante o processo de obtenção da topologia ótima, a estrutura tende a se tornar mais esbelta, aumentando a possibilidade de ocorrência de comportamento não linear, tornando necessário o estudo da não linearidade.

A não linearidade considerada em uma estrutura pode ser classificada em não linearidade geométrica, que representa o efeito de grandes deslocamentos acentuados da configuração geométrica no equilíbrio estrutural, em não linearidade física, na qual o comportamento do material é não linear e em não linearidade de contorno, ou seja, os deslocamentos dependem das condições de contorno (este tipo de não linearidade é muito encontrado em problemas de contato e impacto). Além disso, ao se considerar uma não linearidade na análise de um modelo, o princípio da superposição dos efeitos não pode ser mais aplicado, além de que é importante considerar também o estado inicial de tensões (tensões residuais, por exemplo) na análise (IVANCO, 2011).

Matematicamente falando, ao se considerar que os deslocamentos sofridos pela estrutura são suficientemente pequenos, que o material é linearmente elástico e que não há mudança nas condições de contorno durante a aplicação das cargas, as equações de equilíbrio do sistema em uma análise estática podem ser escritas como:

$$K \cdot u = F \quad (3.9)$$

onde K representa a matriz de rigidez do modelo estrutural, u o vetor de deslocamentos e F o vetor de forças externas aplicadas.

A Eq. (3.9) corresponde a um sistema linear pois a resposta dos deslocamentos é uma função linear do vetor de forças aplicadas F . A linearidade do modelo é comprovada quando a matriz que relaciona os deslocamentos e deformações de cada elemento, derivada da matriz das funções de forma (matriz B descrita no tópico anterior), possui valores constantes e não dependentes dos deslocamentos dos elementos nem das forças de equilíbrio. Além disso, ainda é necessário que a matriz constitutiva do elemento (matriz E , descrita no tópico anterior, e que representa o Módulo de Elasticidade do elemento) também possua valores constantes e independentes. Caso contrário, uma análise não linear deverá ser feita (BATHE, 1996).

Em uma análise não linear o problema é, basicamente, encontrar o estado de equilíbrio de um corpo em resposta às cargas aplicadas, de acordo com a Eq. (3.10).

$$\begin{aligned} R - F &= 0 \\ ({}^tR - {}^tF) &= 0 \end{aligned} \quad (3.10)$$

em que R representa as forças nodais externas aplicadas e F as forças nodais internas que correspondem às tensões do elemento na configuração atual. Aqui é importante ressaltar que ambos vetores podem ou não ser dependentes do tempo (tR e tF), o que caracteriza uma análise dinâmica, quase-estática¹ ou estática, respectivamente, de acordo com o interesse e objetivo da análise. É importante ressaltar que a matriz de rigidez da estrutura, K , passa a ser dependente do vetor de deslocamentos u e, dessa forma, a Eq. (3.9) pode ser reescrita da seguinte forma:

$$K(u) \cdot u = F \quad (3.11)$$

A Eq. (3.11) é válida para o caso de forças conservativas, cujo trabalho não depende das trajetórias para se passar de uma configuração inicial para deformada. Ou seja o vetor das forças não depende dos deslocamentos. Há problemas mais complexos, nos quais as forças não são conservativas, $F = F(u)$, como no caso de problemas de acoplamento fluido-

¹ Relevante em análises de materiais com comportamento viscoelástico, por exemplo.

estrutura envolvendo consideráveis deslocamentos estruturais.

Considerando uma análise estática e combinando a Eq. (3.10) com a (3.11), tem-se:

$$K(u) \cdot u - F \simeq 0 = g(u) \quad (3.12)$$

onde $g(u)$ é o chamado vetor dos resíduos, que no equilíbrio estrutural ideal seria nulo.

Para a resolução de problemas não lineares, métodos iterativos são utilizados para a resolução desses sistemas, como o método de Newton-Raphson, sendo este um método iterativo para o cálculo aproximado de raízes de equações baseado na linearização tangente da equação considerada por meio da expansão de Taylor, ou seja, para este caso, encontrar uma raiz para a Eq. (3.12) a partir de uma previsão inicial u_0 usando a expansão de Taylor.

$$g(u_n) \simeq g(u_0) + \frac{dg(u_0)}{du} \delta u + \left(\frac{1}{2} \frac{d^2g(u_0)}{du^2} (\delta u)^2 \right) + \dots \quad (3.13)$$

Logo, dada uma estimativa inicial u_0 , ao qual $g_0(u_0) \neq 0$, e desconsiderando os termos de alta ordem da Eq. (3.13) por serem valores muito pequenos (a partir da segunda ordem), a Fig. 3.13 pode então representar, de forma resumida, o método de Newton-Raphson tangente.

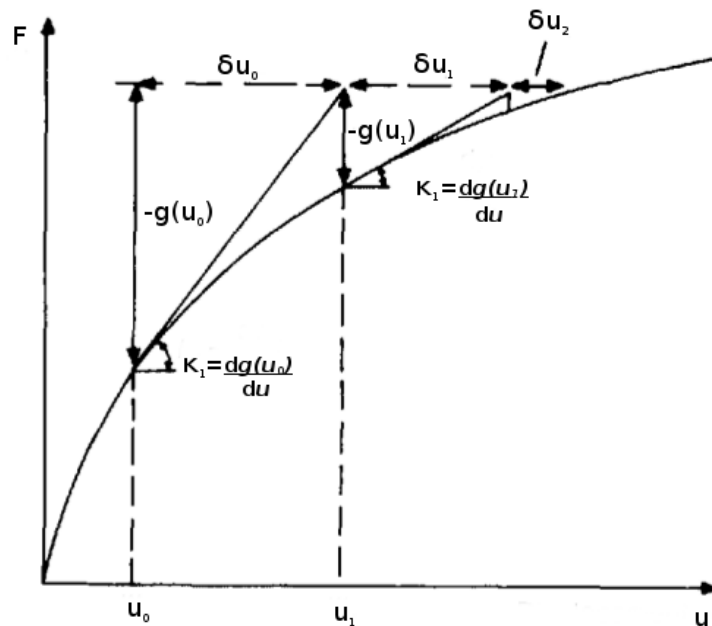


Figura 3.13 – Método de Newton-Raphson. Fonte: Crisfield (1997)

De forma sucinta, o primeiro passo do método consiste em calcular o resíduo e a derivada da função para a previsão inicial. Na sequência, calcula-se a correção δu e atualiza-se a previsão, repetindo esse procedimento até que o módulo do resíduo seja menor do que uma certa tolerância numérica previamente definida.

Outros métodos iterativos e/ou incrementais podem ser utilizados para a realização de uma análise não linear. Para um melhor aprofundamento do método de Newton-Raphson e de outros desses métodos, esse assunto é muito bem abordado por Crisfield (1997).

No que diz respeito à otimização, as formulações implementadas e utilizadas nas análises trabalham em conjunto com os métodos de remoção de elementos. Sendo a análise linear ou não linear, a remoção de elementos pela otimização ocorre de forma igual e gradual. De acordo com Yamada *et al.* (2013), a otimização topológica tem sido amplamente aplicada em uma variedade de problemas de otimização estrutural que lidam principalmente com materiais lineares que não são submetidos a grandes deformações. Contudo, a otimização topológica de estruturas recentemente tem sido aplicada a problemas que incorporam materiais não lineares e grandes deformações (GEA e LUO, 2001). Para resolver problemas de otimização de estruturas geometricamente não lineares, métodos baseados em alterações da densidade são sugeridos.

Logo, Gomes e Senne (2013) propõem a utilização da técnica de otimização SIMP (*Solid Isotropic Microstructures with Penalization*) para a remoção de elementos, cujo objetivo pode ser escrito como:

$$\begin{aligned} \min_{\rho} \quad & f^T K(\rho)^{-1} f \\ \text{sujeito a} \quad & \sum_{i=1}^{n_{el}} \frac{v_i \rho_i}{V^*} - 1 \leq 0 \\ & 0 < \rho_{min} \leq \rho_i \leq 1 \end{aligned} \quad (3.14)$$

com $(i = 1, \dots, n_{el})$. O parâmetro n_{el} representa o número de elementos do domínio, ρ_i e v_i a densidade e o volume do i -ésimo elemento, respectivamente, V^* o volume máximo da estrutura, f o vetor de forças nodais associado ao carregamento externo e $K(\rho)$ a matriz de rigidez global do sistema. O que muda aqui em relação à técnica SIMP descrita anteriormente é que a densidade é limitada a um valor mínimo (ρ_{min}) maior que zero, a fim de tornar a matriz $K(\rho)$ simétrica e positiva definida. O vetor dos deslocamentos, dessa forma, pode ser escrito por:

$$u = K(\rho)^{-1} f \quad (3.15)$$

As Eqs. (3.14) e (3.15) ainda estão no regime linear, uma vez que a rigidez só depende da densidade, que é considerada constante. Logo, é necessário reescrever a Eq. (3.15) em um sistema não linear, com K dependente de u e ρ . Como ρ é considerado constante, os deslocamentos podem ser facilmente obtidos pelo método de Newton-Raphson, sendo o sistema reescrito da forma

$$K_T(u_k, \rho) \Delta u_k = -R(u_k, \rho) \quad (3.16)$$

em que $K_T = \partial R / \partial u$ representa a matriz de rigidez cuja definição, assim como a de $R(u_k, \rho)$, varia de acordo com o modelo elástico adotado para o material (GOMES e SENNE, 2013). Assim, de acordo com um dado vetor inicial u_0 o método resolve o sistema de equações representado pela Eq. (3.16) e atualiza os deslocamentos usando $u_{k+1} = u_k + \Delta u_k$ até

que uma condição predefinida do tipo $\|R(u_k, \rho)\| \leq \varepsilon$ seja satisfeita, considerando uma tolerância numérica $\varepsilon > 0$.

De forma resumida, a ideia de Gomes e Senne (2013) é penalizar a densidade dos elementos. O sistema não linear é calculado pelo método de Newton-Raphson e, após todas as iterações pertinentes à não linearidade geométrica serem realizadas, os elementos são, então, devidamente penalizados. O mais importante aqui está justamente no fato da penalização dos elementos ser realizada apenas após todo o sistema não linear ser resolvido.

3.3 Análise não Linear Física e a Otimização Estrutural

A maioria dos materiais de engenharia exibe uma relação linear entre tensão e deformação dentro da região elástica. Por consequência, um aumento na tensão provoca um aumento proporcional na deformação, fato descoberto por Robert Hooke em 1676, sendo conhecido como Lei de Hooke (HIBBELER, 2010). Assim, a Lei de Hooke é expressa matematicamente por

$$\sigma = E \varepsilon \quad (3.17)$$

onde σ representa a tensão, ε é a deformação e E é a constante de proporcionalidade denominada Módulo de Elasticidade.

O fato é que à medida que o nível de tensão no material for aumentando, em um determinado ponto as deformações passam a não ter mais a proporcionalidade de aumento em relação à tensão. Esse comportamento caracteriza uma não linearidade física, e a Lei de Hooke descrita acima passa a não ser mais válida. Além disso, o material atinge o chamado regime plástico. O regime de comportamento plástico de um material fica evidenciado pelo aparecimento de deformações irreversíveis, ou permanentes, quando se anula a solicitação a que o corpo esteja sujeito. Por outro lado, as deformações plásticas aparecem simultaneamente com as tensões que excedem um determinado limite, denominado limite de escoamento (PROENÇA, 1988).

Ao se considerar material isotrópico, segundo Pimenta (2006), a equação característica das tensões principais considerando que a matriz das componentes do tensor das tensões em uma base ortonormal seja dada por $[\sigma] = \sigma_{ij}$ pode ser descrita por:

$$\sigma^3 - I_1 \sigma^2 + I_2 \sigma - I_3 = 0 \quad (3.18)$$

com $I_i, i = 1,2,3$, são invariantes de tensão dados por:

$$\begin{aligned} I_1 &= \sigma_{11} + \sigma_{22} + \sigma_{33} \\ I_2 &= \sigma_{11} \sigma_{22} + \sigma_{22} \sigma_{33} + \sigma_{33} \sigma_{11} - \sigma_{12}^2 - \sigma_{23}^2 - \sigma_{31}^2 \\ I_3 &= \sigma_{11} \sigma_{22} \sigma_{33} + 2 \sigma_{12} \sigma_{23} \sigma_{31} - \sigma_{11} \sigma_{23}^2 - \sigma_{22} \sigma_{31}^2 - \sigma_{33} \sigma_{12}^2 \end{aligned} \quad (3.19)$$

Os invariantes do tensor das tensões independem da base onde o tensor σ é representado.

Com isso, um critério de resistência isotrópico pode ser expresso da seguinte forma:

$$F(\sigma) = F(I_1, I_2, I_3) \quad (3.20)$$

com $I_i, i = 1,2,3$, os mesmos invariantes principais do tensor das tensões, segundo a Eq. (3.19).

A superfície de resistência associada a um critério de resistência isotrópico é a superfície no espaço R^3 dada pela equação

$$F(\sigma_1, \sigma_2, \sigma_3) = 0 \quad (3.21)$$

onde $\sigma_1, \sigma_2, \sigma_3$ são as tensões principais.

A adoção de um critério de tensão para avaliar o escoamento é necessário para uma análise elastoplástica de estruturas, correspondendo a relações matemáticas entre as componentes de tensão, atuantes em um corpo analisado. Como exemplos mais comuns de critérios de ruptura, tem-se o critério de Mohr-Coulomb, o critério de Drucker-Prager e o critério de von Mises (PIMENTA, 2006).

Além dos critérios de tensão, cada material possui um comportamento característico face às solicitações. Por isso, é necessário enunciar as relações constitutivas, específicas para cada material do qual a estrutura seja feita. Para os materiais dúcteis existem duas “fases”. A primeira correspondente à fase elástico linear do comportamento até que se atinja o limite de escoamento. A segunda relativa ao comportamento plástico a partir de então, diferentemente do que ocorre com o modelo elástico linear onde não se caracteriza a plastificação, a relação constitutiva elastoplástica é dependente do nível de deformação plástica acumulada no carregamento, ou seja, os níveis de tensão e deformação são relacionados em função do nível de deformação plástica atingido (OLIVEIRA, 2001).

Assim, para que o comportamento plástico dos materiais possa ser descrito corretamente, o efeito do modo de deformação (trajetória de deformação) precisa ser considerado, (RAUCH 2000; THUILLIER *et al.*, 2010). Com isso, de modo a facilitar a compreensão dos fenômenos de encruamento², as relações constitutivas (considerando um comportamento elastoplástico uniaxial linearizado) podem ser divididas, segundo Oliveira (1997), em relação elastoplástica perfeita, relação elastoplástica com encruamento cinemático e relação elastoplástica com encruamento isotrópico, sendo a última utilizada neste trabalho.

Na relação elastoplástica perfeita, o material apresenta um comportamento elástico linear até que se atinja a tensão de escoamento. A partir desse instante o material perde a capacidade resistente, e passa a deformar-se indefinidamente, mesmo sem acréscimos de tensões. Como exemplo, a consideração feita pela NBR 6118/14 para as barras de armadura utilizadas para o concreto armado sugere esse tipo de comportamento (Fig. 3.14).

2 O encruamento de um metal consiste no seu endurecimento por deformação plástica a frio. Essa condição de endurecimento do material pode ser representada a través do aumento do valor da tensão (σ) em função do aumento da quantidade de deformação plástica (ϵ) aplicada ao mesmo mediante o traçado de uma curva tensão-deformação (LOPES, 2009).

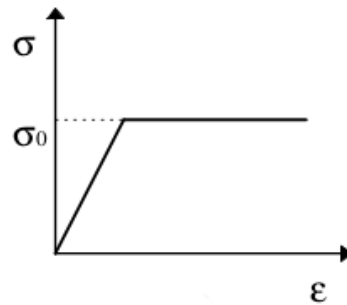


Figura 3.14 – Modelo elastoplástico perfeito

Já para as relações elastoplásticas com encruamento cinemático e com encruamento isotrópico, a Fig. 3.15 mostra ambos comportamentos distintos quanto à resistência ao escoamento de um material após a imposição de um carregamento com reversão no sentido da deformação. Em ambos os casos, durante o carregamento inicial, o material apresenta aumento da tensão com o aumento da deformação, o que caracteriza o encruamento (LOPES, 2009).

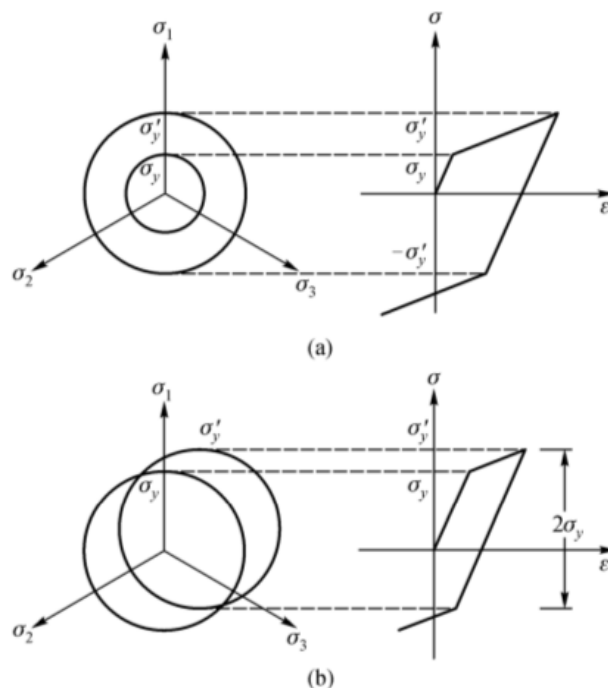


Figura 3.15 – Curvas tensão x deformação usadas para caracterizar o encruamento de um material dúctil: (a) encruamento isotrópico, (b) encruamento cinemático. Fonte: Li, Zhang e Tong (2014)

A Fig. 3.15 (a) exibe um metal que foi solicitado até um valor de tensão igual a σ_y' , após o escoamento ter ocorrido em σ_y . Em seguida, o sentido do carregamento é invertido imediatamente e o material passa a escoar sob a ação da tensão $-\sigma_y'$. Neste caso, o material deformado apresentará um aumento da resistência mecânica com a deformação plástica, classificando o encruamento como isotrópico (LOPES, 2009). A tangente do ângulo formado entre a reta que passa por σ_y , paralela ao eixo da deformação ϵ , e a reta que une os pontos σ_y e σ_y' determina o valor do Módulo de Encruamento Isotrópico, E_{ISO} .

Já a Fig. 3.15 (b), mostra que o encruamento ocasionará uma diminuição da resistência ao escoamento quando novamente submetido ao carregamento. Isso significa que os mecanismos que controlam a plasticidade dos dois carregamentos (inicial e reverso) são diferentes, classificando o encruamento como cinemático (HU *et al.*, 1992; WU, 2002).

De forma geral, a mudança da superfície inicial a partir da plastificação é regido pelo encruamento, que se divide em duas hipóteses básicas, encruamento isotrópico e encruamento cinemático, que caracterizam o comportamento da deformação de materiais dúcteis no regime plástico (CHABOCHE, 2008). O encruamento isotrópico prescreve uma mudança apenas no tamanho enquanto que o cinemático determina uma mudança somente na posição, ambos no regime plástico. Mas, na prática, o comportamento plástico da maioria dos materiais dúcteis segue uma combinação dessas duas regras para o encruamento (LI, ZHANG e TONG, 2014).

Na presente Tese foi adotado o modelo elastoplástico de Von Mises com encruamento isotrópico, caracterizando um comportamento não linear físico. O diagrama de tensão-deformação (σ , ϵ) deve ser fornecido ao programa Salome-Meca[®] de acordo com o material adotado para a análise. A resolução do sistema não linear segue o mesmo princípio da análise não linear geométrica, ou seja, através do Método de Newton-Raphson (descrito no tópico anterior) que atualiza e calcula a cada iteração a matriz de rigidez tangente do modelo.

Assim como na não linearidade geométrica, no que diz respeito à otimização, as formulações implementadas e utilizadas nas análises trabalham em conjunto com os métodos de remoção de elementos. Novamente, sendo a análise linear ou não linear, a remoção de elementos pela otimização ocorre de forma igual e gradual, ou seja, a aplicação da técnica de otimização

SESO em estruturas que apresentam não linearidade não difere em relação às estruturas lineares, sendo que a análise não linear de elementos finitos deve ser utilizada (HUANG e XIE, 2010). No entanto, o cálculo das tensões passa a depender dos modelos constitutivos utilizados para os materiais.

Jung e Gea (2004) sugere que, para efetivamente analisar estruturas submetidas a grandes deformações, a tensão e deformação efetivas sejam expressas em termos do 2º tensor de tensão de Piola-Kirchhoff e do tensor de deformação de Green-Lagrange. A Eq. (3.22) define a tensão efetiva como:

$$\varepsilon_e^2 = \varepsilon_{ij} C_{ijkl}^0 \varepsilon_{kl} \quad (3.22)$$

em que ε_e representa a deformação efetiva, ε_{ij} representa o tensor de deformação de Green-Lagrange e C_{ijkl} é uma matriz positiva definida e adimensional. As características não lineares do material são modeladas por uma relação geral entre a tensão e a deformação efetivas, definidas segundo a Eq. (3.23):

$$\sigma_e = K_E f(\varepsilon_e) \quad (3.23)$$

onde a função $f(\varepsilon_e)$ é uma função geral que representa as características do material e K_E é uma constante ligada ao Módulo de Elasticidade. A energia de deformação específica u , independentemente da tensão e deformação, pode ser descrita segundo a Eq. (3.24):

$$u = \int_0^{\varepsilon_{ij}} S_{ij} d\varepsilon_{ij} = \int_0^{\varepsilon_e} \sigma_e d\varepsilon_e \quad (3.24)$$

onde S_{ij} é o 2º tensor de tensão de Piola-Kirchhoff. Ao se obter a derivada da Eq. (3.22), obtêm-se $\varepsilon_e d\varepsilon_e = \varepsilon_{ij} C_{ijkl}^0 d\varepsilon_{kl}$. Substituindo o resultado na Eq. (3.24) e substituindo σ_e por $Kf(\varepsilon_e)$ da Eq. (3.23), a equação constitutiva será:

$$S_{ij} = Kf(\varepsilon_e) / \varepsilon_e C_{ijkl}^0 \varepsilon_{kl} = Kg(\varepsilon_e) C_{ijkl}^0 \varepsilon_{kl} \quad (3.25)$$

em que $g(\varepsilon_e) = f(\varepsilon_e)/\varepsilon_e$. Finalmente, o 2º tensor de tensão de Piola–Kirchhoff pode ser transformado em um valor escalar para a tensão efetiva:

$$\sigma_e^2 = S_{ij} C_{ijkl}^{-1} S_{kl} \quad (3.26)$$

Nesta Tese, embora as análises que envolvem a não linearidade física terem o foco na plasticidade, a não linearidade física não se resume apenas a isto. Eventos como dano, fratura e impacto também podem configurar análises não lineares. A metodologia adotada nesta Tese pode ser utilizada para tratar problemas de dano, fratura ou impacto apesar destes tipos de problema não serem o objetivo de estudo.

Neste trabalho, a penalização é feita na matriz que define as relações constitutivas do material, ou seja, a Matriz de Elasticidade. As diferenças pertinentes nos modelos analisados serão mostrados nos capítulos a seguir.

4

METODOLOGIA ADOTADA PARA A OTIMIZAÇÃO MULTIOBJETIVO

Este capítulo se destina a tratar especificamente da otimização multiobjetivo e dos métodos usados para a obtenção da topologia ótima a partir desta abordagem. Além disso, o capítulo realiza uma síntese dos parâmetros de otimização SESO, explicando tanto sua importância na análise como a calibração dos mesmos no software desenvolvido.

4.1 Otimização Multiobjetivo

Conforme já descrito anteriormente nesta Tese, o processo de otimização visa encontrar um modelo adequado, de acordo com um ou mais critérios de custo, a fim de minimizar (ou maximizar) um objetivo de projeto. Em outras palavras, pode afirmar-se que otimização é a busca de um modelo melhor para atender um determinado objetivo específico. Porém, dependendo do objetivo de projeto, pode ser necessário otimizar sistematicamente, e simultaneamente, um conjunto de funções objetivo, ou até mesmo realizar uma tomada de decisão face a diversas opções de escolha. A esse processo se dá o nome de Otimização Multiobjetivo (MARLER e ARORA, 2004).

Múltiplos critérios conflitantes precisam ser tratados eventualmente, fazendo com que a otimização multiobjetivo seja um caminho para resolver esse impasse, ou seja, a otimização multiobjetivo busca lidar com objetivos conflitantes, proporcionando uma estrutura matemática para obter um estado ideal que acomoda os vários critérios exigidos pelas

restrições do projeto. Logo, é um processo de otimização que lida sistematicamente e, simultaneamente, com um conjunto de funções objetivo (ODU e CHARLES-OWABA, 2013).

Segundo Rao (2009), de forma matemática, problemas de otimização multiobjetivo com restrições de desigualdade podem ser expressos da seguinte forma:

$$\begin{aligned}
 & \text{Encontrar} \quad \mathbf{X} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \\
 & \text{que minimiza} \quad [f_1(\mathbf{X}), f_2(\mathbf{X}), \dots, f_k(\mathbf{X})] \\
 & \text{sujeito a} \quad g_j(\mathbf{X}) \leq 0, \quad j=1,2,\dots,m
 \end{aligned} \tag{4.1}$$

onde k representa o número de funções objetivo a serem minimizadas e todas as funções $f_i(\mathbf{X})$ e $g_j(\mathbf{X})$ podem ser não lineares. É importante ressaltar que um problema de otimização multiobjetivo pode ser representado pela minimização de um vetor que rege o problema.

De forma geral, não há uma solução para o vetor \mathbf{X} que atenda e minimize todas as funções f_k simultaneamente. Por isso, um conceito denominado Solução Ótima de Pareto (ou simplesmente Ótimo de Pareto) é usado para a resolução de problemas de otimização multiobjetivo. Esse conceito parte do princípio que uma solução viável \mathbf{X} é chamada de Ótimo de Pareto se não existe outra solução viável \mathbf{Y} tal que $f_i(\mathbf{Y}) \leq f_i(\mathbf{X})$, para $i = 1, 2, \dots, k$ com $f_j(\mathbf{Y}) < f_j(\mathbf{X})$ para, pelo menos, um j . Em outras palavras, um vetor viável \mathbf{X} é chamado Ótimo de Pareto se não houver outra solução viável \mathbf{Y} que reduziria alguma função objetivo sem provocar um aumento simultâneo em, pelo menos, uma outra função objetivo.

Os pontos no espaço das funções objetivo que correspondem ao Ótimo de Pareto formam a chamada Fronteira de Pareto (Fig. 4.1).

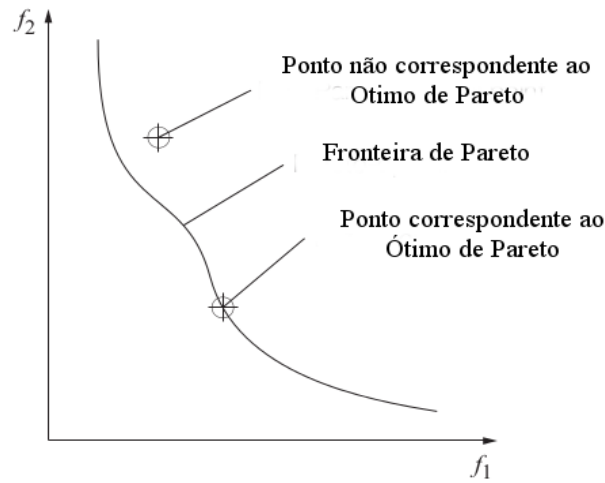


Figura 4.1 – Pontos correspondentes ao Ótimo de Pareto e Fronteira de Pareto para duas funções objetivo f_1 e f_2 . Fonte: adaptado de Sehlström (2013)

Como as funções representam diferentes objetivos e são frequentemente contraditórias entre si, é conveniente utilizar métodos adequados para resolver a Eq. (4.1) ou, em outras palavras, obter o Ótimo de Pareto. Existem várias formas de se obter essa solução. Uma delas, inclusive a usada neste trabalho, consiste em ponderar as funções objetivo do problema com uma função objetivo escalar. De forma geral, uma função útil $U_i(f_i)$ é definida para cada objetivo, de acordo com a importância de f_i comparado às outras funções objetivo. O total correspondente a essa função U pode ser definido por:

$$U = \sum_{i=1}^k U_i f_i \quad (4.2)$$

O vetor solução \mathbf{X}^* pode ser obtido, neste caso, minimizando o total U sujeito às restrições $g_j(\mathbf{X}) \leq 0$, com $j = 1, 2, \dots, m$. Uma representação simples, portanto, pode ser expressa por:

$$U = \sum_{i=1}^k U_i = \sum_{i=1}^k w_i f_i(\mathbf{X}) \quad (4.3)$$

em que $0 \leq w_i < 1$ é um fator de ponderação escalar, que pode ser um ponto da fronteira de Pareto, associado à i -ésima função objetivo. O somatório de todos esses fatores deve ser 1.

4.2 Processo Hierárquico de Análise (AHP) e Método das Somas Ponderadas (WSM)

O Método AHP (*Analytic Hierarchy Process*), introduzido por Saaty (1980), é uma ferramenta eficaz para lidar com tomadas de decisões complexas, ajudando o usuário a definir prioridades para se tomar a melhor decisão. O método também ajuda a perceber os aspectos subjetivos e objetivos de uma decisão, reduzindo problemas complexos a uma simples comparação de pares de atributos. E, além disso, o método incorpora uma técnica útil para verificação da consistência da tomada de decisão, ou seja, analisa a integridade da resposta.

Uma das principais vantagens da AHP é que a análise não exige uma amostra significativa e comparável. Os atributos, diferentes, que não podem ser diretamente comparados podem ser analisados de uma forma simples através desse método. Assim, essa abordagem elimina a complexidade do problema e pode até mesmo ser aplicada (em um caso extremo) a um problema que possua apenas um único atributo (DUTTA, 2007).

O método AHP procura um meio de decompor o problema a partir de uma hierarquia de “subproblemas” que podem ser facilmente compreendidos e avaliados subjetivamente. As avaliações subjetivas são convertidas em valores numéricos e processadas para classificar cada alternativa em uma escala numérica.

De acordo com Baby (2013), a metodologia do AHP pode ser descrita da seguinte forma:

1) *Estruturação do modelo de decisão: a construção de uma hierarquia de objetivos/critérios e alternativas.*

Esta é a parte mais importante de tomada de decisão. A estruturação do problema de forma hierárquica é fundamental para o processo do método AHP (Fig. 4.2). No topo do fluxograma se encontra a meta ou objetivo do problema a ser estudado e analisado. O nível mais baixo são as alternativas a serem comparadas. O nível intermediário possui os parâmetros a serem comparados: volume, deslocamento, tensão de von Mises e índice de performance.

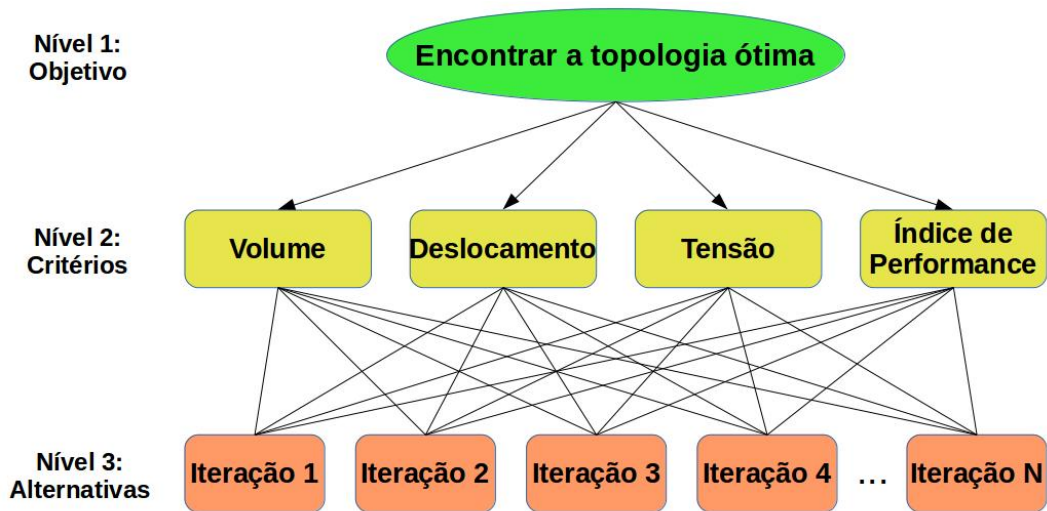


Figura 4.2 – Fluxograma de três níveis hierárquicos para encontrar a topologia ótima

2) *Introdução das alternativas: estabelecer prioridades entre os elementos da hierarquia.*

É necessário avaliar a comparação entre os atributos como igual importância, moderada importância, forte importância, muito forte importância ou extrema importância. Essa avaliação é definida em um formato próprio, de acordo com a Tabela 4.1. Para este trabalho, a valor de cada elemento foi determinado de acordo com uma melhor estabilidade para a estrutura.

Tabela 4.1 – (a) Tabela de preferência dos parâmetros (b) Tabela de pontuações relativas

	9	7	5	3	1	3	5	7	9		9	Extrema Importância
Volume				x						Deslocamento	7	Muito Forte Importância
Volume			x							Tensão	5	Forte Importância
Volume				x						Índice de Performance	3	Moderada Importância
Deslocamento					x					Tensão	3	Moderada Importância
Deslocamento						x				Índice de Performance	5	Forte Importância
Tensão							x			Índice de Performance	7	Muito Forte Importância
											9	Extrema Importância

(a)

(b)

A matriz de comparação gerada a partir da tabela de preferência dos parâmetros é mostrada na Tabela 4.2. Os elementos acima da diagonal principal, cujos valores são unitários, correspondem aos valores inteiros das atribuições correspondentes à coluna marcada. Os elementos abaixo dessa mesma diagonal principal seguem a regra $a_{ij} = 1/a_{ji}$.

Tabela 4.2 – Matriz de comparação dos parâmetros

1	3	5	3
1/3	1	3	3
1/5	1/3	1	1
1/3	1/3	1	1

É importante observar que a ordem da matriz está relacionada com a quantidade de parâmetros adotados para a análise.

3) *Comparação (relativa): sintetização dos resultados, incluindo níveis que possam haver incertezas.*

O principal autovalor e o seu correspondente autovetor normalizado da matriz de comparação, gerada a partir da preferência dos parâmetros, dá a importância relativa aos vários critérios a serem avaliados. Os valores normalizados do vetor são os pesos, que devem ser usados para resolver o problema multiobjetivo. Um programa em código Python foi desenvolvido para encontrar ambos os autovalores e autovetores, como mostrado na Fig. 4.3.

```

#Autovalores e autovetores

import numpy

n = 4
a = [None]*n
vec = [None]*n
for i in range(n):
    a[i] = [None]*n

a[0][1] = 3
a[0][2] = 5
a[0][3] = 3
a[1][2] = 3
a[1][3] = 3
a[2][3] = 1

for i in range(n):
    for j in range(n):
        if (i==j):
            a[i][j] = 1
        if (i>j):
            a[i][j] = 1./a[j][i]

print a

res = numpy.linalg.eig(a)

print res

norma = res[1][0][0] + res[1][1][0] + res[1][2][0] + res[1][3][0]

for i in range(n):
    vec[i] = res[1][i][0]/norma

print vec

#Teste de validade

CI = (res[0][0]-n)/(n-1)
if (n==3):
    CR = (CI/0.58)*100
if (n==4):
    CR = (CI/0.9)*100
if (n==5):
    CR = (CI/1.12)*100

print CR

if (CR<=10):
    print 'OK!'
else:
    print 'INVALIDO!'

```

Figura 4.3 – Programa desenvolvido em Python para a obtenção dos autovalores e autovetores

4) *Realização de análise de sensibilidade: verificação da solução.*

A consistência da matriz de ordem n que foi encontrada anteriormente é avaliada. As comparações realizadas por este método são subjetivas e o AHP tolera a inconsistência através da redundância na abordagem realizada.

Para a consistência das avaliações feitas pelo usuário, o AHP incorpora uma técnica eficaz para isso. O chamado índice de consistência (IC) é obtido pela Eq. (4.4).

$$IC = \frac{\lambda_{max} - n}{n - 1} \quad (4.4)$$

onde λ_{max} é o maior valor da matriz de comparação e n é a dimensão da matriz. No entanto, segundo Saaty (1980), a Taxa de Consistência (TC) mede a eficiência da análise e pode ser obtida pela Eq. (4.5).

$$TC = \frac{IC}{IA} \quad (4.5)$$

em que IA representa o Índice de Consistência Aleatório, obtido pela Tabela 4.3.

Tabela 4.3 – Valores do Índice Aleatório (IA) para pequenos problemas. Fonte: LIANG *et al.* (2000)

n	1	2	3	4	5	6	7	8	9	10
IA	0	0	0,58	0,9	1,12	1,24	1,32	1,41	1,45	1,49

onde n é a dimensão da matriz. Comparativamente, o valor de TC deverá ser inferior a 10% para indicar uma melhor precisão. No presente trabalho, os valores obtidos para os pesos em todos os problemas foram 0,5229 para volume; 0,2626 para deslocamentos; 0,0988 para tensão e 0,1157 para índice de performance. O somatório dos pesos deve ser igual a 1. A fim de validar a escolha das preferências entre cada parâmetro, o valor de TC deve ser inferior a 10%. Para todos os exemplos, o valor final obtido para TC foi de aproximadamente 4,2763%, o que representa uma ótima precisão.

A fim de se aplicar cada peso obtido pelo método AHP no processo de otimização, o método WSM (*Weighted Sum Method*) pode ser utilizado para resolver problemas que envolvem um conjunto discreto de alternativas que devem ser avaliadas com base em objetivos conflitantes (BEINAT e NIJKAMP, 1998). Para qualquer dado objetivo um ou mais atributos ou critérios diferentes são usados para medir o desempenho em relação ao objetivo inicial proposto. Os atributos são, geralmente, fornecidos em diferentes unidades e, por isso, não podem ser comparados diretamente uns com os outros. No entanto, o método WSM faz com que incomparáveis atributos possam ser comparados, priorizando-os de acordo com uma atribuição adequada de pesos e, finalmente, reduzindo a quantidade de informações através da agregação de valores ponderados e padronizados. Este processo fornece não só um ranking sistemático das alternativas, mas também os pontos fortes e fracos das alternativas em destaque. Por fim, o resultado é a soma de todos critérios, ponderados pelos respectivos pesos, gerando um ranking para cada alternativa. O menor valor entre todas as alternativas configura a minimização do problema. De acordo com Marler e Arora (2004), o método WSM é uma das abordagens mais comuns para a otimização multiobjetivo, que pode ser expresso matematicamente pela Eq. (4.6).

$$R_i = \sum_{j=1}^N a_{ij} w_j \quad (4.6)$$

em que R_i é o valor da i -ésima alternativa, a_{ij} é o valor atual da i -ésima alternativa em termos do j -ésimo critério, e w_j é o peso ou importância do j -ésimo critério. Se todos os pesos forem positivos, a minimização da Eq. (3.29) representa o Ótimo de Pareto (ZADEH, 1963), ou seja, minimizar a Eq. (3.29) é suficiente para obter o Ótimo de Pareto. Entretanto, a formulação não fornece uma condição necessária para o Ótimo de Pareto (ZIONTS, 1988).

4.3 Parâmetros de Otimização, Calibração e Metodologia

A calibração do programa é feita de acordo com as condições de contorno e o índice de performance da estrutura inicial. É importante ressaltar que a retirada de elementos não ocorre de forma íntegra, ou seja, a dimensão da matriz de rigidez global não se altera. Ao invés disso,

aos elementos que não contribuem efetivamente para a estrutura, é atribuído uma baixa rigidez (valor próximo ao 0) e, a cada iteração, os valores são atualizados de acordo com a distribuição de tensões. Assim, quando os valores de remoção estão elevados, pode acontecer de a estrutura apresentar certa “hipostaticidade” (rompimento da estrutura, ou seja, remoção de elementos fundamentais para a estabilidade estrutural do modelo) em algumas iterações até que o índice de performance cancele o processo ou até mesmo retifique a estrutura ao anular o efeito hipostático, devolvendo elementos para a estrutura. Para eventuais problemas em análises estáticas, podem ser utilizadas técnicas especiais de resolução de sistemas de equações mal condicionados, como o Teorema de Lax-Milgram (RAYMOND, 1997) e o Método dos Mínimos Quadrados (LAI, ROBBINS e WEI, 1978).

Alguns parâmetros devem ser previamente definidos para que se possa alcançar a topologia ótima final, sendo eles:

- VOLMAX: com o significado de volume máximo final da estrutura, esse parâmetro é o critério de parada mais usado em muitas técnicas de otimização encontradas na literatura. Seu funcionamento é bem simples: caso o volume indicado seja atingido, a otimização é interrompida. É definido em porcentagem.
- RVOL: com o significado de máxima remoção de volume por iteração, esse parâmetro controla a porcentagem máxima de elementos que poderá ser retirado por iteração, impedindo uma remoção brusca de elementos que possa causar mal condicionamento do sistema de equações.
- FRR: com o significado de fator de raio de remoção, esse parâmetro, com o FER, são os mais importantes para que ocorra a otimização. O FRR determina o valor inicial de remoção de elementos. Multiplicado pela tensão máxima de von Mises da estrutura atual (na iteração i), os elementos que possuírem menores tensões de von Mises em relação a esse produto serão removidos de acordo com as funções reguladoras que garantem a possibilidade de remoção.

→ FER: com o significado de fator evolucionário de rejeição, esse parâmetro atualiza o FRR a cada iteração, uma vez que a tensão máxima da estrutura tende a aumentar com a retirada de elementos. Logo, o FRR também aumenta a cada iteração através de sua atualização pelo FER, até que o valor de FRR da i -ésima iteração seja tal que, multiplicado pela tensão máxima da respectiva iteração, não provoque a remoção de mais elementos, definindo portanto, nessa iteração, a topologia ótima final.

Em resumo, a ideia é usar o Processo Analítico Hierárquico (AHP) para encontrar os pesos necessários e aplicá-los ao Método das Somas Ponderadas (WSM). Com os pesos associados, respectivamente, aos atributos volume, deslocamento, tensão e índice de performance, pretende-se minimizar o somatório obtido pelo WSM. O valor mais baixo encontrado na soma ao longo do processo iterativo corresponde à topologia ótima procurada. Estes valores serão apresentados em tabelas e gráficos para os exemplos nos capítulos seguintes da Tese. Todos os valores serão dados em porcentagem, garantindo uma forma padrão de medição. Para isso, o valor mais elevado de cada atributo em toda a análise será atribuído 100% e comparado com os demais valores de forma proporcional. Os valores mais elevados serão utilizados para comparar todos os valores de iterações anteriores e/ou posteriores. Uma atenção especial para o índice de performance é importante, uma vez que quanto maior o valor melhor será o desempenho da estrutura. Assim, foi necessário inverter o valor, a fim de minimizar todos os atributos para encontrar o valor mais baixo do vetor WSM, isto é, esse valor mais baixo do vetor WSM representa o melhor resultado do problema ou a solução otimizada. Além da porcentagem, a tensão é fornecida em MPa e os deslocamentos em centímetros (cm), a fim de se ter uma ordem de grandeza desses atributos.

Finalmente, como um critério de parada global para a otimização, foi considerado que o volume final da estrutura não pode ser inferior a 20% do volume inicial, uma vez que para valores menores que 20% a estrutura pode passar a ser hipostática (alguns exemplos a seguir evidenciam esse efeito), além de deixar membros importantes para a estabilidade com uma espessura muito reduzida. Um dos exemplos foi definido com volume final a 10% para mostrar a esbeltez excessiva dos membros formados. Os parâmetros de otimização para todos os exemplos foram adotados de forma empírica, usando, para FRR, valores abaixo de 2% e, para FER, valores abaixo de 1%. Segundo Xie e Steven (1997), valores pequenos de FRR e

FER são necessários, uma vez que quanto mais alto forem esses parâmetros, maior será a remoção de elementos e maiores serão as chances de ocorrer uma instabilidade estrutural por uma brusca remoção de elementos. Todos os valores usados para estes parâmetros foram citados nos exemplos.

Foram testados diferentes parâmetros para todos os exemplos numéricos que conduziram a soluções ótimas semelhantes para níveis de remoção de material equivalentes. Ou seja, para o mesmo nível de volume a solução otimizada é praticamente a mesma para diferentes parâmetros de otimização. A escolha dos parâmetros do processo de otimização via SESO influencia no volume de remoção por iteração possível de ser alcançado. Finalmente, fatores de remoção elevados podem inviabilizar o processo evolutivo e, por outro lado, fatores de remoção muito baixos demandam passos excessivos de otimização, podendo causar instabilidades numéricas.

5

RESULTADOS LINEARES

Este capítulo se destina a tratar especificamente de nove resultados lineares obtidos pela implementação do código de otimização SESO. As estruturas foram modeladas no *software* Salome-Meca[®] versão 2014.1, descrito no capítulo anterior. Todos os modelos aqui apresentados foram analisados em 3D, sendo que os modelos bidimensionais retirados da literatura foram aproximados com uma espessura consideravelmente menor em relação aos outros parâmetros (ou seja, os modelos foram considerados sendo placas). É importante ressaltar que os valores de deslocamento considerados para todas as análises deste capítulo referem-se sempre ao maior deslocamento ao qual a estrutura fora submetida na iteração correspondente.

5.1 Problema de Duas Barras

Este é um problema clássico encontrado na literatura que consiste em uma chapa retangular biapoiada, cujos apoios encontram-se alinhados verticalmente, e a carga é aplicada no centro da extremidade oposta às restrições, de acordo com a Fig. 5.1.

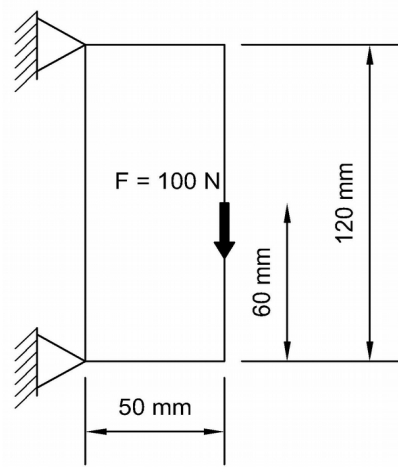


Figura 5.1 – Domínio inicial do problema de duas barras

Para este problema, os dados pertinentes do modelo são os seguintes:

- Malha utilizada (*mesh*): a malha gerada no Code-Aster[®] possui 10208 elementos do tipo TETRA4 (Fig. 5.2);
- Módulo de Elasticidade (E): 210 GPa;
- Coeficiente de Poisson (ν): 0,3;
- Espessura (t): 10 mm;
- Volume máximo desejado (VOLMAX) = 20%;
- Remoção máxima de volume mapeado por iteração (RVOL) = 95% (5% dos elementos são devolvidos para a estrutura);
- Fator de raio de rejeição (FRR) = 1,85%;
- Fator evolucionário de rejeição (FER) = 0,95%.

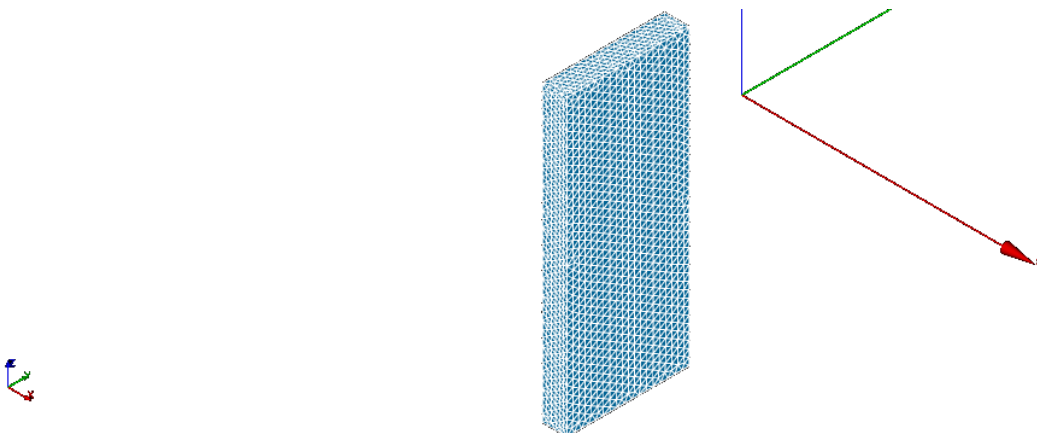


Figura 5.2 – Malha de elementos finitos gerada pelo Salome-Meca[®]

O processo evolutivo de otimização do problema de duas barras é mostrado nas Figs. 5.3 a 5.5.

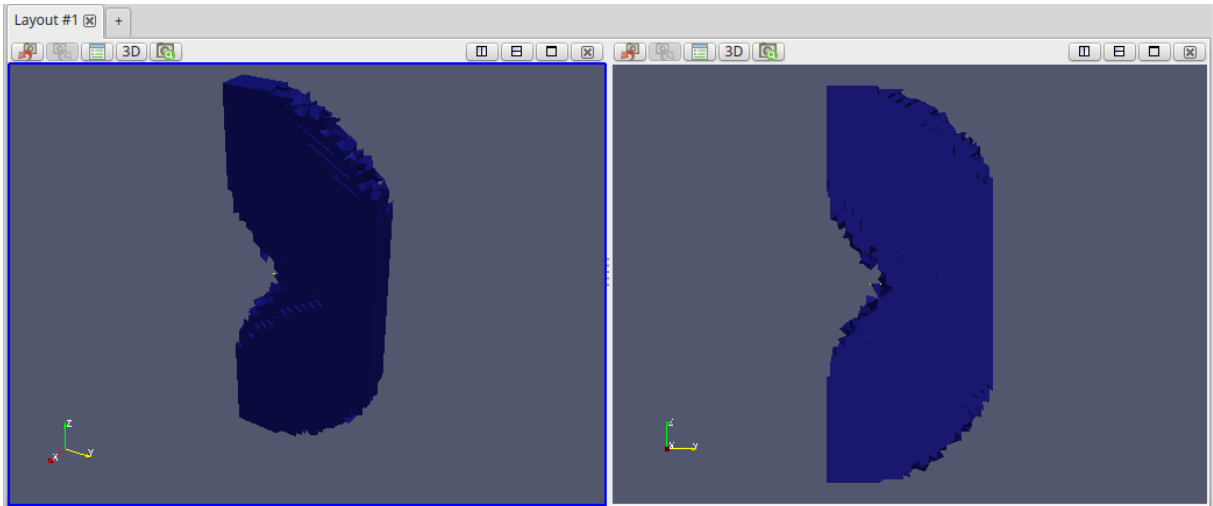


Figura 5.3 – Topologia parcial resultante do passo de otimização 3

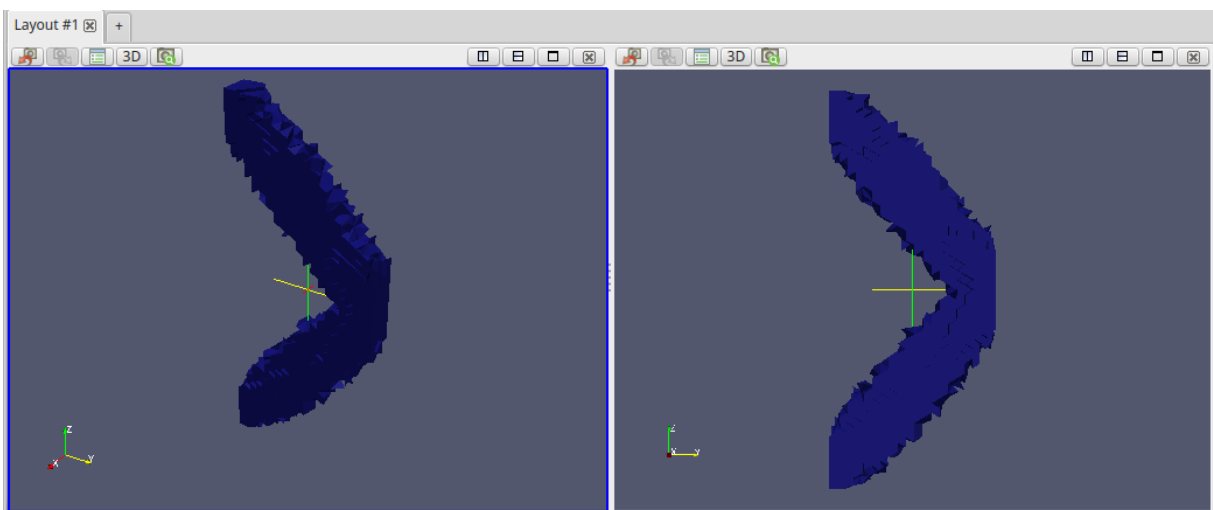


Figura 5.4 – Topologia parcial resultante do passo de otimização 14

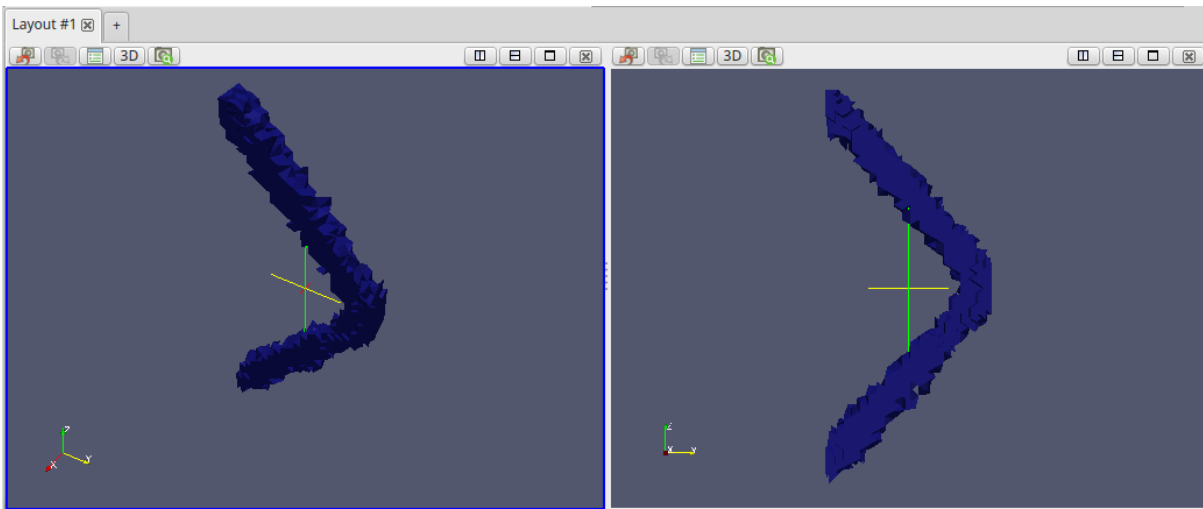


Figura 5.5 – Topologia parcial resultante do passo de otimização 40 (última iteração)

A topologia final ótima obtida para o problema é mostrada na Fig. 5.6.

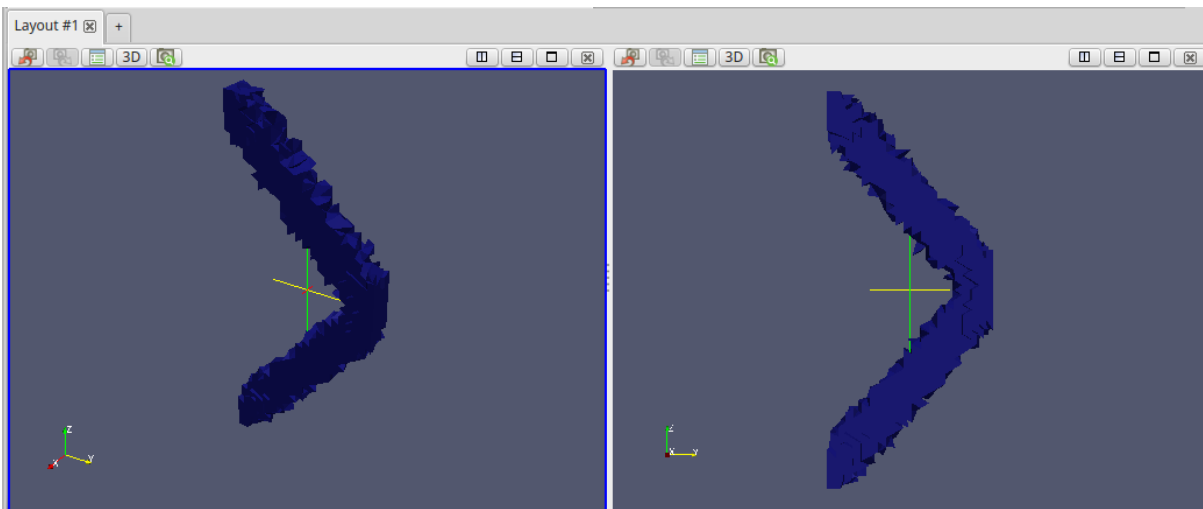
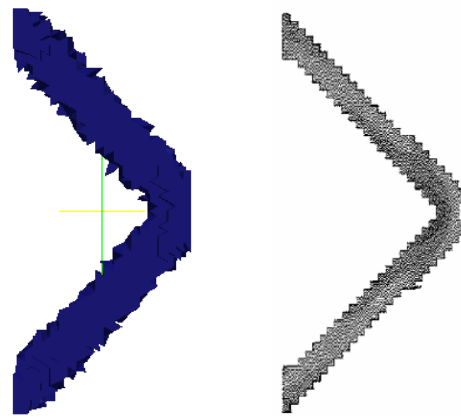


Figura 5.6 – Topologia final ótima resultante do passo de otimização 24

A Fig. 5.7 mostra um comparativo entre a topologia final ótima obtida do modelo proposto pela encontrada por Xie e Steven (1993). Para este problema, Xie e Steven não se preocuparam com o volume final da topologia ótima em si, mas com o aspecto da resposta em proporção com o FRR, ou seja, as porcentagens de FRR em função das topologias parciais obtidas ao longo do processo iterativo. Como várias respostas parciais foram obtidas, os autores definiram a topologia ótima como sendo a estrutura cujos membros fossem mais finos que corresponde a estrutura com menor volume entre todas as outras respostas.



(a) Formulação SESO

(b) Xie e Steven (1993)

Figura 5.7 – Topologias finais ótimas para o problema de duas barras

O campo de tensões para o problema de duas barras é mostrado na Fig. 5.8.

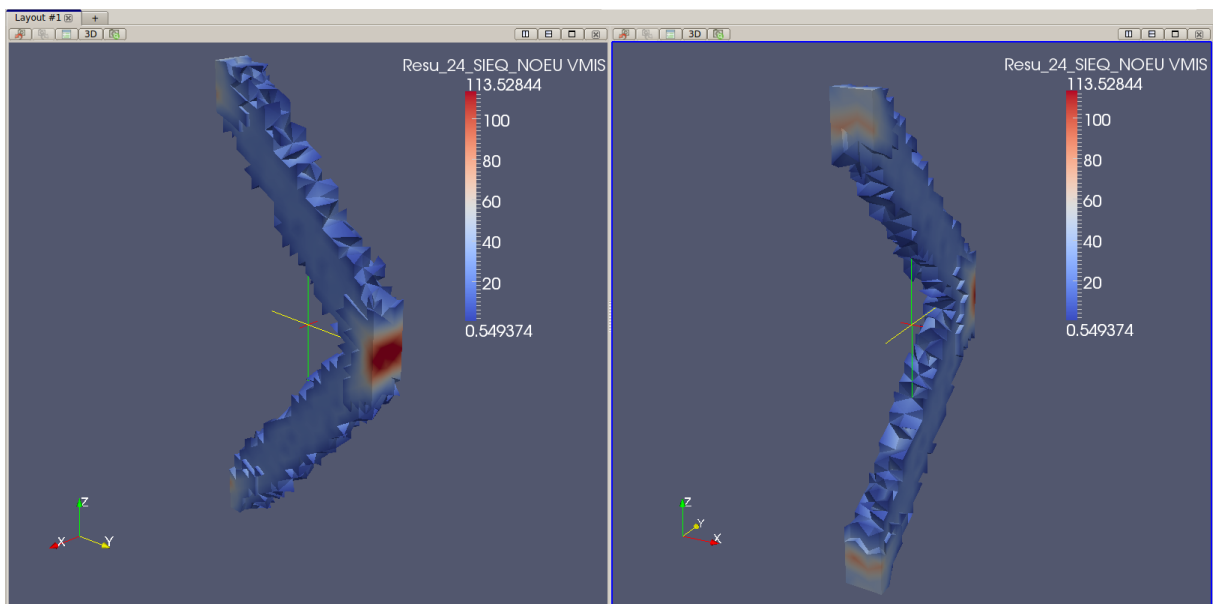


Figura 5.8 – Campo de tensões da topologia ótima final para o problema de duas barras

Os elementos que sofrem as maiores solicitações são os que se concentram próximo ao ponto de aplicação da carga, assim como nos apoios. Percebe-se que há uma tendência de a tensão ser maior nos elementos que compreendem o trecho mais curto entre a carga aplicada e as condições de contorno impostas inicialmente gerando, para este exemplo, duas barras.

Ainda segundo Fernandes (2013), diferentes malhas podem gerar alterações na resposta devido à dependência da malha da técnica heurística SESO, embora minimizada pela suavização proposta. Considerando diferentes malhas para este exemplo, os resultados são mostrados na Fig. 5.9.

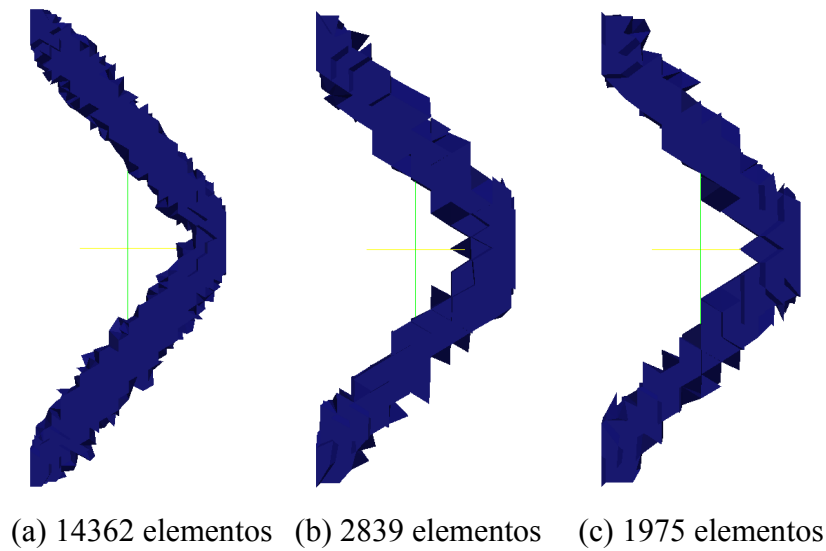


Figura 5.9 – Diferentes malhas para o problema de duas barras

Segundo Fernandes (2013), com o refinamento da malha tende-se a obter respostas mais adequadas. Para este exemplo, o aspecto da topologia ótima se manteve para as diferentes malhas adotadas.

Além disso, a Tabela 5.1 mostra os valores dos parâmetros e do vetor WSM. A linha em destaque nessa tabela marca a iteração e o menor valor do vetor WSM, caracterizando a resposta da melhor topologia obtida pela análise de otimização multiobjetivo. Em seguida, tem-se na Fig. 5.10 um gráfico que descreve o comportamento de cada um destes parâmetros ao longo do processo iterativo.

Tabela 5.1 – Valores dos parâmetros e do Vetor WSM

Iteração	Volume	Deslocamento (cm, %)		Tensão (MPa, %)		IP	WSM
0	100,00	0,0108	39,09	113,53	100,00	100,00	84,01
1	94,41	0,0109	39,13	113,51	99,99	99,81	81,07
2	88,66	0,0109	39,24	113,48	99,96	99,73	78,08
3	82,74	0,0109	39,47	113,44	99,92	99,53	75,02
4	76,94	0,0111	39,87	113,37	99,86	89,81	70,96
5	71,28	0,0112	40,45	113,27	99,77	85,13	67,60
6	65,00	0,0115	41,31	113,24	99,75	84,48	64,47
7	59,99	0,0118	42,64	113,13	99,65	82,69	61,98
8	55,74	0,0122	44,04	112,99	99,53	78,41	59,62
9	52,16	0,0126	45,56	112,91	99,45	77,81	58,07
10	49,23	0,0131	47,11	112,91	99,45	77,31	56,88
11	46,82	0,0135	48,65	112,82	99,38	72,65	55,48
12	44,65	0,0139	50,23	112,78	99,34	72,31	54,72
13	42,79	0,0143	51,60	112,55	99,14	70,84	53,92
14	40,86	0,0147	53,03	112,51	99,10	66,27	52,75
15	39,20	0,0151	54,59	112,57	99,16	65,60	52,22
16	37,82	0,0156	56,42	112,58	99,17	63,84	51,78
17	36,19	0,0161	57,89	112,45	99,05	63,66	51,28
18	34,97	0,0166	59,77	112,07	98,72	63,48	51,08
19	33,83	0,0170	61,29	111,90	98,57	62,94	50,81
20	32,74	0,0175	62,99	111,34	98,08	61,84	50,51
21	31,64	0,0179	64,57	111,07	97,84	61,84	50,33
22	30,48	0,0184	66,32	110,93	97,71	57,95	49,71
23	29,52	0,0189	68,27	110,89	97,68	57,40	49,65
24	28,63	0,0194	70,10	110,74	97,55	56,64	49,57
25	27,90	0,0200	71,98	110,92	97,70	56,34	49,66
26	27,18	0,0205	73,94	111,04	97,81	55,66	49,73
27	26,62	0,0209	75,53	111,23	97,98	54,38	49,72
28	25,90	0,0214	76,99	111,82	98,49	54,55	49,80
29	25,36	0,0218	78,57	111,99	98,64	54,50	49,94
30	24,81	0,0222	79,89	112,18	98,81	51,39	49,66
31	24,20	0,0226	81,64	112,14	98,78	50,75	49,72
32	23,82	0,0232	83,56	112,19	98,82	48,30	49,75
33	23,24	0,0235	84,69	112,47	99,07	48,29	49,77
34	22,81	0,0240	86,44	111,66	98,36	47,50	49,84
35	22,22	0,0246	88,61	111,79	98,47	49,75	50,37
36	21,80	0,0251	90,67	111,42	98,14	49,42	50,62
37	21,20	0,0257	92,70	111,41	98,14	49,79	50,88
38	20,65	0,0264	95,35	110,71	97,52	49,41	51,19
39	20,26	0,0271	97,82	110,50	97,33	49,32	51,60
40	20,00	0,0277	100,00	110,11	96,99	47,17	51,76

Neste exemplo, o volume começa a decrescer de forma quase linear até próximo da iteração 10, onde o material é removido (penalizado) mais suavemente a partir de então. Quanto aos deslocamentos, quase não há variações nas primeiras iterações, mas passa a ser praticamente linear a partir da iteração 7, mantendo essa postura até o fim do processo de otimização. Mas

o que, de fato, tem um certo destaque neste exemplo é o comportamento da tensão, que pouco se altera ao longo de toda a análise. Em outras palavras, além dos elementos penalizados não fazerem falta para os novos modelos, parciais, que surgem a cada iteração, eles fazem com que os remanescentes suportem toda a carga inicial da mesma maneira até o fim, o que comprova que a topologia obtida é uma estrutura otimizada de acordo com as condições iniciais impostas ao modelo. O índice de performance decresce ao longo do processo iterativo, mantendo poucas variações.

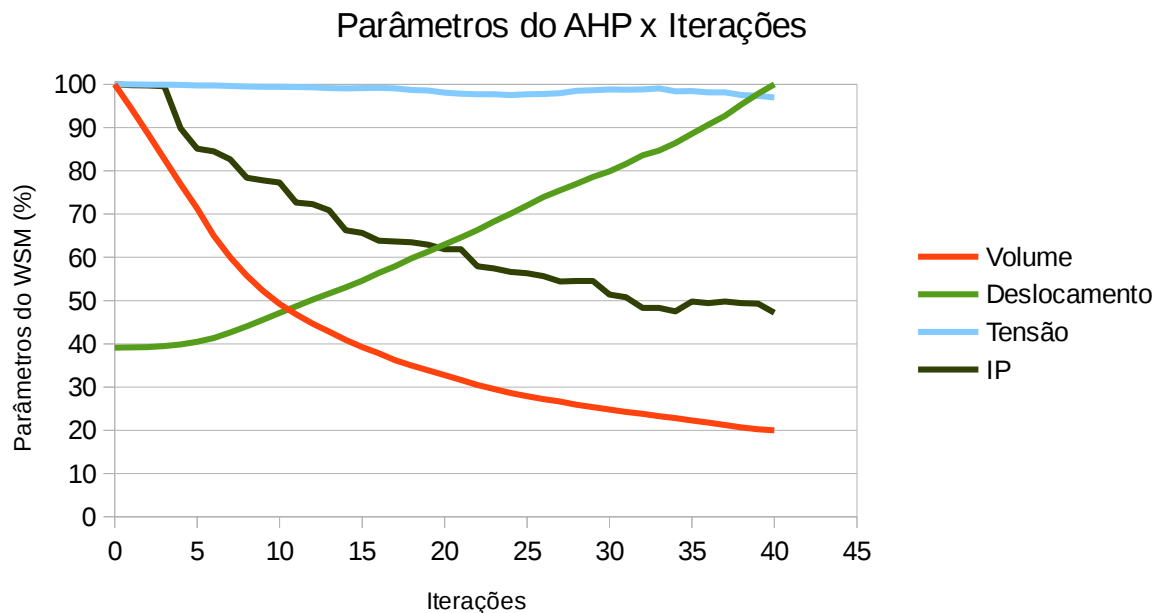


Figura 5.10 – Parâmetros do Vetor WSM (Volume, Deslocamentos, Tensão, Índice de Performance) x Iterações

Para demonstrar a evolução da otimização, a Fig. 5.11 resume o problema, mostrando os valores do vetor WSM ao longo do processo iterativo. O ponto vermelho mostra a iteração com o valor mínimo, que representa a melhor solução para o problema.

Neste exemplo, o vetor WSM possui valor mínimo na iteração 24, configurando esta iteração como a solução para o conjunto de possíveis soluções para a análise. Nessa iteração, o volume é de aproximados 28,5% em relação ao volume inicial. Percebe-se que o volume cai lentamente até atingir os 20% de volume máximo, ou máxima redução de volume, uma vez que a análise termina na iteração 40 e o ótimo encontra-se praticamente na metade da análise.

Ainda, a tensão nessa iteração é um dos valores mais baixos de todo o processo de otimização, configurando uma ótima resposta em termos de tensão máxima na estrutura. O deslocamento está na faixa de 70% em relação ao maior deslocamento de toda a análise e o índice de performance mantém os aproximados 56,5%.

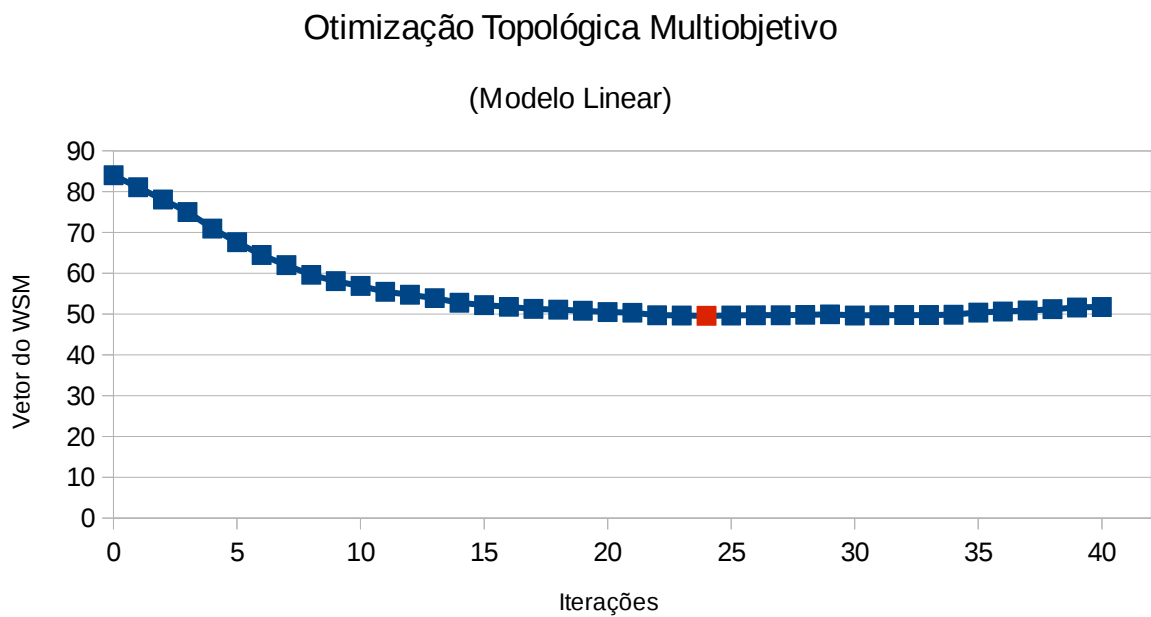


Figura 5.11 – Vetor do WSM x Iterações

5.2 Estrutura de Michell

Este modelo, estudado por Michell em 1904, consiste em uma chapa retangular isostática biapoiada com uma carga centrada na parte inferior, como mostrado na Fig. 5.12.

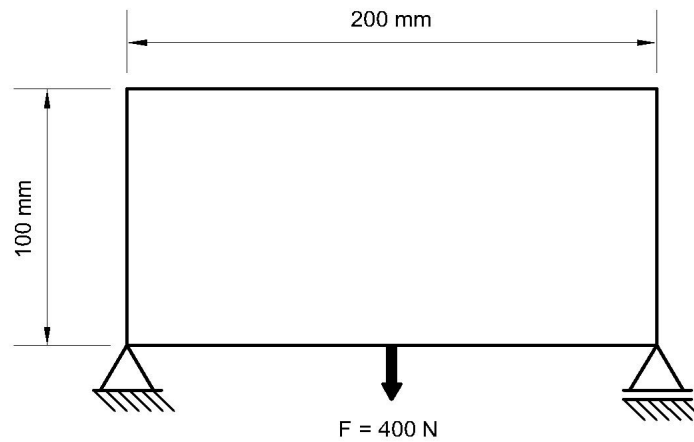


Figura 5.12 – Domínio inicial da Estrutura de Michell

Para este problema, os dados pertinentes do modelo são os seguintes:

- Malha utilizada (*mesh*): a malha gerada no Code-Aster® possui 11296 elementos do tipo TETRA4 (Fig. 5.13);
- Módulo de Elasticidade (E): 210 GPa;
- Coeficiente de Poisson (ν): 0,3;
- Espessura (t): 10 mm;
- Volume máximo desejado (VOLMAX) = 20%;
- Remoção máxima de volume mapeado por iteração (RVOL) = 95% (5% dos elementos são devolvidos para a estrutura);
- Fator de raio de rejeição (FRR) = 1,85%;
- Fator evolucionário de rejeição (FER) = 0,75%.

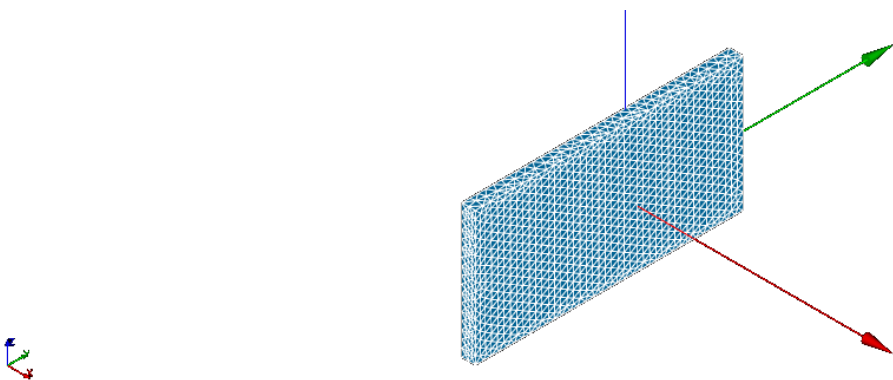


Figura 5.13 – Malha de elementos finitos gerada pelo Salome-Meca®

O processo evolutivo de otimização da Estrutura de Michell é mostrado nas Figs. 5.14 a 5.16.

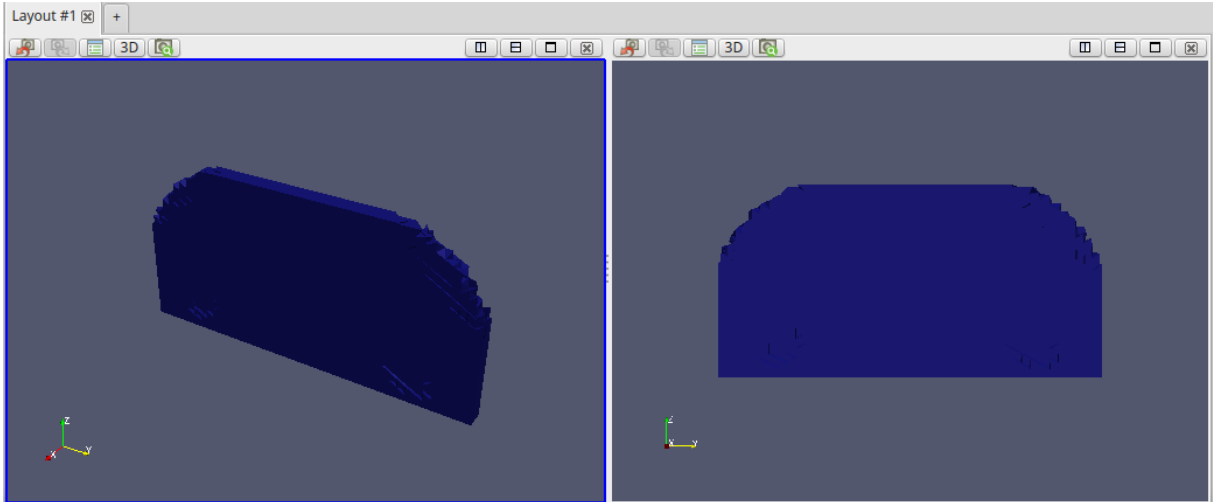


Figura 5.14 – Topologia parcial resultante do passo de otimização 7

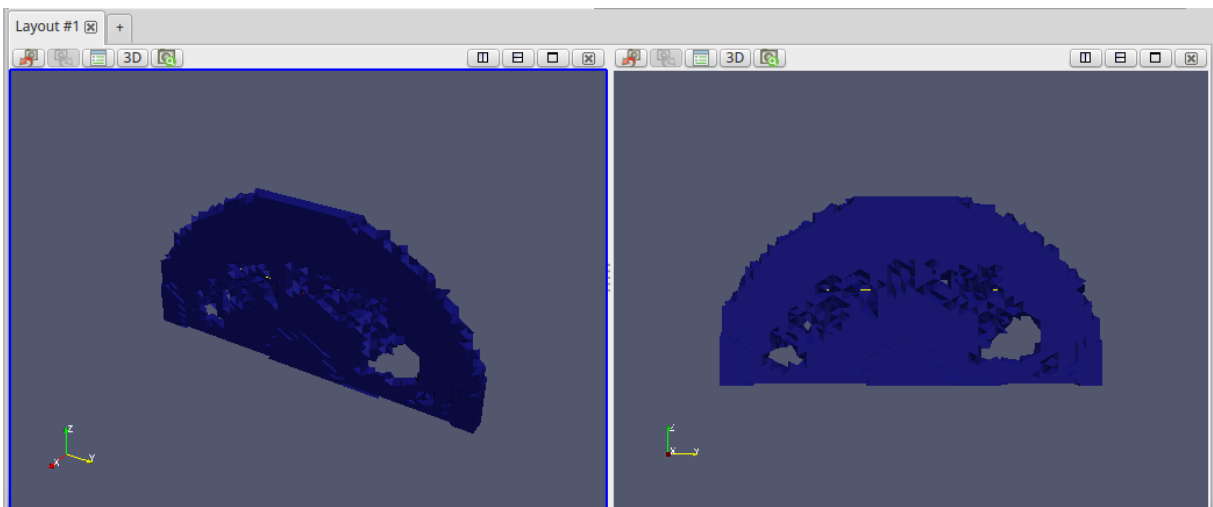


Figura 5.15 – Topologia parcial resultante do passo de otimização 20

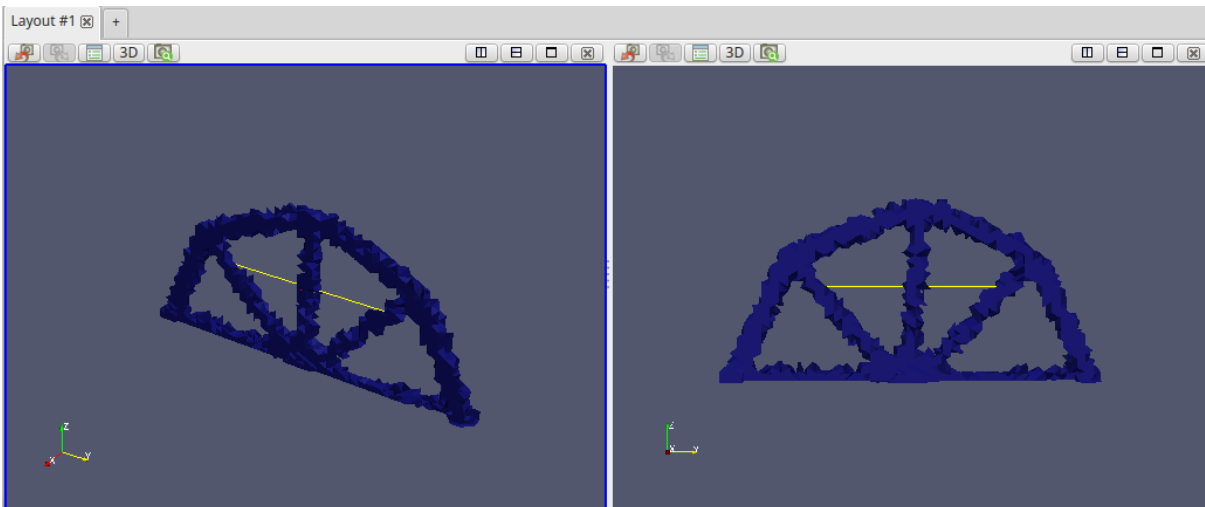


Figura 5.16 – Topologia parcial resultante do passo de otimização 51 (última iteração)

A topologia final ótima obtida para o problema é mostrada na Fig. 5.17.

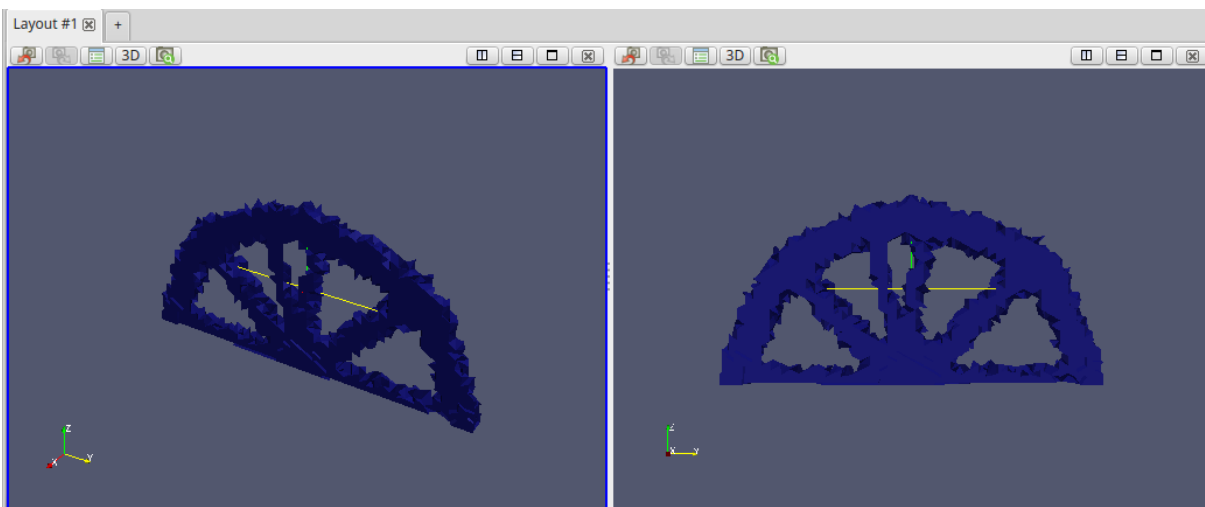
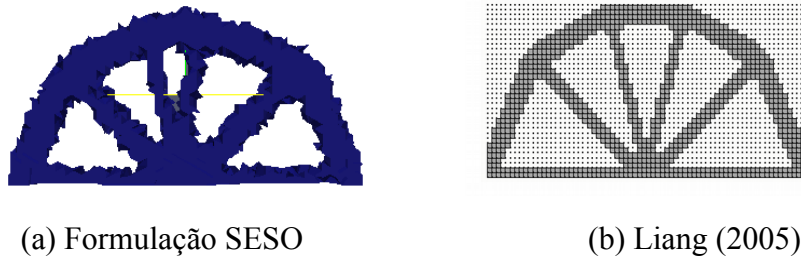


Figura 5.17 – Topologia final ótima resultante do passo de otimização 32

A Fig. 5.18 mostra um comparativo entre a topologia final ótima obtida do modelo proposto pela encontrada por Liang (2005).



(a) Formulação SESO (b) Liang (2005)
 Figura 5.18 – Topologias finais ótimas para Estrutura de Michell

Além disso, a Tabela 5.2 mostra os valores dos parâmetros e do vetor WSM. A linha em destaque nessa tabela marca a iteração e o menor valor do vetor WSM, caracterizando a resposta da melhor topologia obtida pela análise de otimização multiobjetivo. Em seguida, tem-se na Fig. 5.19 um gráfico que descreve o comportamento de cada um destes parâmetros ao longo do processo iterativo.

Aqui, o volume decresce mais lentamente que o exemplo anterior, o que justifica as 51 iterações necessárias para o término da análise. O deslocamento máximo não se altera significativamente nas primeiras iterações, já que pouco material é removido (penalizado). Percebe-se que, na iteração 7, por exemplo, a estrutura ainda é muito similar ao modelo inicial. Os únicos elementos penalizados são os superiores de ambos os lados que, de fato, não contribuem em praticamente nada para as condições impostas para a estrutura. Consequentemente, de forma semelhante aos deslocamentos, a tensão pouco varia no início da análise, reforçando a ideia da pouca alteração do modelo até então.

Tabela 5.2 – Valores dos parâmetros e do Vetor WSM

Iteração	Volume	Deslocamento (cm, %)		Tensão (MPa, %)		IP	WSM
0	100,00	0,0881	23,55	250,97	60,16	100,00	75,99
1	99,16	0,0882	23,55	250,97	60,16	96,96	75,20
2	98,39	0,0882	23,56	250,98	60,16	82,75	73,15
3	97,52	0,0882	23,57	250,99	60,16	80,70	72,46
4	96,64	0,0883	23,58	251,02	60,17	79,80	71,90
5	95,27	0,0884	23,61	251,06	60,18	79,71	71,19
6	93,78	0,0886	23,67	251,11	60,19	78,95	70,34
7	92,32	0,0889	23,76	251,16	60,20	78,69	69,57
8	90,22	0,0894	23,89	251,21	60,21	78,66	68,50
9	86,56	0,0902	24,10	251,32	60,24	78,35	66,61
10	82,32	0,0920	24,59	252,11	60,43	78,50	64,55
11	78,87	0,0946	25,27	253,39	60,74	78,50	62,96
12	76,56	0,0970	25,91	254,36	60,97	75,88	61,64
13	74,45	0,0989	26,41	255,20	61,17	75,80	60,68
14	72,53	0,1008	26,94	256,04	61,37	74,82	59,72
15	70,44	0,1029	27,48	256,78	61,55	73,36	58,62
16	68,04	0,1055	28,20	257,43	61,71	71,38	57,34
17	65,04	0,1088	29,07	257,86	61,81	71,00	55,96
18	62,33	0,1135	30,31	258,33	61,92	70,80	54,86
19	59,57	0,1173	31,33	258,85	62,05	71,28	53,76
20	56,36	0,1219	32,57	260,94	62,55	71,32	52,45
21	53,63	0,1290	34,47	263,92	63,26	69,80	51,42
22	51,12	0,1357	36,26	265,01	63,52	70,00	50,62
23	49,06	0,1429	38,19	266,71	63,93	69,74	50,07
24	46,88	0,1496	39,96	269,47	64,59	70,06	49,49
25	45,49	0,1571	41,97	271,79	65,15	69,33	49,27
26	44,31	0,1613	43,09	275,62	66,07	66,45	48,70
27	42,60	0,1654	44,18	278,39	66,73	66,98	48,22
28	40,96	0,1728	46,16	282,75	67,77	64,23	47,67
29	39,28	0,1794	47,93	284,81	68,27	63,63	47,23
30	37,87	0,1872	50,00	287,39	68,89	63,48	47,09
31	36,37	0,1939	51,80	290,91	69,73	62,37	46,73
32	34,90	0,2014	53,81	295,20	70,76	59,59	46,27
33	33,85	0,2094	55,96	301,36	72,24	58,89	46,34
34	32,84	0,2170	57,96	304,65	73,02	59,40	46,48
35	31,64	0,2246	60,00	308,72	74,00	59,18	46,46
36	30,78	0,2322	62,03	310,87	74,51	59,32	46,61
37	29,93	0,2402	64,17	315,02	75,51	59,40	46,83
38	28,81	0,2521	67,35	320,57	76,84	59,38	47,21
39	27,83	0,2610	69,72	327,43	78,48	59,56	47,51
40	27,05	0,2683	71,67	330,97	79,33	59,00	47,63
41	26,28	0,2762	73,78	338,12	81,05	58,07	47,84
42	25,42	0,2841	75,90	344,49	82,57	57,20	48,00
43	24,25	0,2928	78,22	354,50	84,97	56,65	48,17
44	23,74	0,3063	81,83	361,98	86,76	56,32	48,99
45	22,91	0,3134	83,72	367,88	88,18	55,51	49,10
46	22,38	0,3264	87,20	374,90	89,86	53,80	49,71
47	21,61	0,3331	88,99	382,28	91,63	53,07	49,86
48	21,07	0,3457	92,36	387,79	92,95	53,10	50,60
49	20,57	0,3556	95,01	406,49	97,43	48,73	50,97
50	20,26	0,3672	98,10	417,19	100,00	46,88	51,66
51	19,68	0,3743	100,00	411,50	98,64	46,78	51,71

Na iteração 20 algumas mudanças já podem ser vistas, tanto na topologia quanto no comportamento da nova estrutura parcial. Vazios internos começam a ser formados, o que gera aumento de tensão em relação ao modelo inicial. Membros internos¹ também começam a aparecer, dando forma à topologia ótima obtida pela técnica SESO. O índice de performance melhora significativamente nessa iteração.

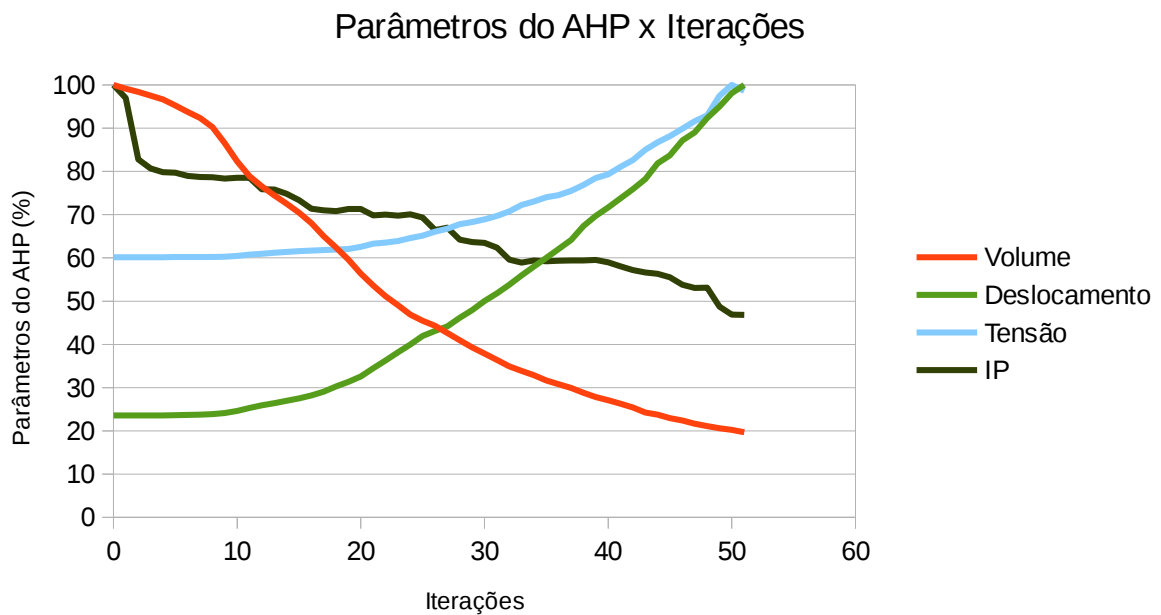


Figura 5.19 – Parâmetros do Vetor WSM (Volume, Deslocamentos, Tensão, Índice de Performance) x Iterações

Para demonstrar a evolução da otimização, a Fig. 5.20 resume o problema, mostrando os valores do vetor WSM ao longo do processo iterativo. O ponto vermelho mostra a iteração com o valor mínimo, que representa a melhor solução para o problema.

Neste exemplo, o vetor WSM possui valor mínimo na iteração 32, configurando esta iteração como a solução para o conjunto de possíveis soluções para a análise. Nessa iteração, o volume é de aproximados 35% em relação ao volume inicial. Percebe-se que, assim como no exemplo anterior, o volume cai lentamente até atingir os 20% de volume máximo, ou máxima redução

¹ O termo “membros internos” é uma analogia às barras de uma treliça, que podem trabalhar à tração ou compressão, de acordo com sua disposição e a posição e direção do carregamento imposto. Nos exemplos listados nesta Tese, alguns deles garantem a formação de membros internos tanto ao longo do processo iterativo quanto na topologia final ótima.

de volume, uma vez que a análise termina na iteração 51 e o ótimo também se encontra praticamente na metade da análise. A tensão nessa iteração é da ordem de 1,4 vezes menor que a tensão máxima oriunda de toda a análise (última iteração), configurando uma adequada resposta em termos de tensão máxima na estrutura. O deslocamento também se encontra na faixa de 70% em relação ao maior deslocamento de toda a análise e o índice de performance mantém os aproximados 59,5%.

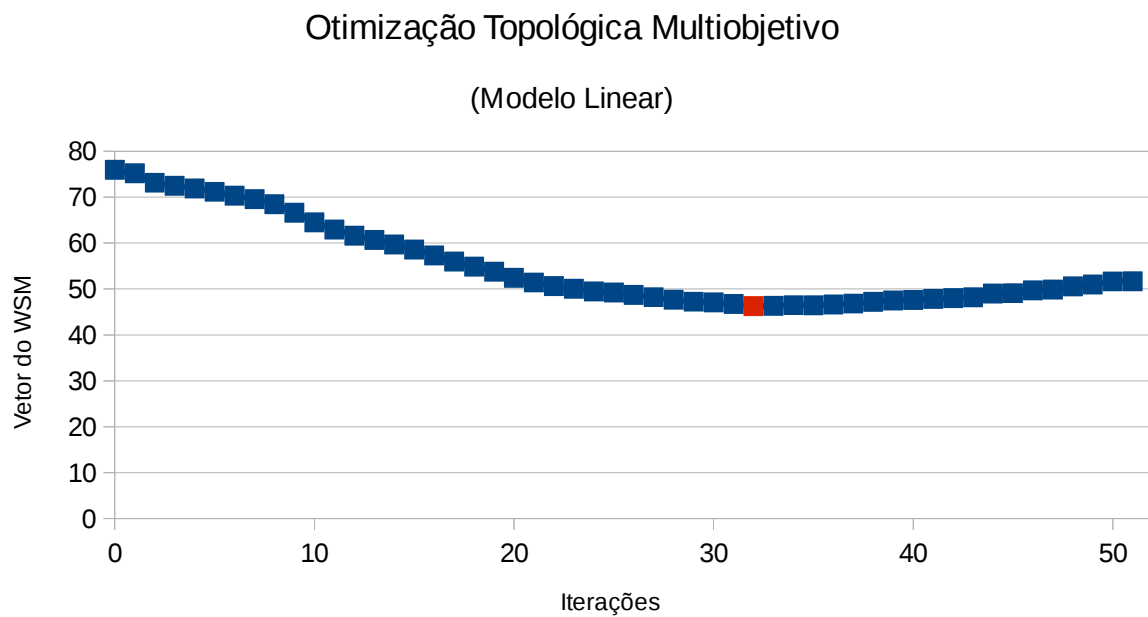


Figura 5.20 – Vetor do WSM x Iterações

5.3 Placa Fina Apoiada nas Quatro Extremidades com Carga Centrada

Este modelo consiste em uma placa retangular apoiada nos quatro vértices com uma carga centrada na parte superior, como mostrado na Fig. 5.21.

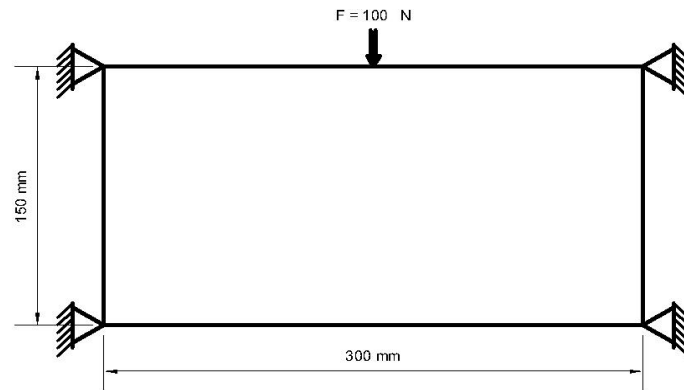


Figura 5.21 – Domínio inicial da placa fina apoiada nas quatro extremidades com carga centrada

Para este problema, os dados pertinentes do modelo são os seguintes:

- Malha utilizada (*mesh*): a malha gerada no Code-Aster® possui 11059 elementos do tipo TETRA4 (Fig. 5.22);
- Módulo de Elasticidade (E): 210 GPa;
- Coeficiente de Poisson (ν): 0,3;
- Espessura (t): 20 mm;
- Volume máximo desejado (VOLMAX) = 20%;
- Remoção máxima de volume mapeado por iteração (RVOL) = 95% (5% dos elementos são devolvidos para a estrutura);
- Fator de raio de rejeição (FRR) = 1,75%;
- Fator evolucionário de rejeição (FER) = 0,75%.

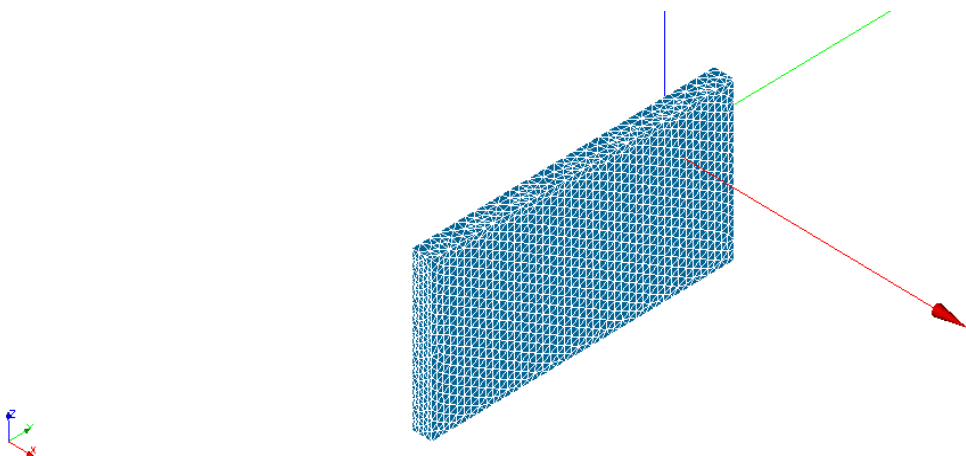


Figura 5.22 – Malha de elementos finitos gerada pelo Salome-Meca®

O processo evolutivo de otimização da placa fina apoiada nos quatro vértices com carga centrada é mostrado nas Figs. 5.23 a 5.25.

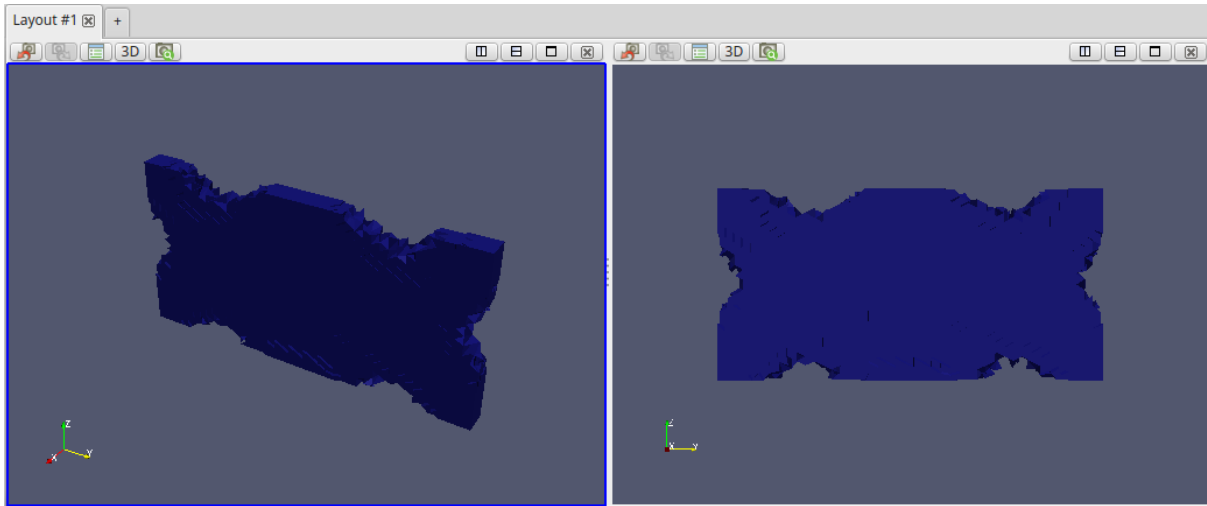


Figura 5.23 – Topologia parcial resultante do passo de otimização 6

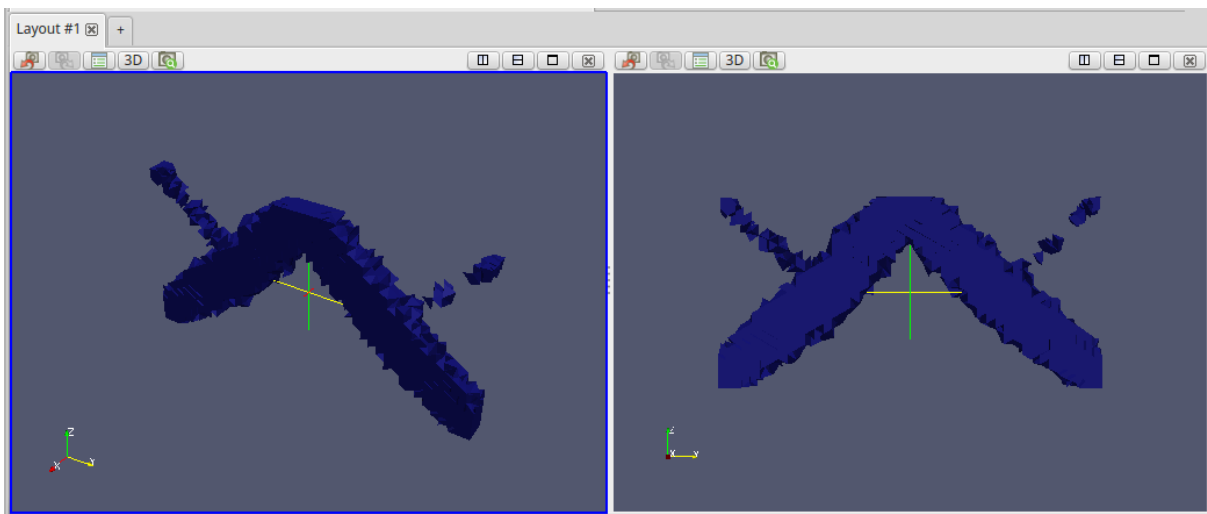


Figura 5.24 – Topologia parcial resultante do passo de otimização 20

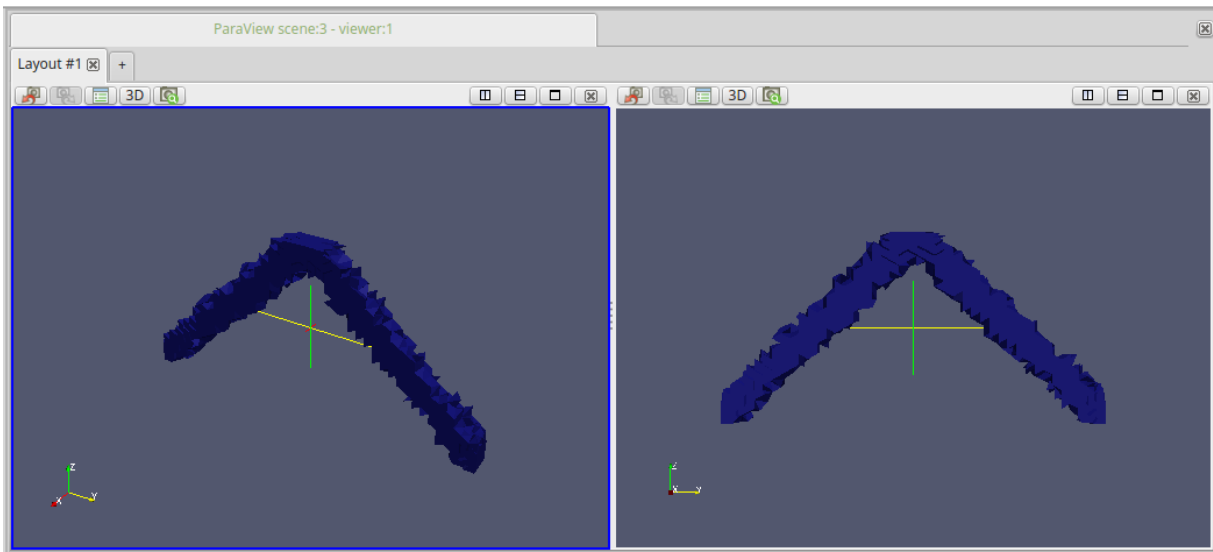


Figura 5.25 – Topologia parcial resultante do passo de otimização 42 (última iteração)

A topologia final ótima obtida para o problema é mostrada na Fig. 5.26.

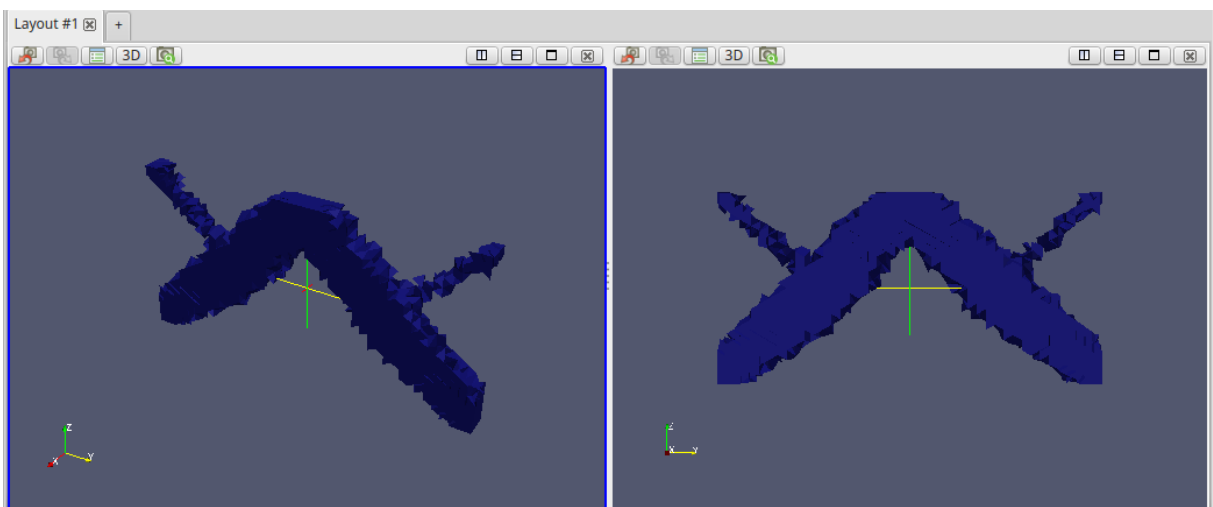
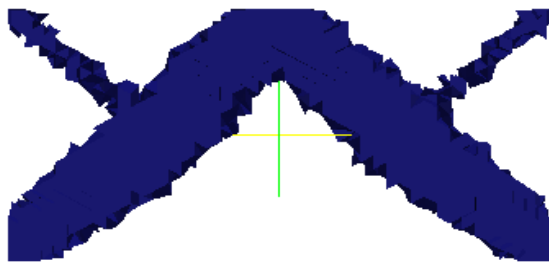
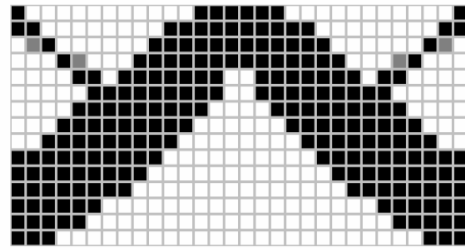


Figura 5.26 – Topologia final ótima resultante do passo de otimização 19

A Fig. 5.27 mostra um comparativo entre a topologia final ótima obtida do modelo proposto pela encontrada por Wang (2003).



(a) Formulação SESO



(b) Wang (2003)

Figura 5.27 – Topologias finais ótimas para placa fina presa nas quatro extremidades com carga centrada

Além disso, a Tabela 5.3 mostra os valores dos parâmetros e do vetor WSM. A linha em destaque nessa tabela marca a iteração e o menor valor do vetor WSM, caracterizando a resposta da melhor topologia obtida pela análise de otimização multiobjetivo. Em seguida, tem-se na Fig. 5.28 um gráfico que descreve o comportamento de cada um destes parâmetros ao longo do processo iterativo.

Tabela 5.3 – Valores dos parâmetros e do Vetor WSM

Iteração	Volume	Deslocamento (cm, %)		Tensão (MPa, %)		IP	WSM
0	100,00	2,6349	34,45	71,20	77,06	100,00	80,52
1	98,63	2,6356	34,46	71,20	77,06	97,25	79,49
2	96,92	2,6378	34,48	71,19	77,05	94,81	78,32
3	94,47	2,6422	34,54	71,16	77,02	93,97	76,95
4	91,12	2,6520	34,67	71,09	76,95	86,40	74,35
5	87,94	2,6730	34,94	71,07	76,92	83,07	72,37
6	83,78	2,7019	35,32	70,98	76,82	81,34	70,08
7	79,28	2,7492	35,94	70,84	76,67	78,96	67,60
8	73,89	2,8168	36,83	70,73	76,56	78,65	64,97
9	68,07	2,9174	38,14	70,84	76,67	78,14	62,23
10	62,67	3,0538	39,92	71,04	76,89	76,79	59,73
11	58,26	3,2129	42,00	71,48	77,36	76,17	57,95
12	54,81	3,3839	44,24	71,98	77,91	76,97	56,88
13	52,33	3,4881	45,60	72,21	78,15	77,01	55,97
14	49,54	3,5867	46,89	72,35	78,30	77,05	54,87
15	46,47	3,7107	48,51	72,31	78,27	77,20	53,70
16	43,26	3,8720	50,62	72,60	78,58	77,40	52,63
17	40,22	4,1032	53,64	73,45	79,50	77,76	51,97
18	37,71	4,2922	56,11	74,81	80,97	78,82	51,57
19	35,59	4,5111	58,97	73,95	80,04	79,57	51,21
20	32,94	4,9755	65,05	85,35	92,38	75,23	52,13
21	31,98	5,1131	66,85	92,39	100,00	70,82	52,35
22	31,21	5,1976	67,95	90,43	97,88	71,93	52,16
23	30,32	5,2964	69,24	88,42	95,71	72,49	51,88
24	29,57	5,4374	71,09	88,31	95,58	72,45	51,95
25	28,77	5,5299	72,29	87,29	94,47	73,48	51,86
26	28,19	5,6491	73,85	87,41	94,61	71,05	51,70
27	27,46	5,7354	74,98	87,33	94,53	70,78	51,58
28	26,83	5,8576	76,58	86,51	93,63	71,25	51,63
29	26,05	5,9730	78,09	85,88	92,95	71,17	51,54
30	25,39	6,1388	80,25	84,68	91,65	72,06	51,74
31	24,79	6,2957	82,31	80,07	86,67	75,33	51,86
32	24,23	6,4360	84,14	80,02	86,60	73,99	51,88
33	23,58	6,5480	85,60	80,48	87,11	73,04	51,87
34	23,18	6,6649	87,13	81,50	88,21	72,39	52,09
35	22,58	6,7388	88,10	81,33	88,03	72,06	51,98
36	22,09	6,8926	90,11	81,90	88,64	70,25	52,10
37	21,64	7,0393	92,03	82,37	89,15	69,70	52,35
38	21,14	7,1752	93,80	83,91	90,82	67,14	52,43
39	20,69	7,3183	95,67	84,56	91,53	66,43	52,67
40	20,46	7,4540	97,45	85,89	92,96	64,13	52,89
41	20,09	7,5279	98,41	86,09	93,18	63,27	52,88
42	19,72	7,6492	100,00	87,31	94,50	62,71	53,16

De acordo com os parâmetros de otimização adotados, o volume decresce suavemente. A partir da iteração 20, a taxa de remoção por iteração passa a ser menor até o fim da análise. Os deslocamentos neste exemplo sofrem um aumento significativo ao longo de todo o processo,

chegando o deslocamento do modelo da última iteração a quase 3 vezes o deslocamento inicial. Em relação à tensão, um fato interessante acontece na iteração 20. Há um salto expressivo do valor da tensão exatamente na mesma iteração onde os membros superiores são removidos. Esse aumento de tensão permanece por mais algumas iterações, até ocorrer a adaptação do novo modelo sem os membros superiores, ou seja, apenas a parte inferior da estrutura passa, a partir de então, a suportar toda a carga inicial. O índice de performance se mantém ligeiramente irregular próximo a iteração 20.

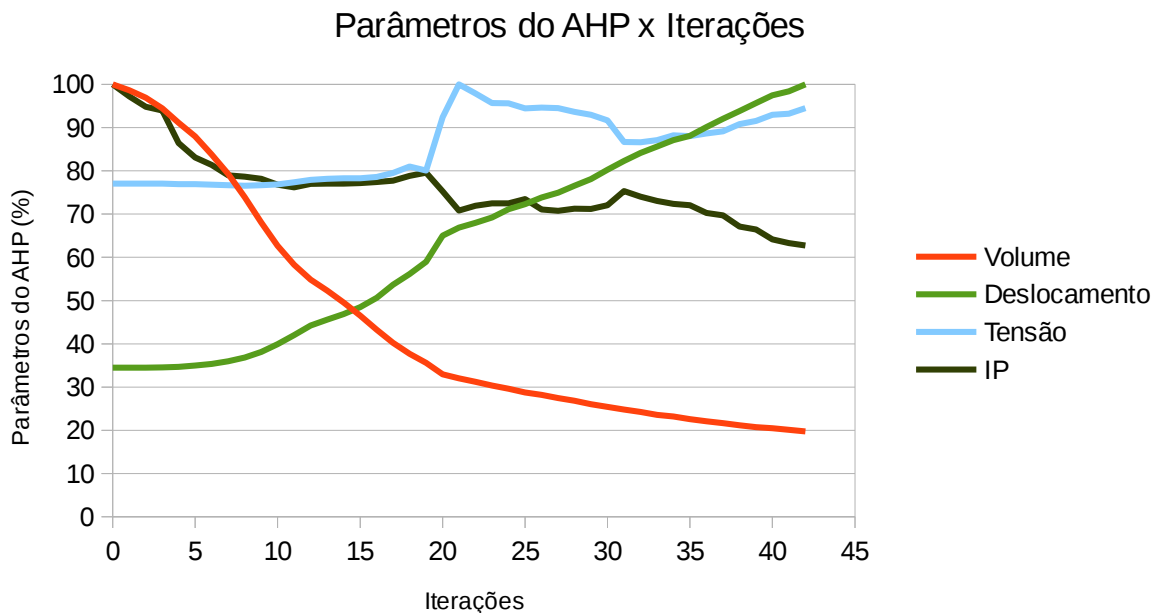


Figura 5.28 – Parâmetros do Vetor WSM (Volume, Deslocamentos, Tensão, Índice de Performance) x Iterações

Para demonstrar a evolução da otimização, a Fig. 5.29 resume o problema, mostrando os valores do vetor WSM ao longo do processo iterativo. O ponto vermelho mostra a iteração com o valor mínimo, que representa a melhor solução para o problema.

Para este modelo, o vetor WSM possui valor mínimo na iteração 19, configurando esta iteração como a solução para o conjunto de possíveis soluções para a análise. Nessa iteração, o volume é de aproximados 35,5% em relação ao volume inicial. É interessante o fato da topologia ótima para este exemplo ser exatamente a última iteração onde os membros superiores ainda existem na análise, concordando com a resposta encontrada por Wang

(2003). Neste ponto, os deslocamentos possuem um aproximado valor médio ligeiramente inferior entre os valores relativos à primeira e última iteração. Esta também é a última iteração onde a tensão mantém valores próximos ao início do modelo.

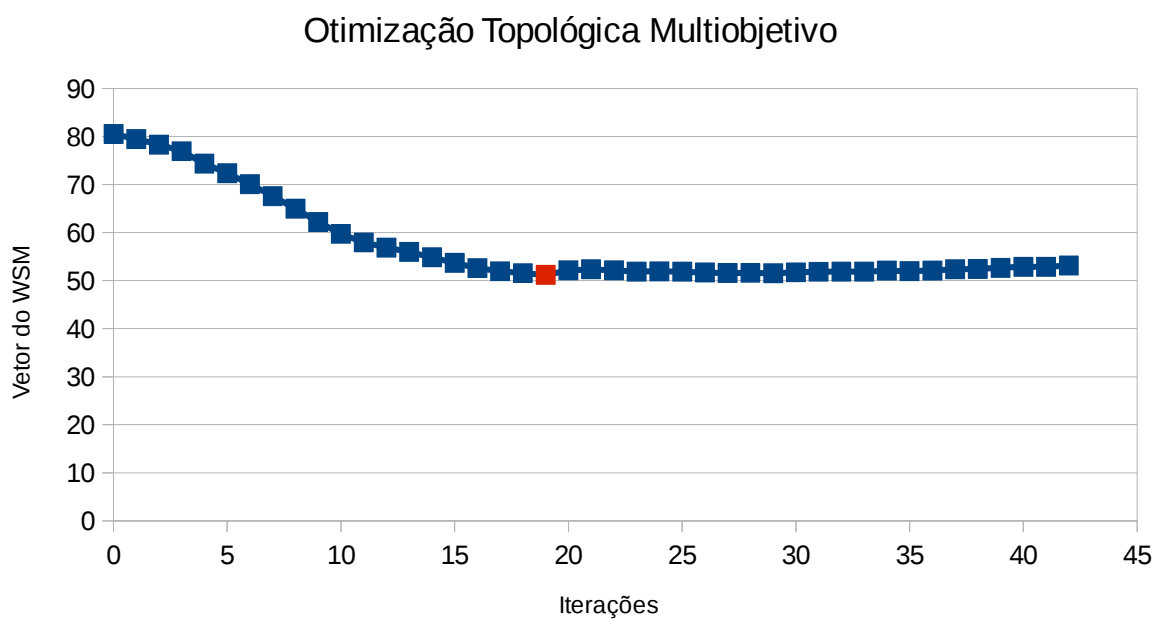


Figura 5.29 – Vetor do WSM x Iterações

5.4 Placa Biapoiada com Duas Cavidades

Este modelo, estudado por Liang *et al.* (2003), consiste em uma placa retangular isostática biapoiada com duas cargas aplicadas na face superior e duas cavidades quadradas igualmente espaçadas, como mostrado na Fig. 5.30.

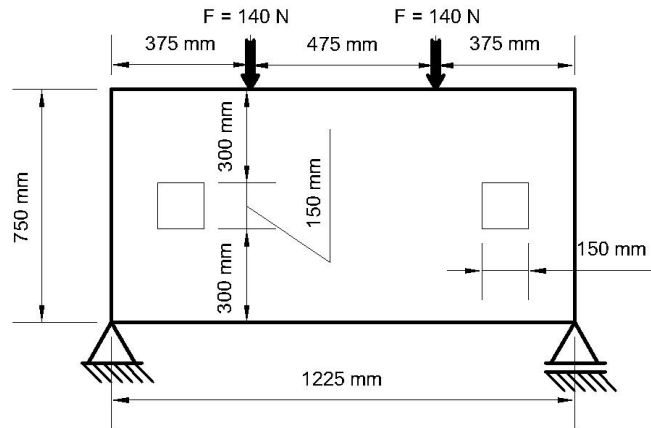


Figura 5.30 – Domínio inicial da placa biapoiada com duas cavidades

Para este problema, os dados pertinentes do modelo são os seguintes:

- Malha utilizada (*mesh*): a malha gerada no Code-Aster® possui 25668 elementos do tipo TETRA4 (Fig. 5.31);
- Módulo de Elasticidade (E): 33048 MPa;
- Coeficiente de Poisson (ν): 0,3;
- Espessura (t): 100 mm;
- Volume máximo desejado (VOLMAX) = 20%;
- Remoção máxima de volume mapeado por iteração (RVOL) = 95% (5% dos elementos são devolvidos para a estrutura);
- Fator de raio de rejeição (FRR) = 1,5%;
- Fator evolucionário de rejeição (FER) = 0,75%.

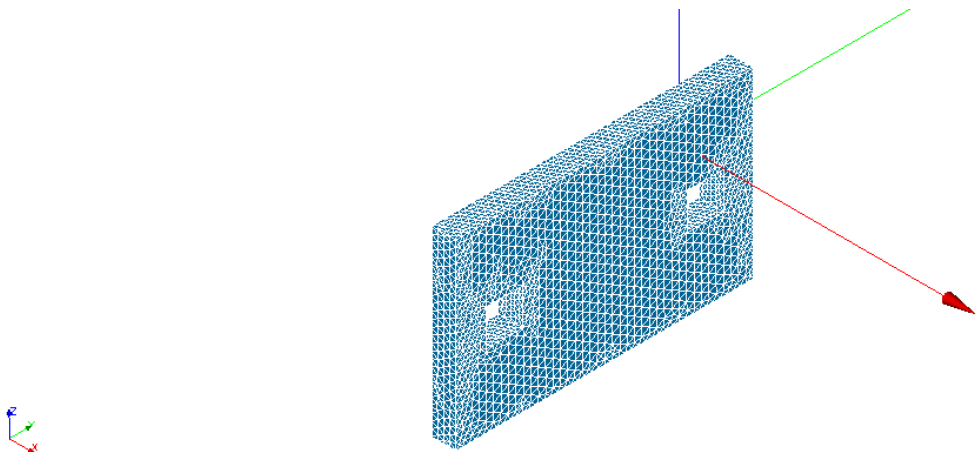


Figura 5.31 – Malha de elementos finitos gerada pelo Salome-Meca®

O processo evolutivo de otimização da placa biapoada com duas cavidades é mostrado nas Figs. 5.32 a 5.34.

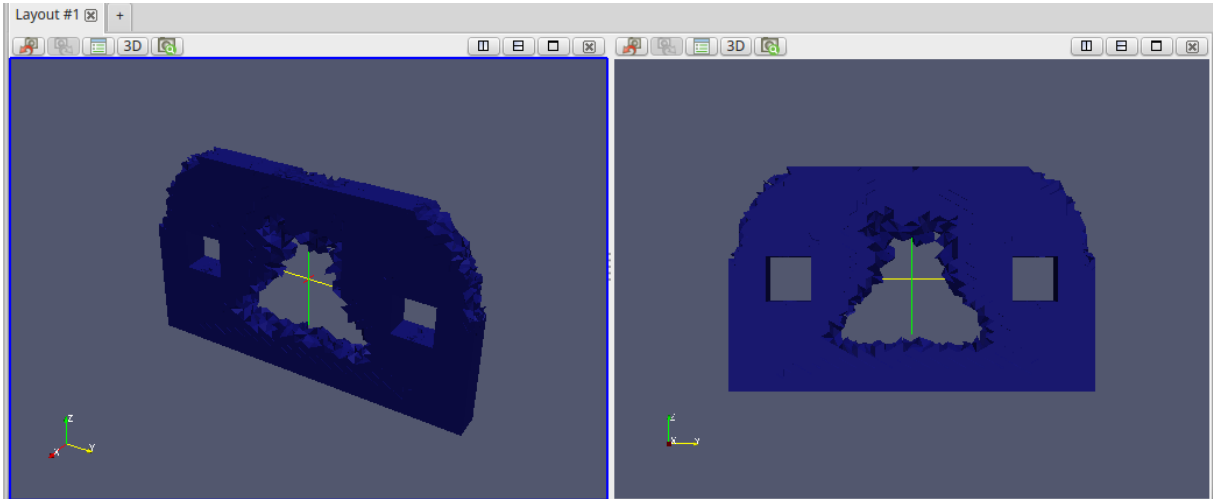


Figura 5.32 – Topologia parcial resultante do passo de otimização 5

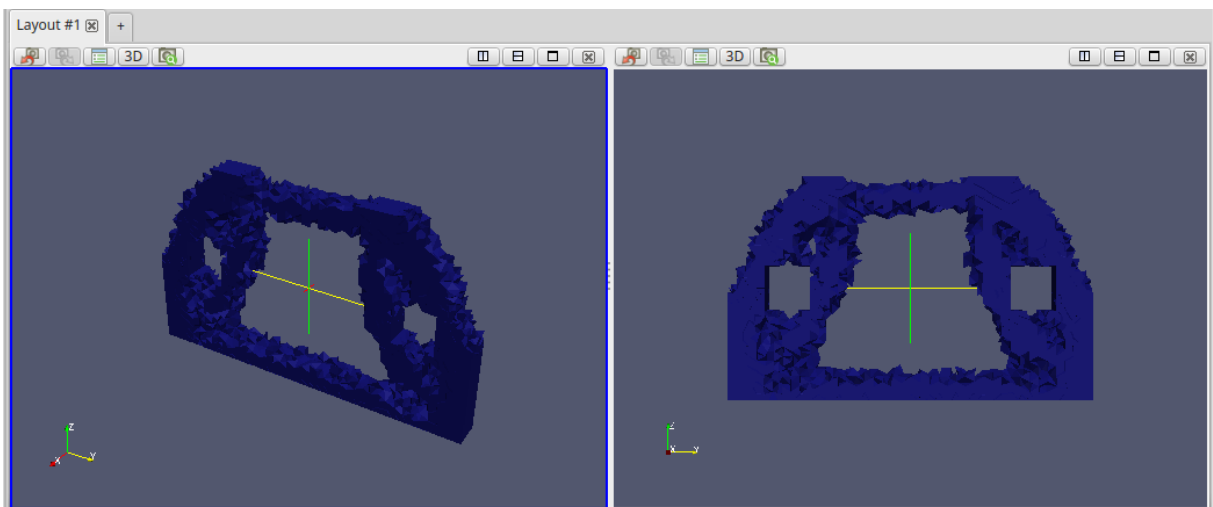


Figura 5.33 – Topologia parcial resultante do passo de otimização 12

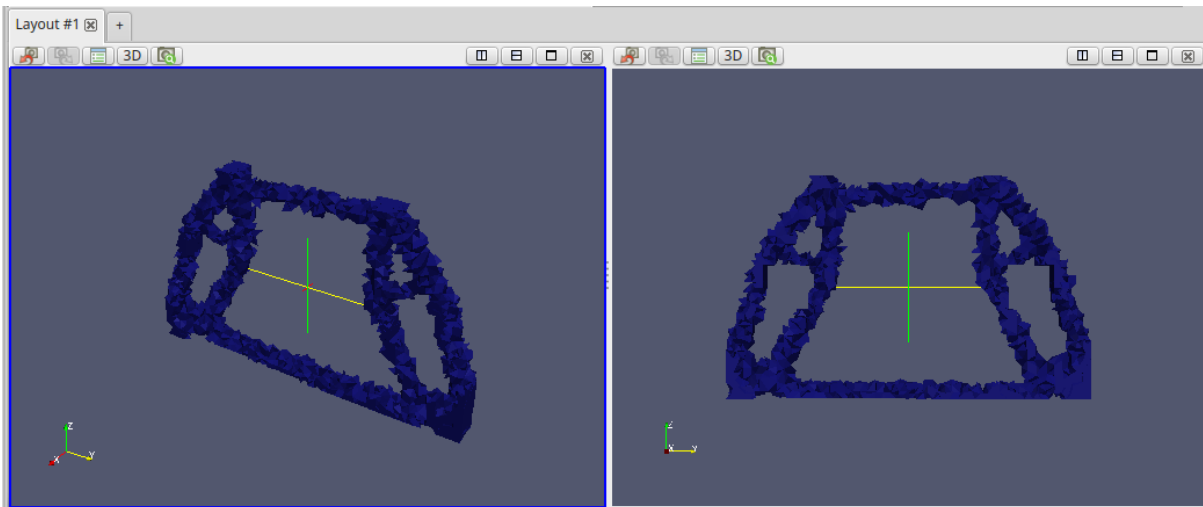


Figura 5.34 – Topologia parcial resultante do passo de otimização 27 (última iteração)

A topologia final ótima obtida para o problema é mostrada na Fig. 5.35.

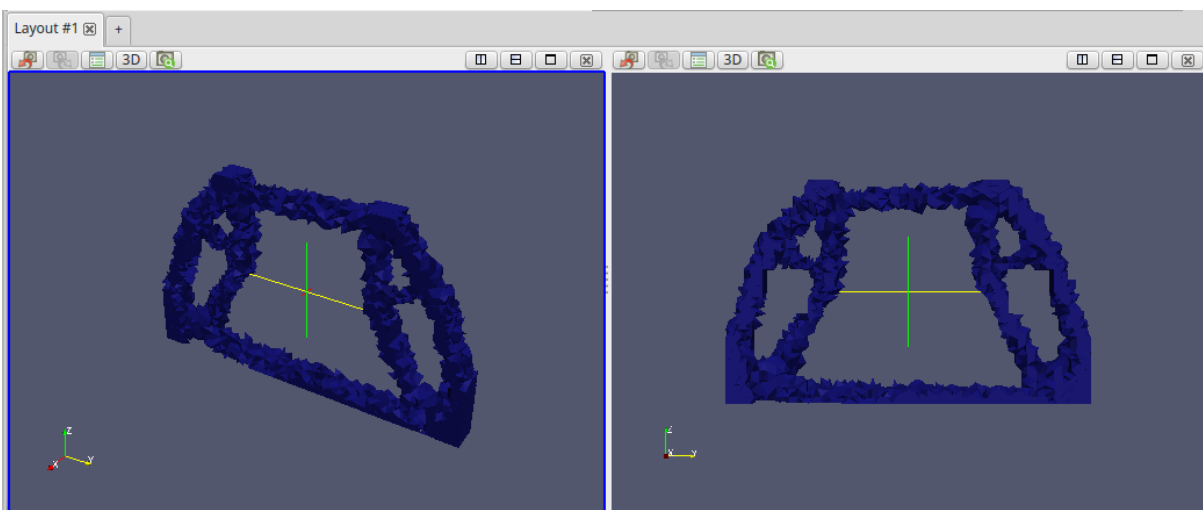
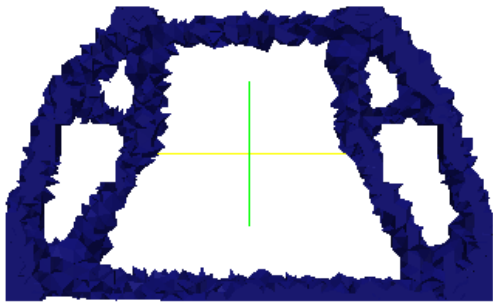


Figura 5.35 – Topologia final ótima resultante do passo de otimização 21

A Fig. 5.36 mostra um comparativo entre a topologia final ótima obtida do modelo proposto pela encontrada por Liang *et al.* (2000).



(a) Formulação SESO

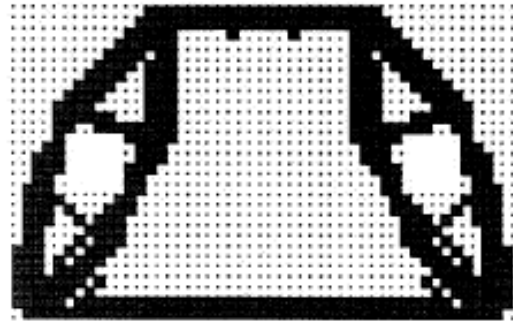
(b) Liang *et al.* (2003)

Figura 5.36 – Topologias finais ótimas para placa biapoiada com duas cavidades

Além disso, a Tabela 5.4 mostra os valores dos parâmetros e do vetor WSM. A linha em destaque nessa tabela marca a iteração e o menor valor do vetor WSM, caracterizando a resposta da melhor topologia obtida pela análise de otimização multiobjetivo. Em seguida, tem-se na Fig. 5.37 um gráfico que descreve o comportamento de cada um destes parâmetros ao longo do processo iterativo.

Tabela 5.4 – Valores dos parâmetros e do Vetor WSM

Iteração	Volume	Deslocamento (cm, %)		Tensão (MPa, %)		IP	WSM
0	100,00	1,25E-04	29,17	234,69	64,83	100,00	77,93
1	95,83	1,25E-04	29,18	234,71	64,83	96,40	77,07
2	90,47	1,25E-04	29,24	234,76	64,85	95,72	76,61
3	85,31	1,26E-04	29,39	234,79	64,85	91,24	75,68
4	80,41	1,27E-04	29,60	234,60	64,80	90,05	75,13
5	75,82	1,29E-04	30,08	235,62	65,08	87,80	74,30
6	71,65	1,31E-04	30,43	235,50	65,05	84,51	73,24
7	67,49	1,33E-04	31,06	235,77	65,13	84,50	72,64
8	63,44	1,37E-04	31,86	237,30	65,55	78,39	71,09
9	59,41	1,40E-04	32,65	237,36	65,56	75,90	69,10
10	55,07	1,44E-04	33,67	236,19	65,24	74,59	66,96
11	50,70	1,49E-04	34,83	236,02	65,20	73,37	65,32
12	47,03	1,60E-04	37,26	239,84	66,25	72,51	64,75
13	43,95	1,69E-04	39,49	242,85	67,08	72,22	64,29
14	41,44	1,89E-04	44,09	250,18	69,10	72,45	64,72
15	39,16	2,00E-04	46,69	254,18	70,21	72,03	64,36
16	36,97	2,09E-04	48,76	257,93	71,25	72,15	63,77
17	34,78	2,28E-04	53,21	269,88	74,55	70,29	63,48
18	32,72	2,40E-04	56,02	276,82	76,46	69,97	62,95
19	30,51	2,59E-04	60,29	286,62	79,17	69,24	62,82
20	28,29	2,74E-04	63,75	284,40	78,56	69,41	62,00
21	26,92	2,79E-04	65,08	282,78	78,11	70,89	61,05
22	25,56	2,99E-04	69,71	298,45	82,44	68,54	61,11
23	24,12	3,13E-04	72,95	320,34	88,49	65,30	61,11
24	22,74	3,39E-04	79,10	338,24	93,43	62,41	61,74
25	21,65	3,63E-04	84,71	353,14	97,55	60,96	62,72
26	20,39	3,95E-04	91,96	357,75	98,82	61,16	64,16
27	19,31	4,29E-04	100,00	362,03	100,00	62,13	65,61

Neste exemplo, até a metade da análise, aproximadamente, o volume decresce em uma taxa praticamente constante. Após a metade, há uma suavização em relação à remoção (penalização) que, também, permanece praticamente constante até o fim do processo de otimização. Os deslocamentos permanecem quase inalterados até a iteração 8, onde começam a crescer consideravelmente. A Fig. 5.32 mostra que os elementos no interior da estrutura sofrem penalizações e, como os elementos entre a carga e os apoios ainda estão bem definidos nessa topologia parcial, não há uma diferença expressiva em relação aos deslocamentos comparados ao modelo inicial. O mesmo acontece com a tensão. Como o carregamento é aplicado próximo das cavidades inicialmente consideradas e os apoios estão nas extremidades inferiores, a tensão pouco varia no início da análise, até que as cavidades oriundas da otimização topológica começam a aparecer (Fig. 5.33). A partir daí, tanto os deslocamentos quanto a tensão máxima ganham um aumento significativo. O índice de performance decresce

até o surgimento das cavidades, mantendo-se constante até próximo da topologia ótima final.

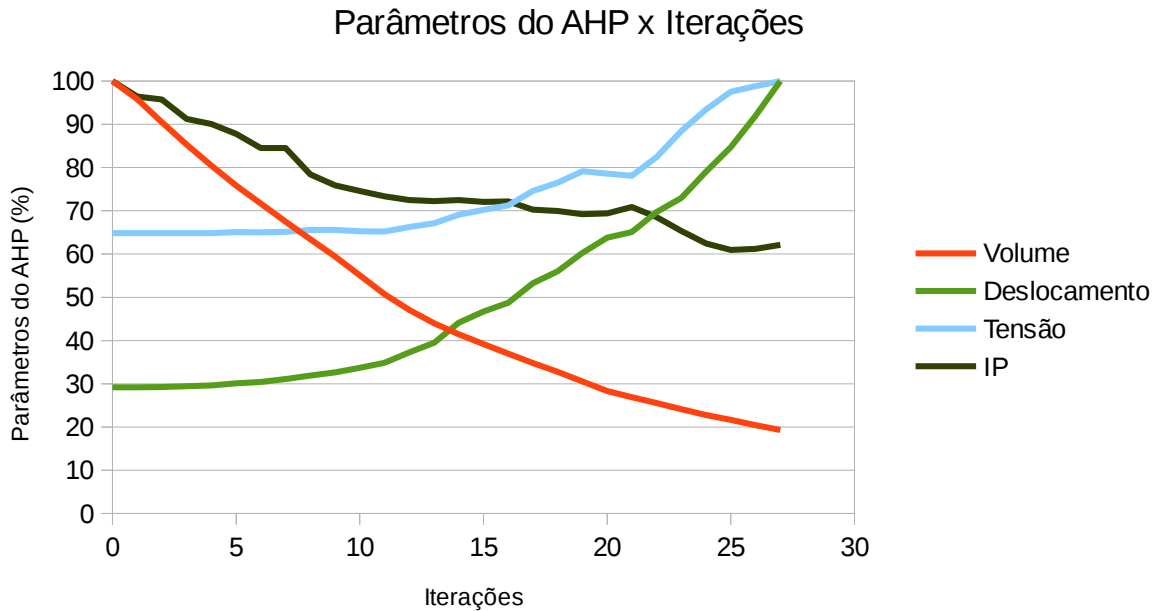


Figura 5.37 – Parâmetros do Vetor WSM (Volume, Deslocamentos, Tensão, Índice de Performance) x Iterações

Para demonstrar a evolução da otimização, a Fig. 5.38 resume o problema, mostrando os valores do vetor WSM ao longo do processo iterativo. O ponto vermelho mostra a iteração com o valor mínimo, que representa a melhor solução para o problema.

Neste exemplo, o vetor WSM possui valor mínimo na iteração 21, configurando esta iteração como a solução para o conjunto de possíveis soluções para a análise. Nessa iteração, o volume é de aproximados 27% em relação ao volume inicial. A tensão nessa iteração é ligeiramente inferior à tensão média relativa a toda a análise. O mesmo vale para os deslocamentos. A topologia obtida se assemelha em relação à obtida por Liang *et al.* (2003), configurando uma adequada topologia ótima.

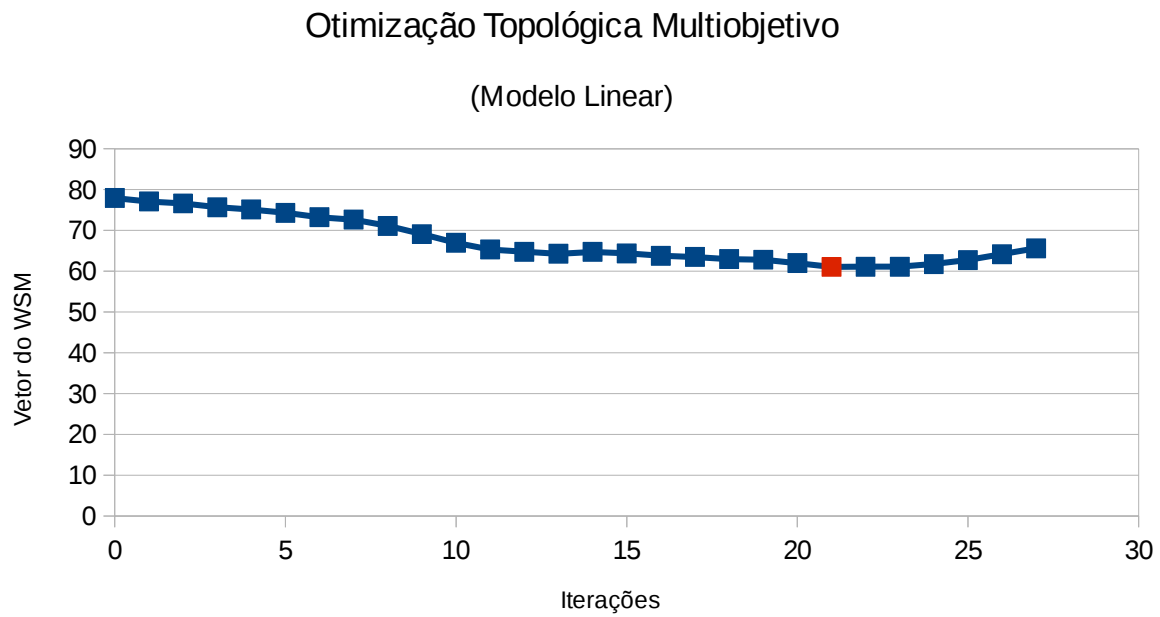


Figura 5.38 – Vetor do WSM x Iterações

5.5 Bloco Apoiado nos Quatro Vértices Inferiores

A otimização deste modelo baseia-se em um modelo original que consiste em um bloco hexaédrico com carga aplicada na face superior e apoiado nos quatro vértices inferiores conforme Fig. 5.39.

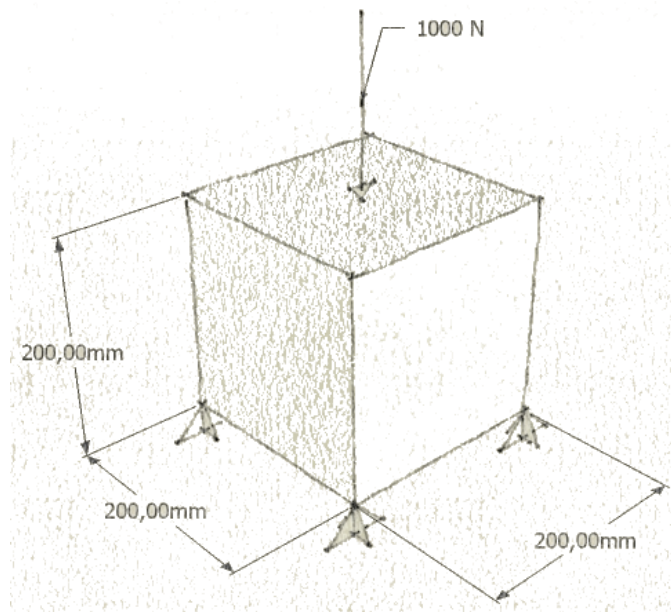


Figura 5.39 – Domínio inicial do bloco apoiado nos quatro vértices inferiores

Para este problema, os dados pertinentes do modelo são os seguintes:

- Malha utilizada (*mesh*): a malha gerada no Code-Aster® possui 29435 elementos do tipo HEXA8 (Fig. 5.40);
- Módulo de Elasticidade (E): 33048 MPa;
- Coeficiente de Poisson (ν): 0,3;
- Volume máximo desejado (VOLMAX) = 10%;
- Remoção máxima de volume mapeado por iteração (RVOL) = 95% (5% dos elementos são devolvidos para a estrutura);
- Fator de raio de rejeição (FRR) = 1,25%;
- Fator evolucionário de rejeição (FER) = 0,75%.

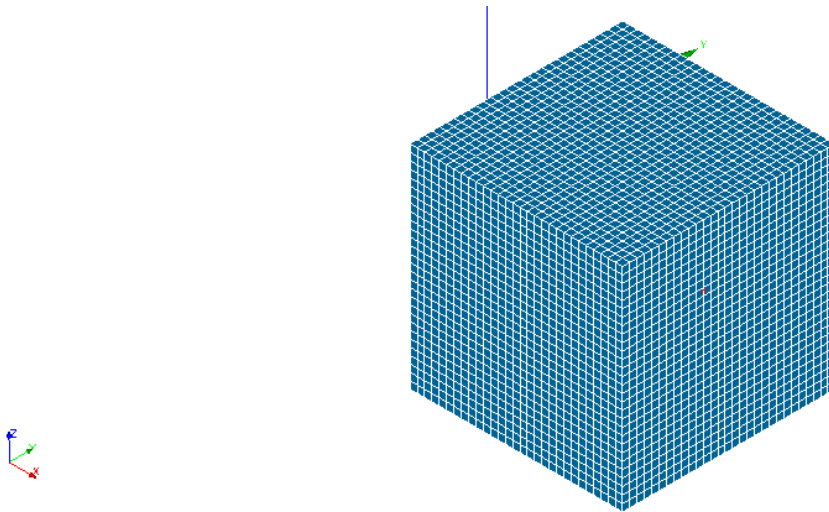


Figura 5.40 – Malha de elementos finitos gerada pelo Salome-Meca®

O processo evolutivo de otimização do bloco engastado nas quatro extremidades é mostrado nas Figs. 5.41 a 5.43.

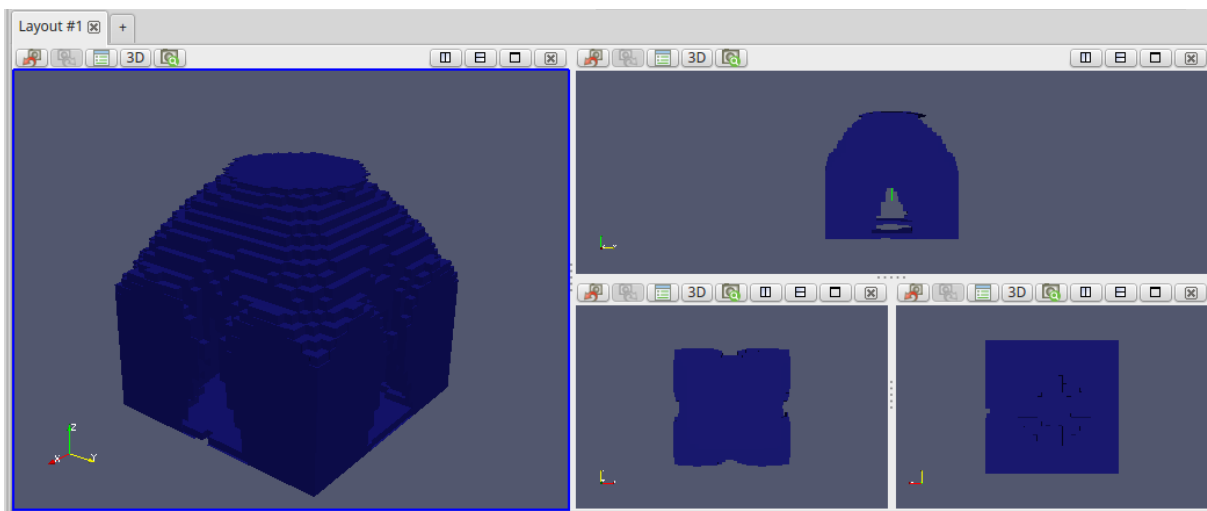


Figura 5.41 – Topologia parcial resultante do passo de otimização 2

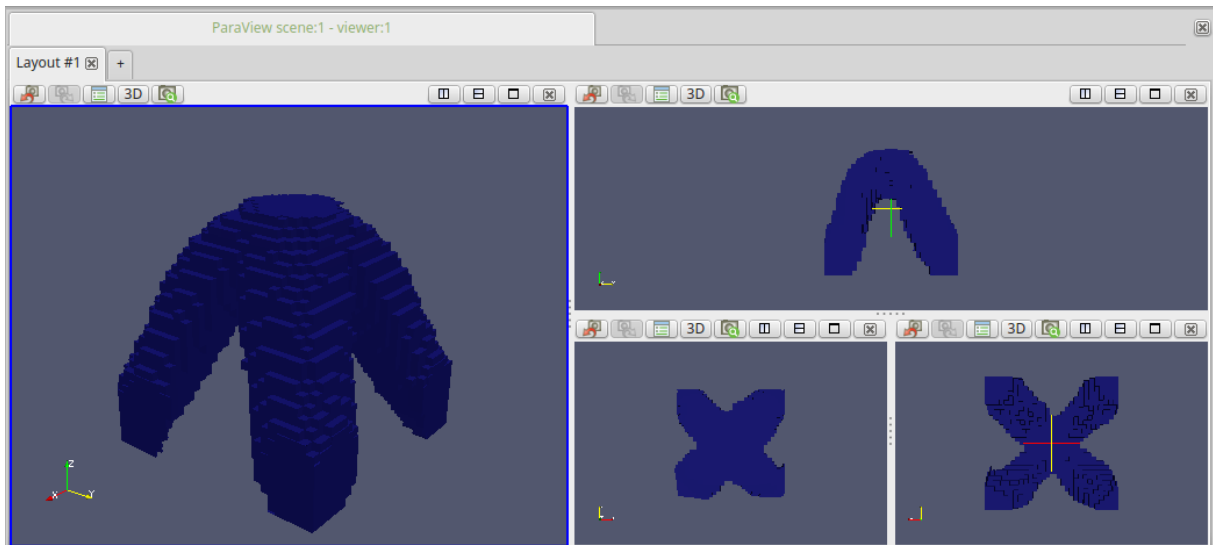


Figura 5.42 – Topologia parcial resultante do passo de otimização 7

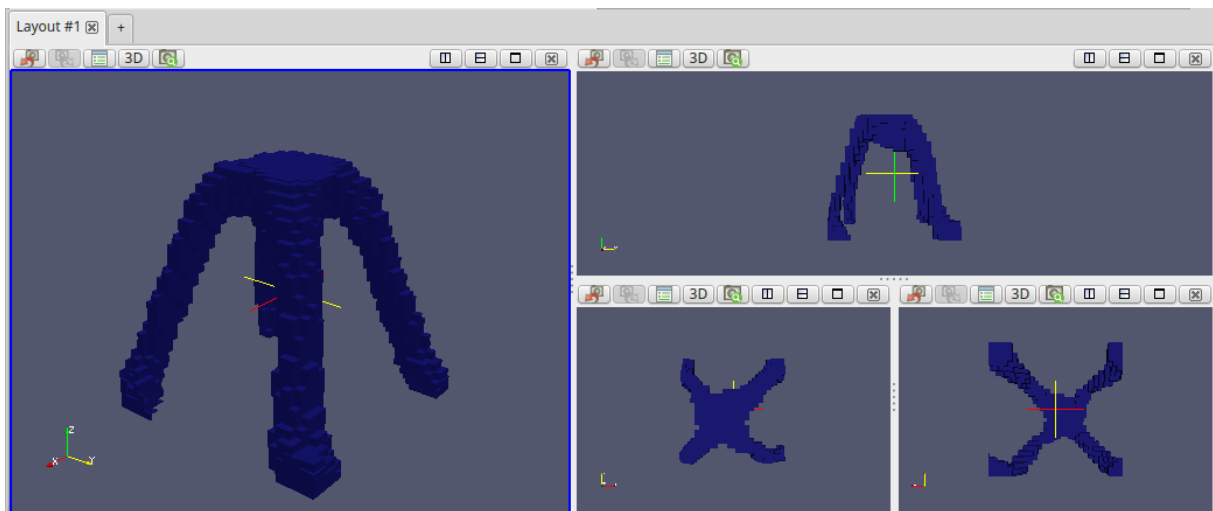


Figura 5.43 – Topologia parcial resultante do passo de otimização 20 (última iteração)

A topologia final ótima obtida para o problema é mostrada na Fig. 5.44.

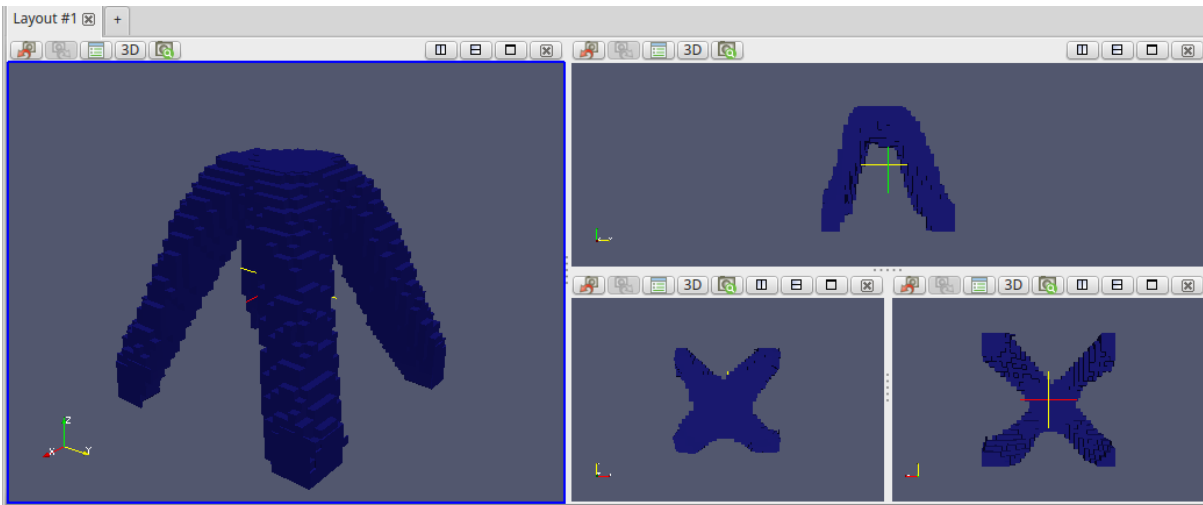


Figura 5.44 – Topologia final ótima resultante do passo de otimização 13

Além disso, a Tabela 5.5 mostra os valores dos parâmetros e do vetor WSM. A linha em destaque nessa tabela marca a iteração e o menor valor do vetor WSM, caracterizando a resposta da melhor topologia obtida pela análise de otimização multiobjetivo. Em seguida, tem-se na Fig. 5.45 um gráfico que descreve o comportamento de cada um destes parâmetros ao longo do processo iterativo.

Tabela 5.5 – Valores dos parâmetros e do Vetor WSM

Iteração	Volume	Deslocamento (cm, %)	Tensão (MPa, %)	IP	WSM		
0	100,00	1,01E-04	13,16	372,53	11,99	100,00	68,50
1	82,19	1,01E-04	13,23	376,20	12,11	99,92	59,21
2	69,86	1,03E-04	13,40	393,57	12,66	98,74	52,72
3	58,13	1,05E-04	13,67	411,54	13,24	96,47	46,45
4	50,15	1,08E-04	14,13	447,88	14,41	91,24	41,91
5	43,50	1,13E-04	14,72	480,70	15,47	84,75	37,94
6	39,14	1,19E-04	15,55	525,57	16,91	78,18	35,27
7	34,93	1,27E-04	16,52	591,73	19,04	76,86	33,38
8	32,02	1,36E-04	17,75	668,60	21,51	67,69	31,36
9	29,50	1,46E-04	19,04	758,35	24,40	63,69	30,20
10	27,00	1,58E-04	20,57	862,19	27,74	63,03	29,56
11	24,79	1,72E-04	22,43	967,87	31,14	49,91	27,71
12	22,66	1,90E-04	24,74	1103,59	35,51	47,19	27,32
13	20,54	2,11E-04	27,48	1245,24	40,07	44,68	27,08
14	18,38	2,39E-04	31,19	1447,69	46,58	43,76	27,47
15	16,36	2,77E-04	36,17	1686,30	54,26	41,17	28,18
16	14,64	3,32E-04	43,35	1924,40	61,92	39,77	29,76
17	13,20	4,07E-04	53,03	2146,77	69,08	42,44	32,56
18	11,76	5,02E-04	65,40	2434,38	78,33	42,71	36,00
19	10,49	6,18E-04	80,60	2732,88	87,94	43,56	40,38
20	9,56	7,67E-04	100,00	3107,69	100,00	44,78	46,32

Neste problema o volume decresce rapidamente, caracterizando que muitos elementos não contribuem efetivamente para definição da topologia estrutural otimizada. É possível perceber claramente a formação dos quatro membros que suportarão toda a carga até o fim do processo de otimização. Na metade da análise o volume se encontra a aproximados 27% do inicial. Este é o único exemplo cujo volume máximo utilizado como critério de parada foi 10% em relação ao inicial, exatamente para mostrar o que acontece com a estrutura com um volume tão baixo. Abaixo dos 20% de volume, a partir da iteração 14, os deslocamentos já aumentam consideravelmente. Da mesma forma, a tensão aumenta a níveis inadequados para uma estrutura, o que configura que, quanto menos elementos, maiores os deslocamentos e maior será a tensão. E ainda, o índice de performance também começa a aumentar, caracterizando respostas inadequadas a partir da iteração 17.

De forma geral, a tensão máxima de von Mises para este modelo tende naturalmente a aumentar consideravelmente seu valor uma vez que os elementos internos são removidos, fazendo com que o “tabuleiro” fique com uma menor espessura a cada passo de otimização.

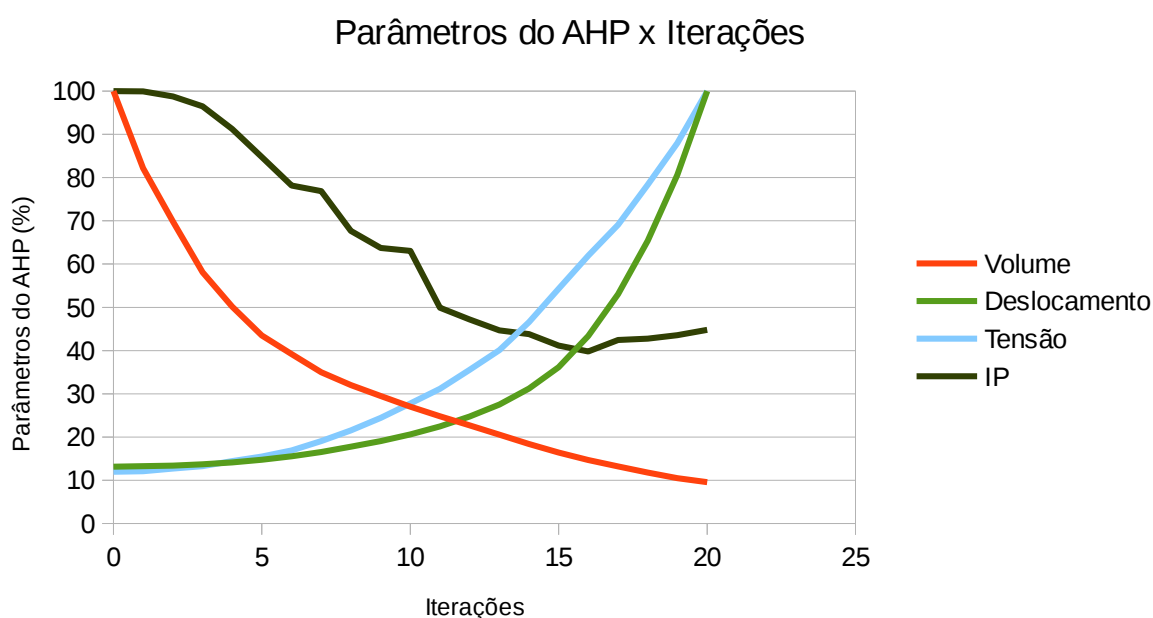


Figura 5.45 – Parâmetros do Vetor WSM (Volume, Deslocamentos, Tensão, Índice de Performance) x Iterações

Para demonstrar a evolução da otimização, a Fig. 5.46 resume o problema, mostrando os

valores do vetor WSM ao longo do processo iterativo. O ponto vermelho mostra a iteração com o valor mínimo, que representa a melhor solução para o problema.

Para este problema, o vetor WSM possui valor mínimo na iteração 13, configurando esta iteração como a solução para o conjunto de possíveis soluções para a análise. Nessa iteração, o volume é de aproximados 20,5% em relação ao volume inicial. A tensão nessa iteração corresponde a cerca de 40% da tensão máxima da análise. O deslocamento é cerca de duas vezes maior que relativo ao inicial. A topologia obtida se assemelha a um banco, sugerindo uma grande economia de material em relação ao modelo inicialmente proposto.

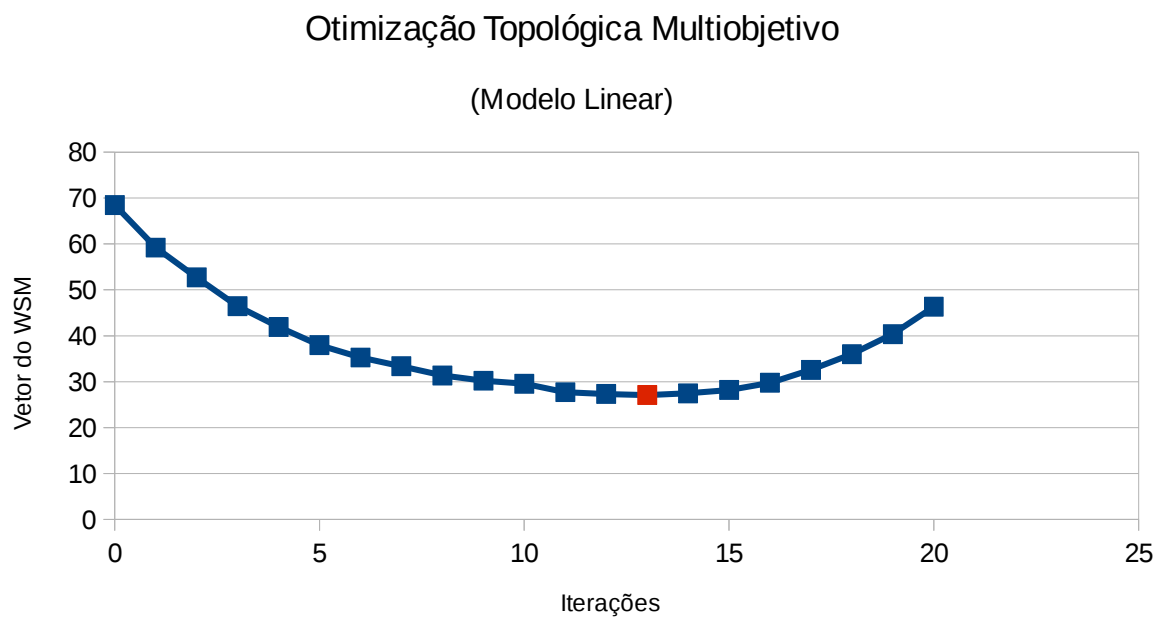


Figura 5.46 – Vetor do WSM x Iterações

5.6 Torre em formato de “T”

Este modelo consiste em uma estrutura sólida em formato de “T” suportado na base com quatro cargas concentradas em sua face superior, como mostrado na Fig 5.47.

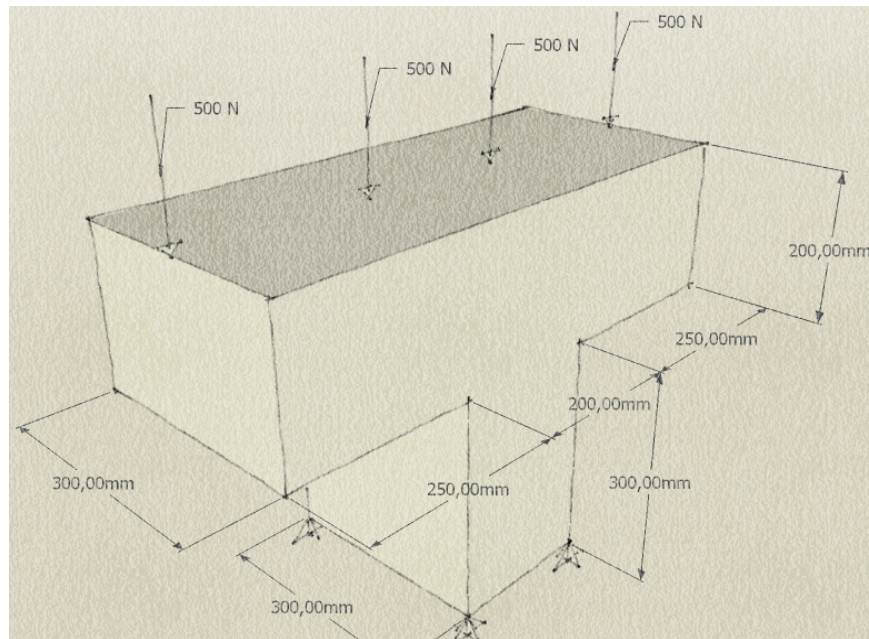


Figura 5.47 – Domínio inicial da torre em formato de “T”

Para este problema, os dados pertinentes do modelo são os seguintes:

- Malha utilizada (*mesh*): a malha gerada no Code-Aster® possui 26843 elementos do tipo TETRA4 (Fig. 5.48);
- Módulo de Elasticidade (E): 210 GPa;
- Coeficiente de Poisson (ν): 0,3;
- Volume máximo desejado (VOLMAX) = 20%;
- Remoção máxima de volume mapeado por iteração (RVOL) = 95% (5% dos elementos são devolvidos para a estrutura);
- Fator de raio de rejeição (FRR) = 1,75%;
- Fator evolucionário de rejeição (FER) = 0,95%.

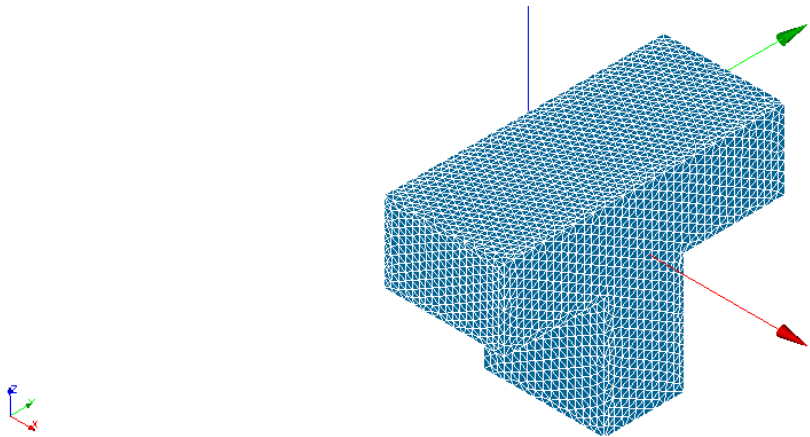


Figura 5.48 – Malha de elementos finitos gerada pelo Salome-Meca®

O processo evolutivo de otimização da torre em formato de “T” é mostrado nas Figs. 5.49 a 5.51.

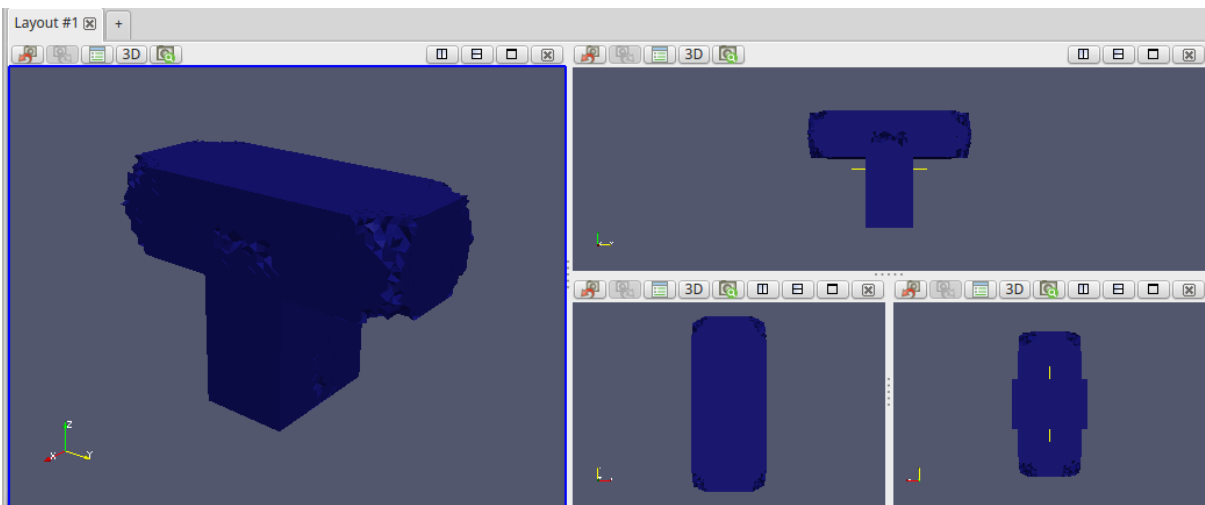


Figura 5.49 – Topologia parcial resultante do passo de otimização 2

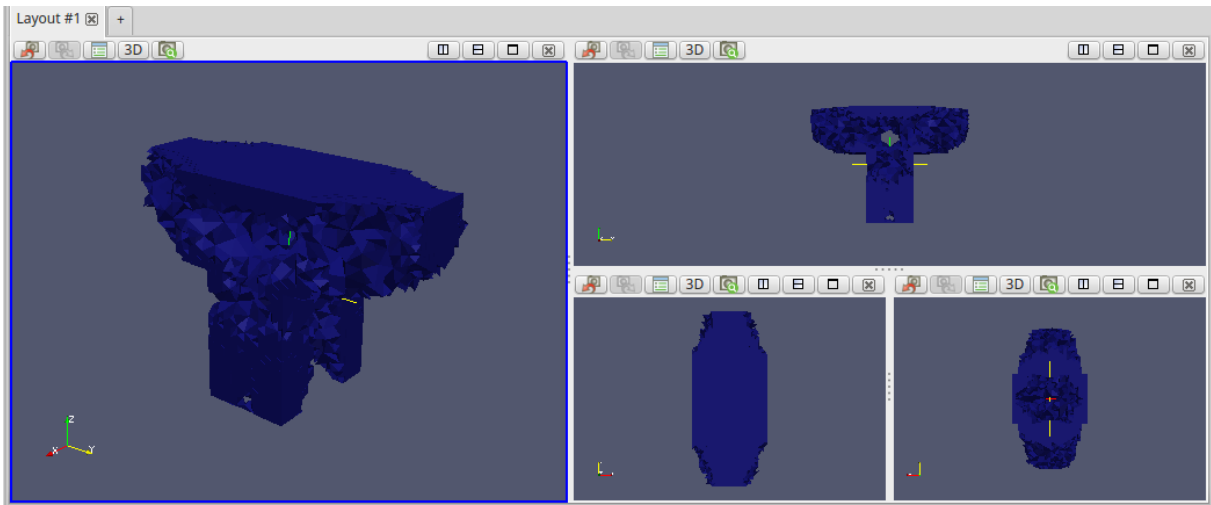


Figura 5.50 – Topologia parcial resultante do passo de otimização 6

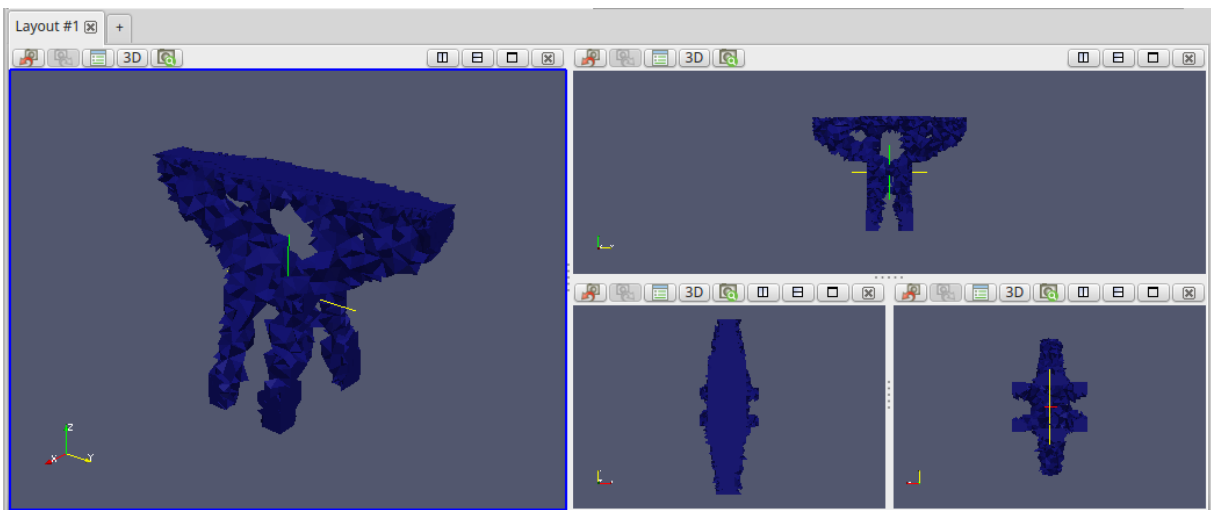


Figura 5.51 – Topologia parcial resultante do passo de otimização 20 (última iteração)

A topologia final ótima obtida para o problema é mostrada na Fig. 5.52.

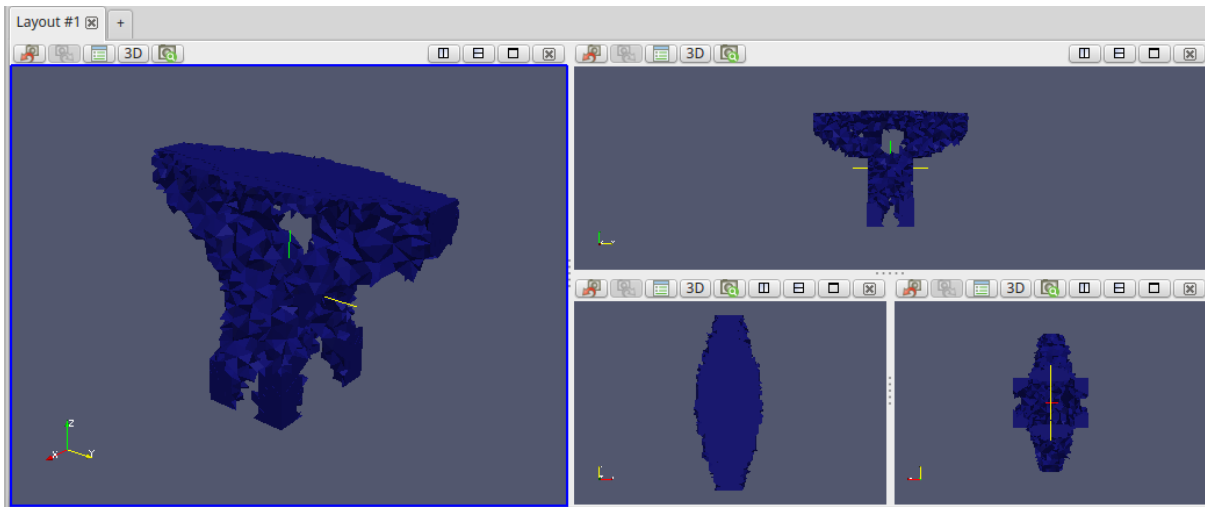


Figura 5.52 – Topologia final ótima resultante do passo de otimização 13

A Tabela 5.6 mostra os valores dos parâmetros e do vetor WSM. A linha em destaque nessa tabela marca a iteração e o menor valor do vetor WSM, caracterizando a resposta da melhor topologia obtida pela análise de otimização multiobjetivo. Em seguida, tem-se na Fig. 5.53 um gráfico que descreve o comportamento de cada um destes parâmetros ao longo do processo iterativo.

Tabela 5.6 – Valores dos parâmetros e do Vetor WSM

Iteração	Volume	Deslocamento (cm, %)		Tensão (MPa, %)		IP	WSM
0	100,00	0,3670	35,94	347,46	67,97	100,00	84,14
1	97,45	0,3671	35,95	347,46	67,97	98,20	82,53
2	92,34	0,3677	36,01	347,50	67,98	98,11	79,86
3	85,44	0,3701	36,24	352,17	68,89	93,34	75,66
4	76,53	0,3756	36,78	353,73	69,20	92,35	71,01
5	67,12	0,3887	38,07	349,18	68,31	90,50	66,05
6	58,02	0,4084	40,00	348,80	68,23	89,96	61,71
7	50,19	0,4441	43,49	354,07	69,26	88,10	58,34
8	44,37	0,4843	47,43	351,54	68,77	86,46	56,03
9	39,53	0,5290	51,80	344,62	67,42	87,61	54,69
10	35,90	0,5713	55,95	337,38	66,00	93,15	54,61
11	33,14	0,6223	60,94	363,08	71,03	87,61	54,11
12	30,44	0,6642	65,05	388,67	76,03	83,78	53,67
13	28,66	0,7009	68,64	400,09	78,27	81,41	53,53
14	27,04	0,7473	73,19	413,37	80,86	78,23	53,63
15	25,28	0,7921	77,57	435,27	85,15	73,53	53,55
16	24,08	0,8508	83,33	462,94	90,56	68,55	54,18
17	22,86	0,8818	86,36	474,10	92,74	69,91	54,77
18	21,72	0,9117	89,29	490,50	95,95	66,52	54,73
19	20,80	0,9765	95,63	511,19	100,00	63,09	55,77
20	19,83	1,0211	100,00	507,27	99,23	66,06	56,80

O volume deste modelo decai rapidamente já nas primeiras iterações, chegando na metade do seu valor inicial na iteração 7. Uma suavização na penalização acontece a partir desse ponto, até o fim da análise. Ainda nessa iteração, os deslocamentos permanecem sem grandes variações até esse ponto, onde começam a aumentar praticamente linearmente. O maior valor de todo o processo de otimização chega a perto de três vezes o deslocamento inicial. A tensão permanece quase constante até a iteração 10, onde começa a aumentar significativamente. Isso ocorre devido ao aumento da cavidade interna iniciado próximo à iteração 6 (Fig. 5.50) e, também, pela formação da base de sustentação com quatro membros, uma vez que os elementos internos em relação a esses novos membros que contribuíam para o modelo são penalizados. Nessa mesma iteração 10, o índice de performance tem um pico, que representa um mau aproveitamento da estrutura em relação à tensão máxima naquele instante. Nas iterações seguintes, o parâmetro volta a apresentar uma ligeira queda, com poucas variações até o fim da análise.

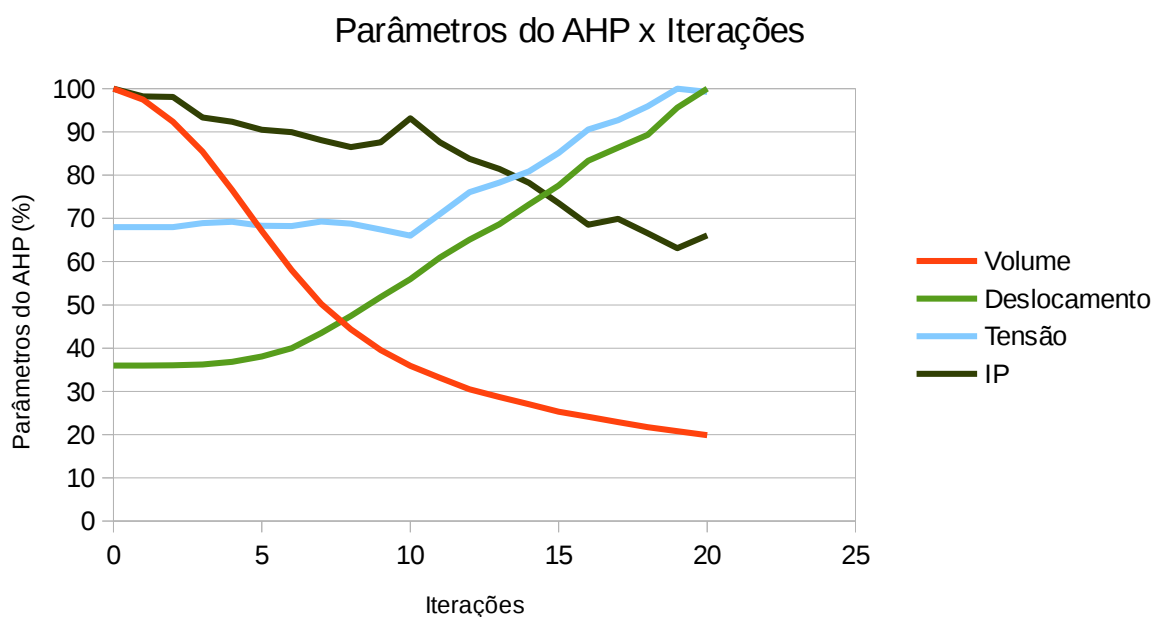


Figura 5.53 – Parâmetros do Vetor WSM (Volume, Deslocamentos, Tensão, Índice de Performance) x Iterações

Para demonstrar a evolução da otimização, a Fig. 5.54 resume o problema, mostrando os valores do vetor WSM ao longo do processo iterativo. O ponto vermelho mostra a iteração com o valor mínimo, que representa a melhor solução para o problema.

Neste exemplo, o vetor WSM possui valor mínimo na iteração 13, configurando esta iteração como a solução para o conjunto de possíveis soluções para a análise. Nessa iteração, o volume é de aproximados 30% em relação ao volume inicial. A tensão nessa iteração é certa de 1,3 vezes menor que a máxima tensão em relação a todo o processo de otimização. O deslocamento nesse ponto é praticamente o deslocamento médio da análise e o índice de performance se mantém nos 84%.

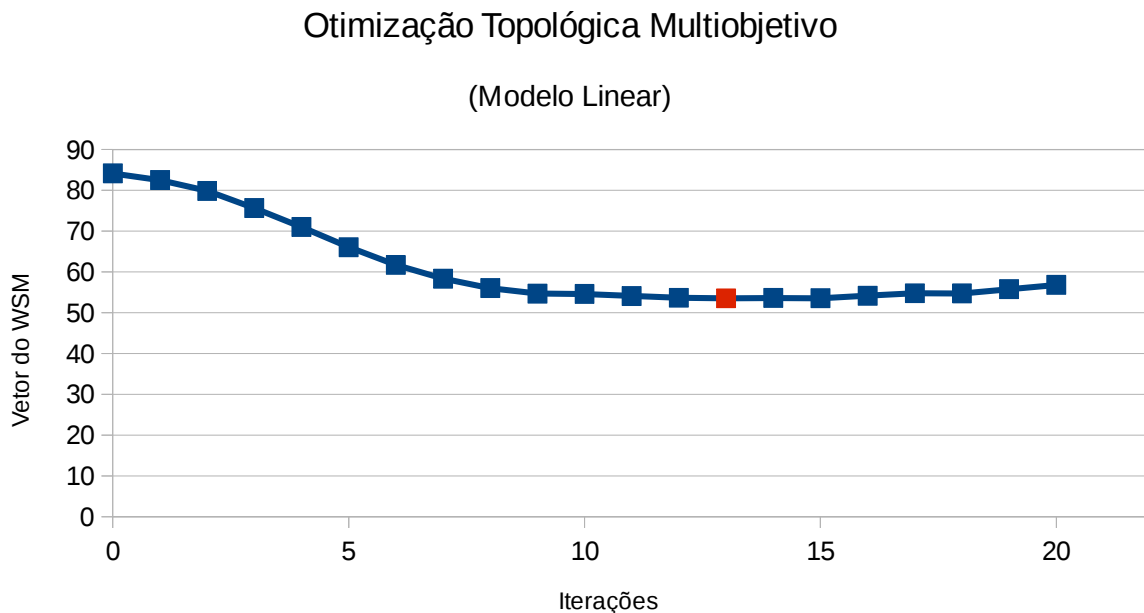


Figura 5.54 – Vetor do WSM x Iterações

5.7 Bloco com Balanços Simétricos e Furo Central

Este modelo consiste em um bloco com balanços simétricos e furo central, suportado na base com quatro cargas concentradas em sua face superior, como mostrado na Fig 5.55.

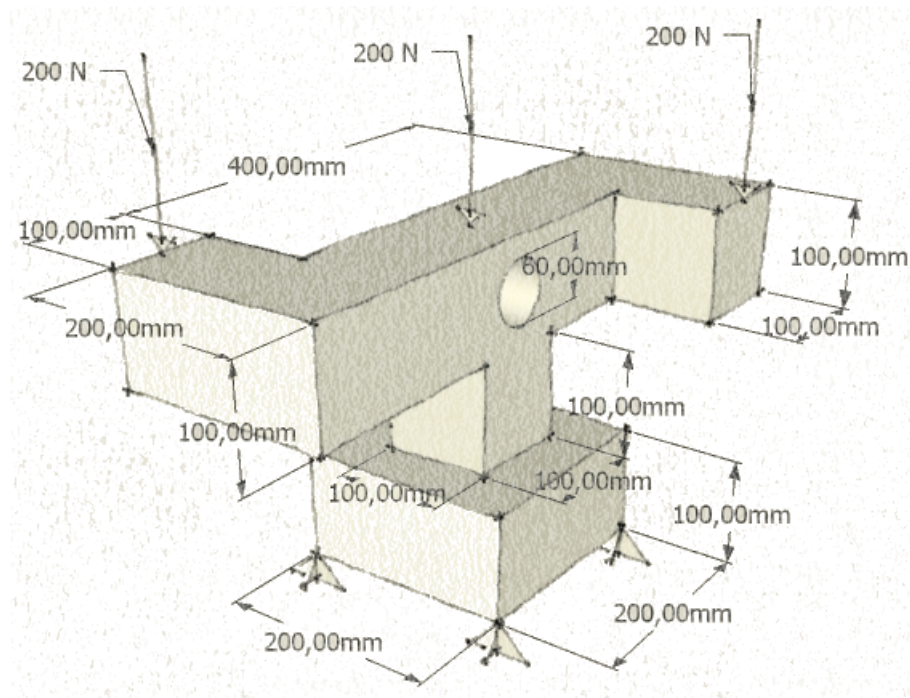


Figura 5.55 – Domínio inicial do bloco com balanços simétricos e furo central

Para este problema, os dados pertinentes do modelo são os seguintes:

- Malha utilizada (*mesh*): a malha gerada no Code-Aster® possui 28257 elementos do tipo TETRA4 (Fig. 5.56);
- Módulo de Elasticidade (E): 210 GPa;
- Coeficiente de Poisson (ν): 0,3;
- Volume máximo desejado (VOLMAX) = 20%;
- Remoção máxima de volume mapeado por iteração (RVOL) = 95% (5% dos elementos são devolvidos para a estrutura);
- Fator de raio de rejeição (FRR) = 1,75%;
- Fator evolucionário de rejeição (FER) = 0,75%.

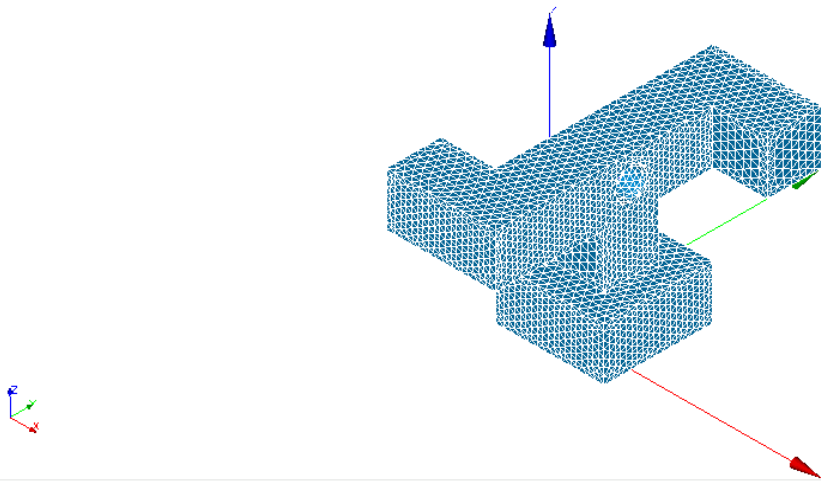


Figura 5.56 – Malha de elementos finitos gerada pelo Salome-Meca®

O processo evolutivo de otimização da torre em formato de “T” é mostrado nas Figs. 5.57 a 5.59.

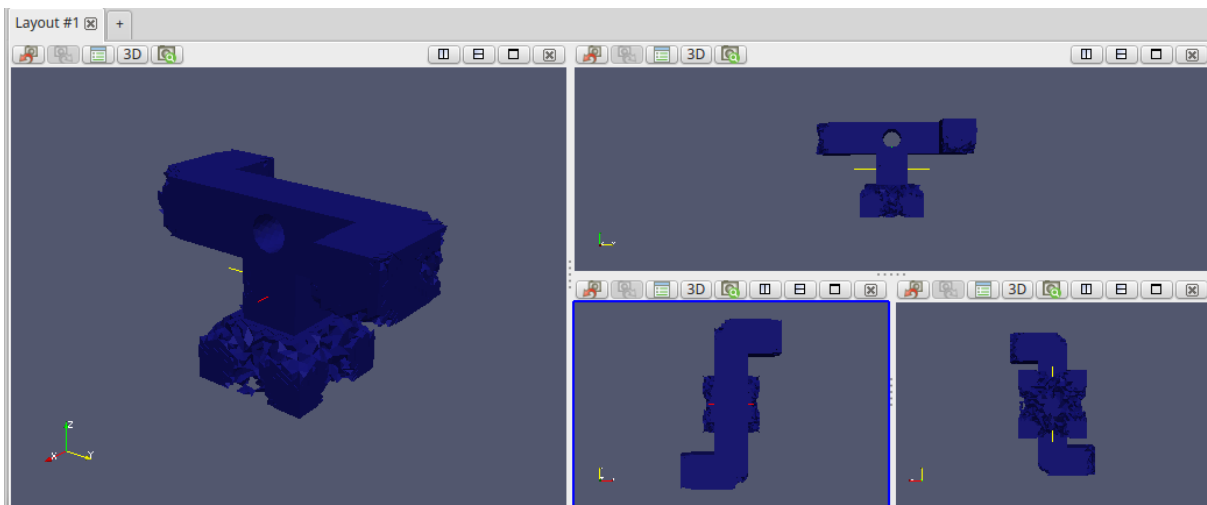


Figura 5.57 – Topologia parcial resultante do passo de otimização 3

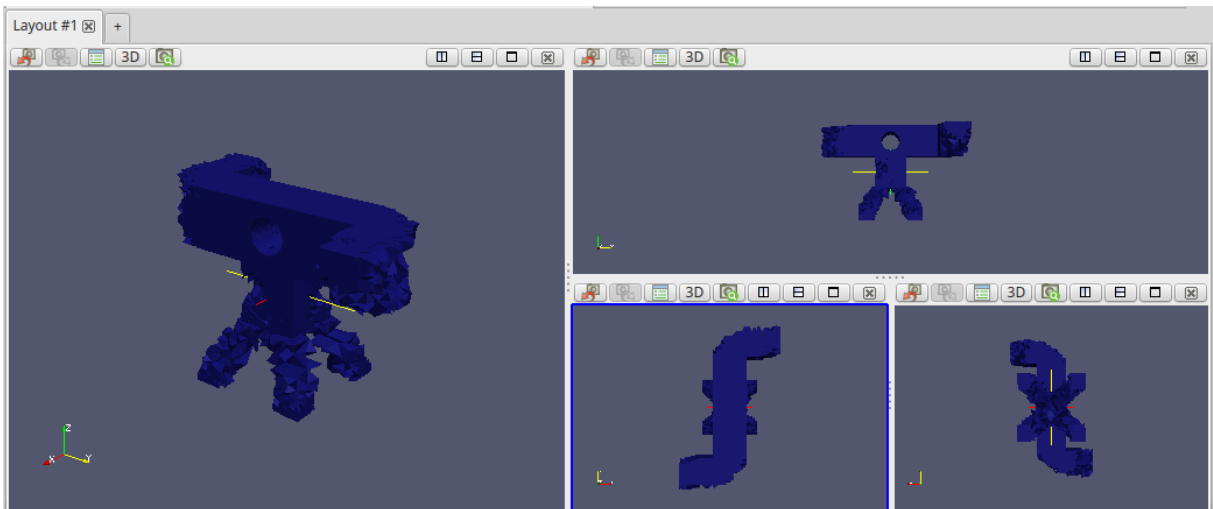


Figura 5.58 – Topologia parcial resultante do passo de otimização 10

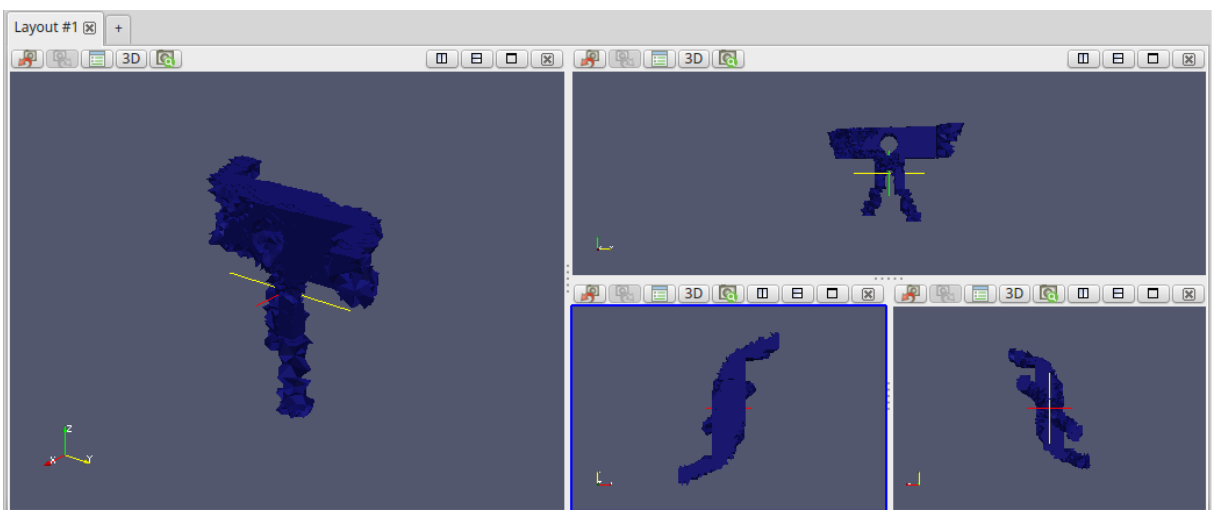


Figura 5.59 – Topologia parcial resultante do passo de otimização 25 (última iteração)

A topologia final ótima obtida para o problema é mostrada na Fig. 5.60.

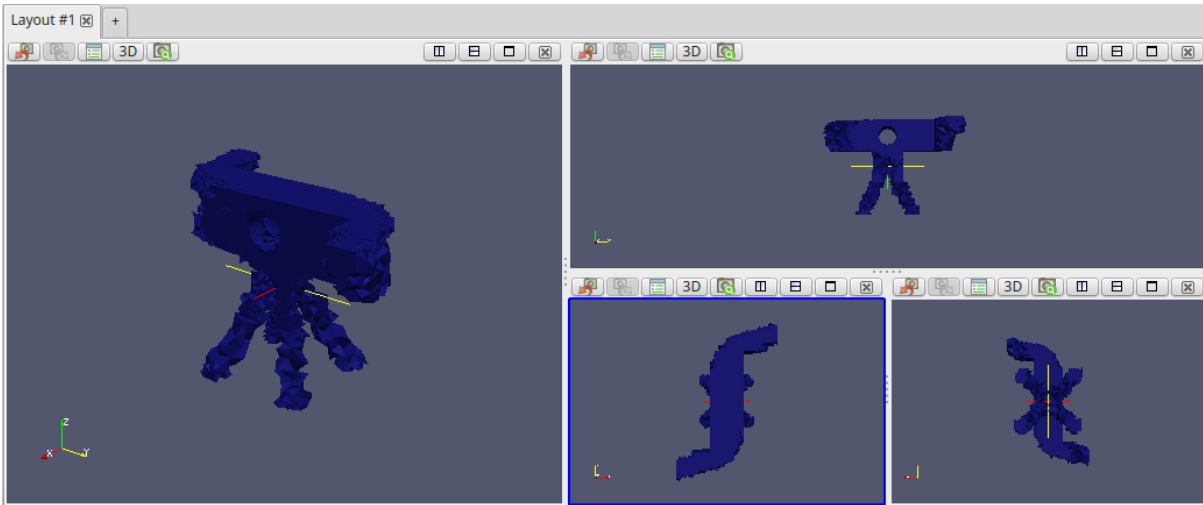


Figura 5.60 – Topologia final ótima resultante do passo de otimização 18

A Tabela 5.7 mostra os valores dos parâmetros e do vetor WSM. A linha em destaque nessa tabela marca a iteração e o menor valor do vetor WSM, caracterizando a resposta da melhor topologia obtida pela análise de otimização multiobjetivo. Em seguida, tem-se na Fig. 5.61 um gráfico que descreve o comportamento de cada um destes parâmetros ao longo do processo iterativo.

Tabela 5.7 – Valores dos parâmetros e do Vetor WSM

Iteração	Volume	Deslocamento (cm, %)		Tensão (MPa, %)		IP	WSM
0	100,00	1,0494	40,93	765,61	68,72	100,00	81,40
1	96,68	1,0495	40,93	765,61	68,72	95,41	79,13
2	89,79	1,0503	40,97	765,61	68,72	95,09	75,50
3	80,99	1,0537	41,10	765,59	68,72	87,10	70,01
4	75,33	1,0619	41,42	765,40	68,70	83,36	66,70
5	70,41	1,0681	41,66	765,31	68,70	80,49	63,86
6	66,20	1,0725	41,83	765,20	68,69	69,86	60,47
7	63,08	1,0920	42,59	765,38	68,70	65,35	58,52
8	59,73	1,1058	43,13	765,31	68,70	64,42	56,80
9	56,56	1,1192	43,65	765,10	68,68	64,82	55,32
10	53,88	1,1442	44,63	764,28	68,60	65,43	54,24
11	51,08	1,1630	45,36	763,48	68,53	65,27	52,94
12	48,91	1,1978	46,72	762,64	68,46	67,12	52,37
13	46,65	1,2345	48,15	762,19	68,42	67,20	51,57
14	44,20	1,2610	49,18	761,02	68,31	67,23	50,56
15	41,91	1,3469	52,53	762,16	68,41	65,76	50,08
16	39,74	1,3801	53,83	765,88	68,75	64,18	49,13
17	37,49	1,4168	55,26	775,93	69,65	61,21	48,08
18	35,02	1,5335	59,81	785,71	70,53	60,27	47,96
19	32,76	1,6493	64,33	794,03	71,27	64,48	48,52
20	30,17	1,7158	66,92	812,50	72,93	65,04	48,08
21	26,69	1,9239	75,04	821,16	73,71	71,73	49,24
22	23,94	2,0684	80,67	881,34	79,11	74,14	50,10
23	22,30	2,2708	88,57	969,56	87,03	69,61	51,57
24	20,59	2,3592	92,01	1032,78	92,71	68,77	52,04
25	19,09	2,5639	100,00	1114,05	100,00	67,48	53,93

Neste exemplo o volume decresce quase linearmente, de forma geral. Um fato interessante deste modelo é que, apesar de uma das cargas estar aplicada exatamente na mesma direção do centro do furo na face superior do modelo, o furo se mantém bem definido durante toda a análise, ou seja, praticamente nenhum elemento do seu entorno é penalizado. Quanto aos deslocamentos, o comportamento é similar ao do modelo anterior, com pouca variação nas primeiras iterações, seguido de um aumento gradual até o fim. Este aumento é da ordem de 2,5 vezes, considerando a primeira e a última iteração. A tensão deste exemplo merece certo destaque por se manter praticamente constante até a iteração 17, onde começa a aumentar significativamente. Neste ponto, os quatro membros da base que suportam toda a carga já estão bem definidos, como é possível ver na Fig. 5.60, que representa a topologia ótima do problema. Quando há a penalização dos membros inferiores tornando a estrutura hipostática (Fig. 5.59), a tensão já possui um nível elevado em relação ao inicial (cerca de 1,45 vezes). Além disso, o índice de performance começa a subir enquanto a tensão aumenta, indicando

que a estrutura começa a se tornar inadequada.

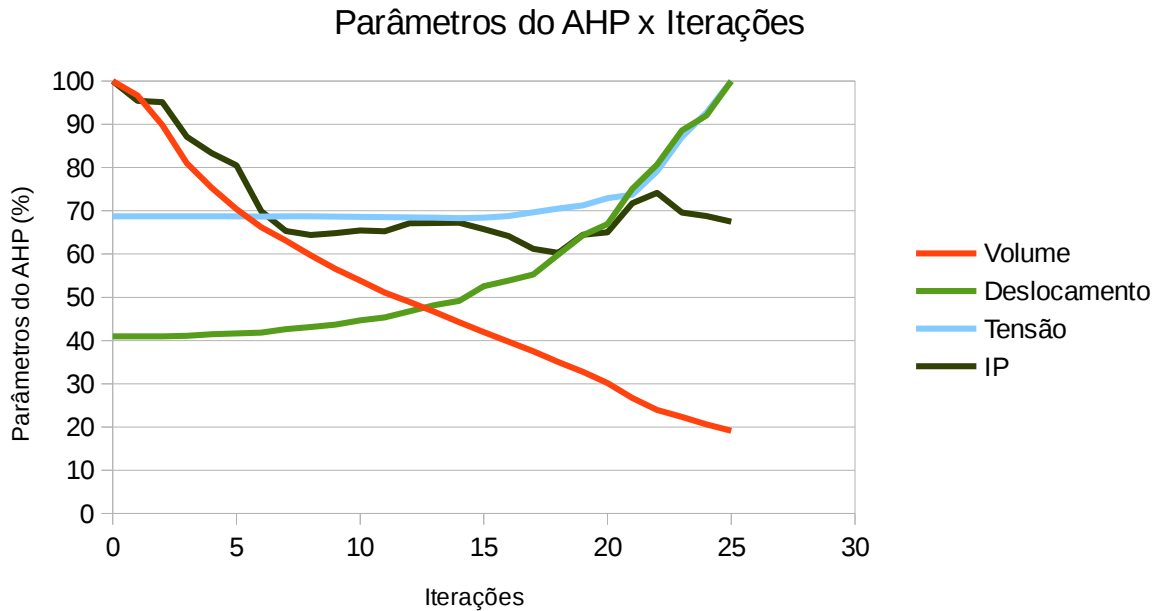


Figura 5.61 – Parâmetros do Vetor WSM (Volume, Deslocamentos, Tensão, Índice de Performance) x Iterações

Para demonstrar a evolução da otimização, a Fig. 5.62 resume o problema, mostrando os valores do vetor WSM ao longo do processo iterativo. O ponto vermelho mostra a iteração com o valor mínimo, que representa a melhor solução para o problema.

Neste exemplo, o vetor WSM possui valor mínimo na iteração 18, configurando esta iteração como a solução para o conjunto de possíveis soluções para a análise. Nessa iteração, o volume é de aproximados 35% em relação ao volume inicial. A tensão nessa iteração é ainda próxima da tensão inicial, antes do aumento devido à penalização de elementos dos membros inferiores. O índice de performance apresenta o menor valor em toda a análise, garantindo a melhor topologia face a todas as respostas parciais.

Otimização Topológica Multiobjetivo

(Modelo Linear)

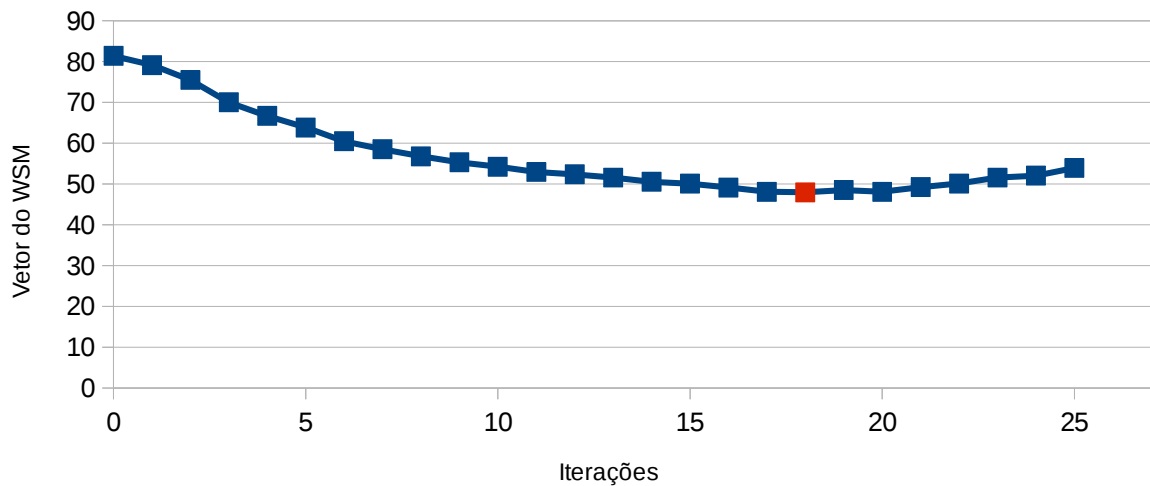


Figura 5.62 – Vetor do WSM x Iterações

5.8 Quadro de Bicicleta

Este exemplo consiste em uma estrutura estudada por Lin e Hsu (2008) que representa um quadro de bicicleta, cujas condições iniciais são mostradas na Fig. 5.63.

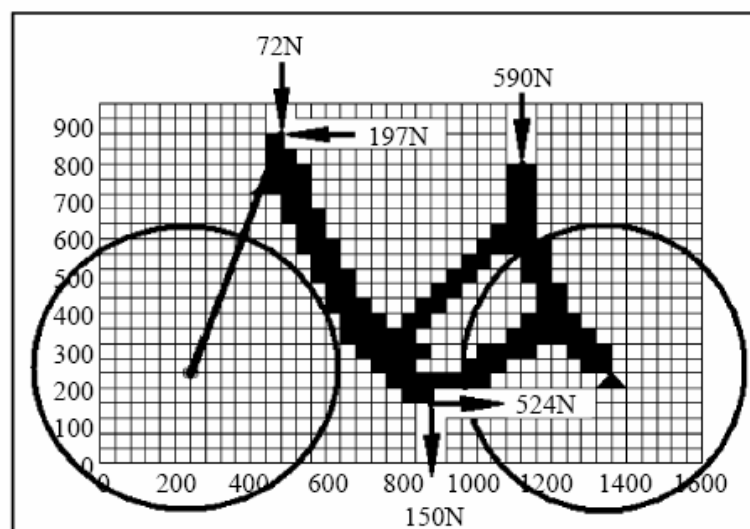


Figura 5.63 – Domínio inicial do quadro de bicicleta. Fonte: Lin e Hsu (2008)

Para este problema, os dados pertinentes do modelo são os seguintes:

- Malha utilizada (*mesh*): a malha gerada no Code-Aster® possui 14392 elementos do tipo TETRA4 (Fig. 5.64);
- Módulo de Elasticidade (E): 210 GPa;
- Coeficiente de Poisson (ν): 0,3;
- Espessura (t): 100 mm;
- Volume máximo desejado (VOLMAX) = 20%;
- Remoção máxima de volume mapeado por iteração (RVOL) = 95% (5% dos elementos são devolvidos para a estrutura);
- Fator de raio de rejeição (FRR) = 1,75%;
- Fator evolucionário de rejeição (FER) = 0,75%.

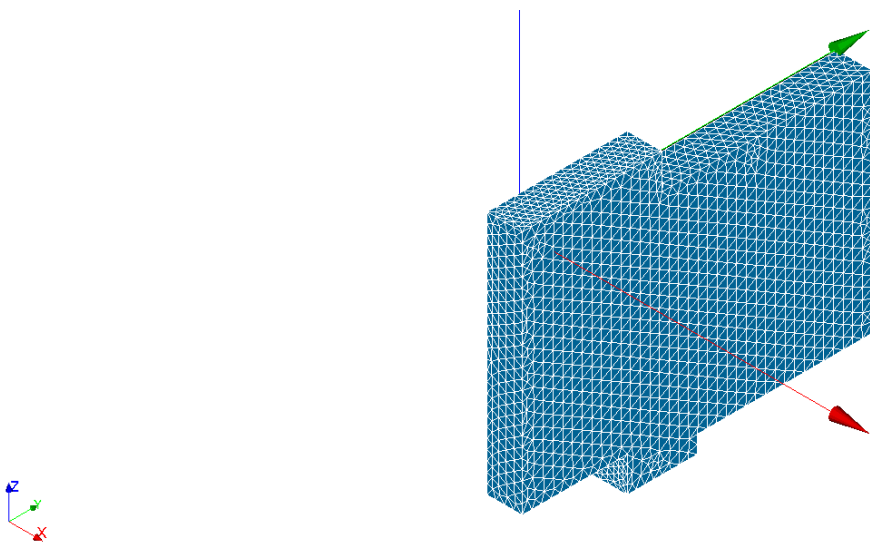


Figura 5.64 – Malha de elementos finitos gerada pelo Salome-Meca®

O processo evolutivo de otimização do quadro de bicicleta é mostrado nas Figs. 5.65 a 5.67.

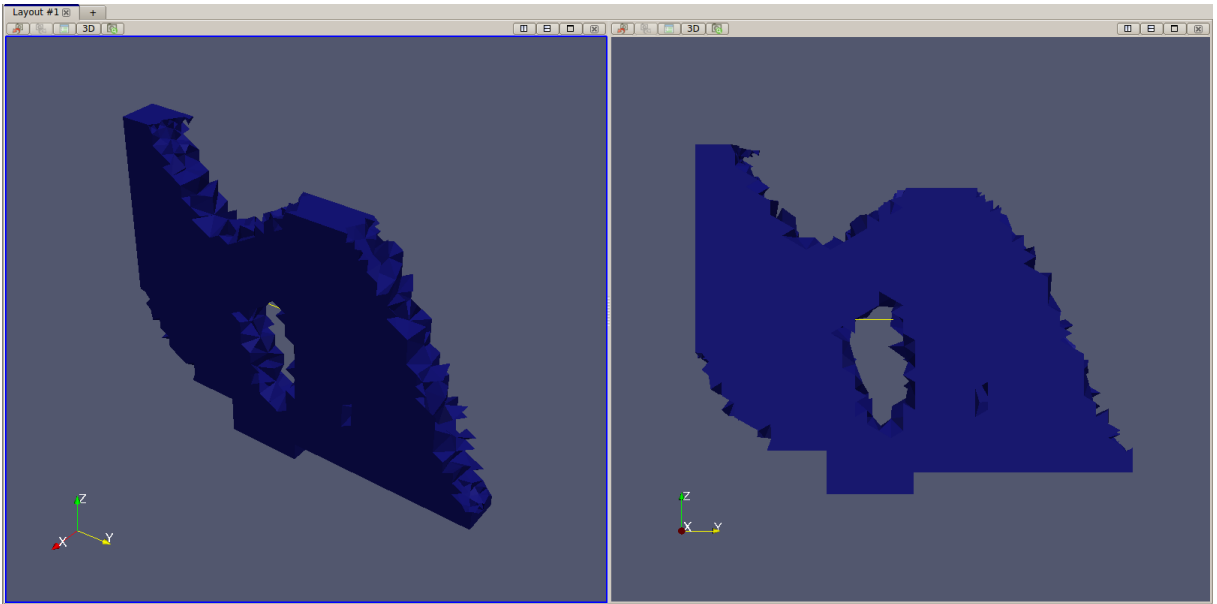


Figura 5.65 – Topologia parcial resultante do passo de otimização 6

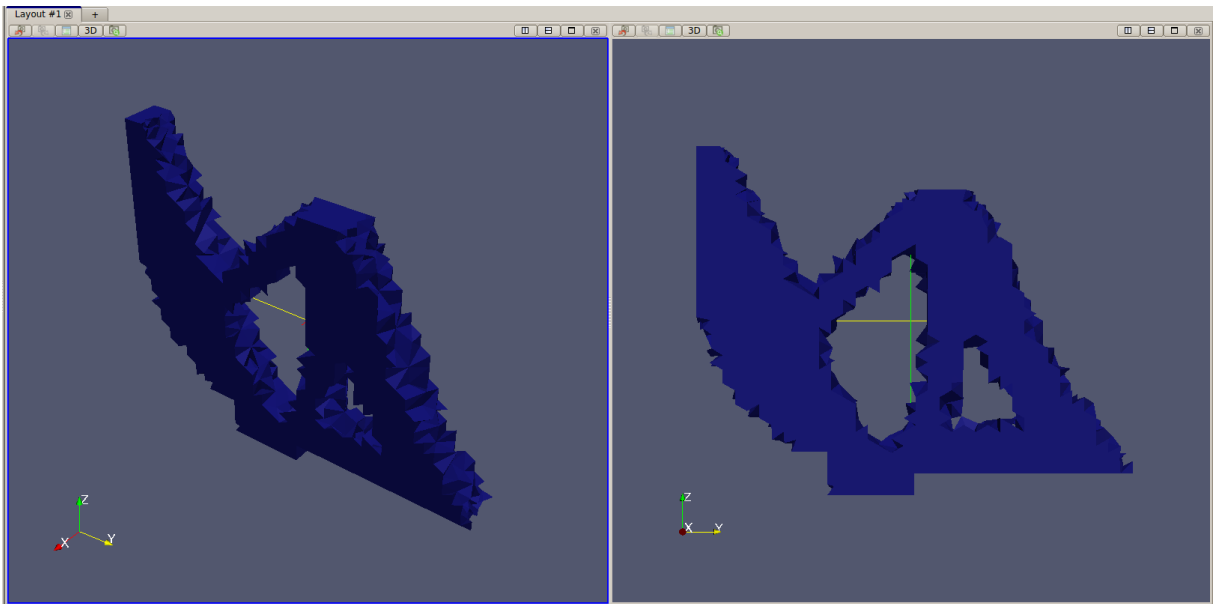


Figura 5.66 – Topologia parcial resultante do passo de otimização 11

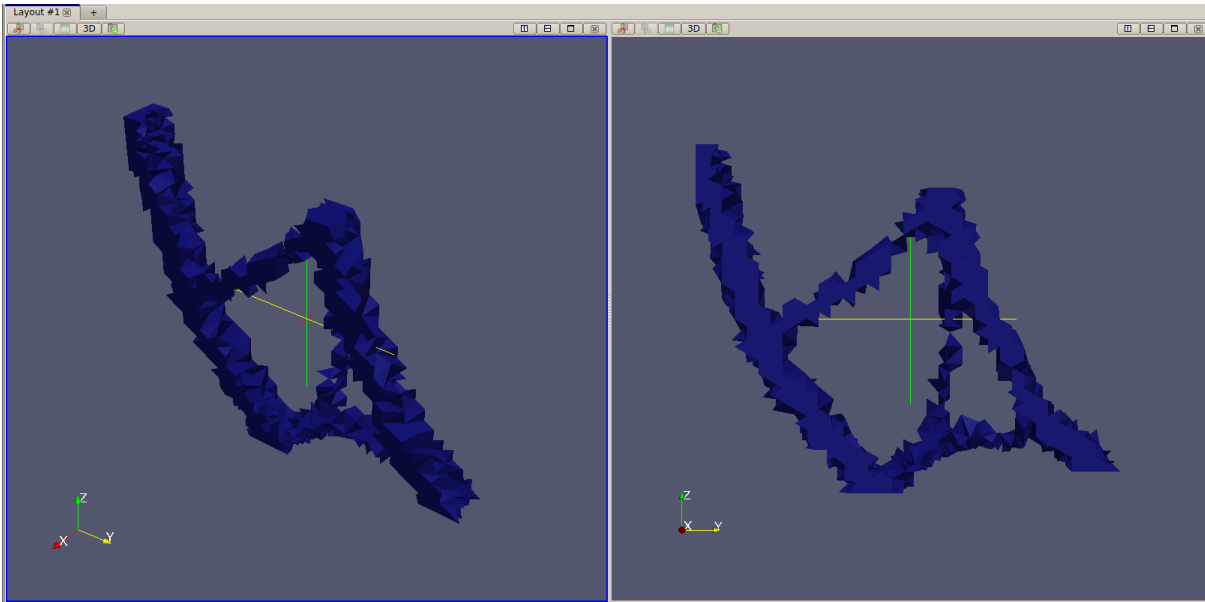


Figura 5.67 – Topologia parcial resultante do passo de otimização 36 (última iteração)

A topologia final ótima obtida para o problema é mostrada na Fig. 5.68.

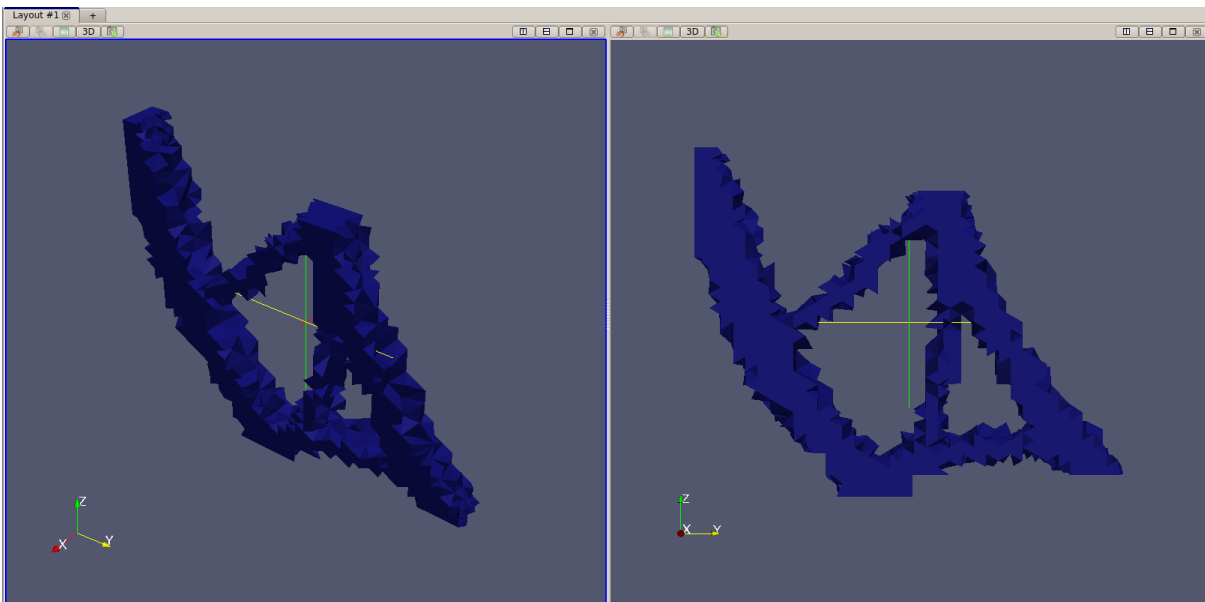


Figura 5.68 – Topologia final ótima resultante do passo de otimização 21

A Tabela 5.8 mostra os valores dos parâmetros e do vetor WSM. A linha em destaque marca a iteração e o menor valor do vetor WSM, caracterizando a resposta da melhor topologia obtida pela análise de otimização multiobjetivo. Em seguida, tem-se na Fig. 5.69 um gráfico que descreve o comportamento de cada um destes parâmetros ao longo do processo iterativo.

Tabela 5.8 – Valores dos parâmetros e do Vetor WSM

Iteração	Volume	Deslocamento (cm, %)		Tensão (MPa, %)		IP	WSM
0	100,00	0,52	24,27	647,11	94,72	100,00	79,59
1	95,65	0,52	24,31	647,84	94,83	95,82	76,86
2	90,64	0,53	24,49	649,99	95,15	95,26	74,25
3	86,29	0,54	25,06	657,14	96,19	96,00	72,31
4	81,57	0,55	25,57	664,02	97,20	96,17	70,10
5	76,09	0,56	26,29	678,10	99,26	94,69	67,46
6	70,67	0,57	26,71	683,15	100,00	93,41	64,65
7	65,87	0,59	27,40	681,40	99,74	89,83	61,88
8	60,75	0,62	28,66	676,29	98,99	74,18	57,66
9	57,09	0,65	30,35	669,40	97,99	73,98	56,06
10	54,42	0,67	31,28	660,11	96,63	73,72	54,74
11	51,56	0,69	31,99	656,59	96,11	74,08	53,43
12	48,94	0,71	33,14	653,06	95,60	73,29	52,21
13	46,26	0,74	34,59	652,44	95,50	71,86	51,02
14	43,66	0,79	36,59	653,99	95,73	70,74	50,08
15	41,36	0,83	38,58	655,69	95,98	69,28	49,26
16	39,25	0,86	40,12	654,96	95,87	67,25	48,31
17	37,66	0,90	42,11	655,85	96,00	65,13	47,77
18	35,51	0,94	43,69	653,07	95,60	65,08	47,01
19	34,28	1,01	46,99	651,37	95,35	65,06	47,22
20	32,92	1,02	47,29	654,79	95,85	63,42	46,44
21	31,28	1,07	49,75	653,80	95,70	63,58	46,23
22	29,98	1,14	52,92	660,07	96,62	62,98	46,40
23	28,86	1,24	57,87	660,20	96,64	62,69	47,09
24	27,59	1,33	62,04	663,99	97,20	62,44	47,54
25	26,65	1,47	68,31	661,66	96,85	61,90	48,60
26	25,93	1,61	75,10	642,50	94,05	63,18	49,88
27	24,95	1,79	83,27	636,15	93,12	64,83	51,61
28	24,63	1,86	86,57	634,21	92,84	64,07	52,20
29	24,19	1,88	87,38	634,22	92,84	63,42	52,10
30	23,47	1,89	87,83	629,96	92,21	63,53	51,80
31	22,90	1,91	88,93	622,11	91,06	64,39	51,78
32	22,14	1,93	89,92	613,52	89,81	65,15	51,60
33	21,70	2,01	93,51	615,38	90,08	64,46	52,26
34	20,94	2,05	95,27	638,54	93,47	60,58	52,21
35	20,12	2,08	96,79	648,58	94,94	59,18	52,17
36	19,73	2,15	100,00	682,02	99,83	57,27	53,07

Neste exemplo o volume decresce suavemente ao longo de todo o processo iterativo. Os valores dos deslocamentos aumentam gradativamente à medida que os elementos são penalizados, havendo uma ligeira suavização a partir da iteração 28. A variação entre os valores da primeira e última iterações, é da ordem de quatro vezes. A tensão sofre pouca variação ao longo do processo iterativo, definindo que a topologia ótima obtida possui certa eficiência. O maior valor de tensão de toda a análise aparece na iteração 6, onde há o início de

uma cavidade. O índice de performance sofre uma queda acentuada devido à formação desta cavidade (iteração 7), mas decresce de forma suave a partir deste ponto até o fim da análise.

Embora a topologia ótima obtida seja ligeiramente diferente do original devido a esta cavidade formada, o membro interno que aparece a partir de então garante uma certa rigidez para o modelo estrutural, assemelhando-se a quadros convencionais de bicicleta existentes.

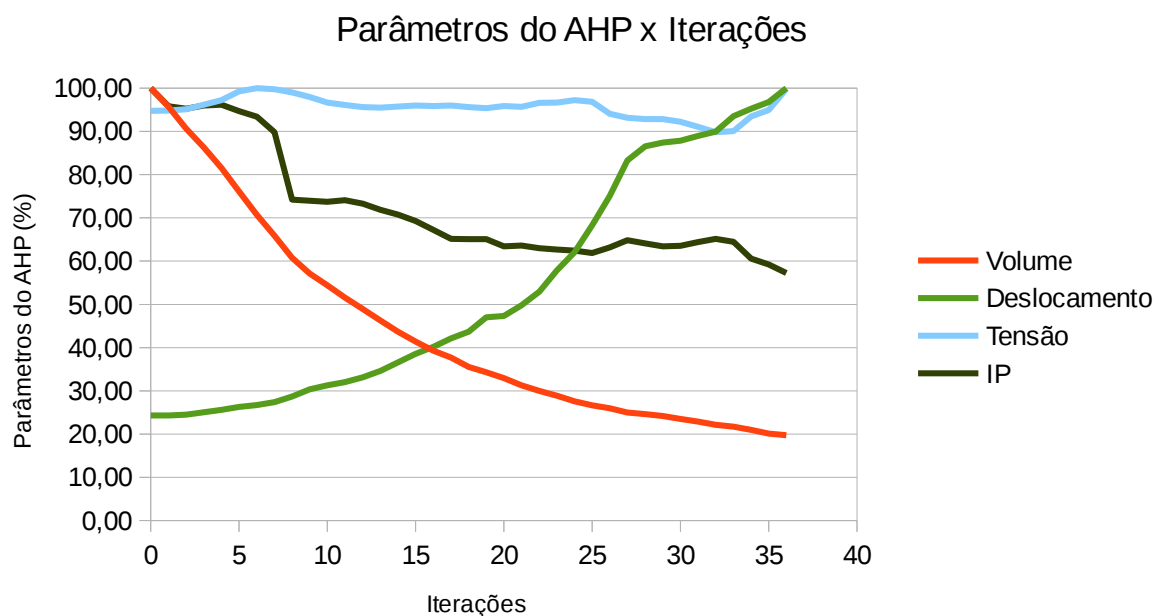


Figura 5.69 – Parâmetros do Vetor WSM (Volume, Deslocamentos, Tensão, Índice de Performance) x Iterações

Para demonstrar a evolução da otimização, a Fig. 5.70 resume o problema, mostrando os valores do vetor WSM ao longo do processo iterativo. O ponto vermelho mostra a iteração com o valor mínimo, que representa a melhor solução para o problema.

Neste exemplo, o vetor WSM possui valor mínimo na iteração 21, configurando esta como a solução para o conjunto de possíveis soluções para a análise. Nessa iteração, o volume é de aproximados 30% em relação ao volume inicial. Nesta iteração, tanto a tensão quanto o deslocamento é praticamente igual ao valor médio destes atributos, respectivamente, considerando toda a análise e o índice de performance se mantém nos 63%.

Otimização Topológica Multiobjetivo

(Modelo Linear)

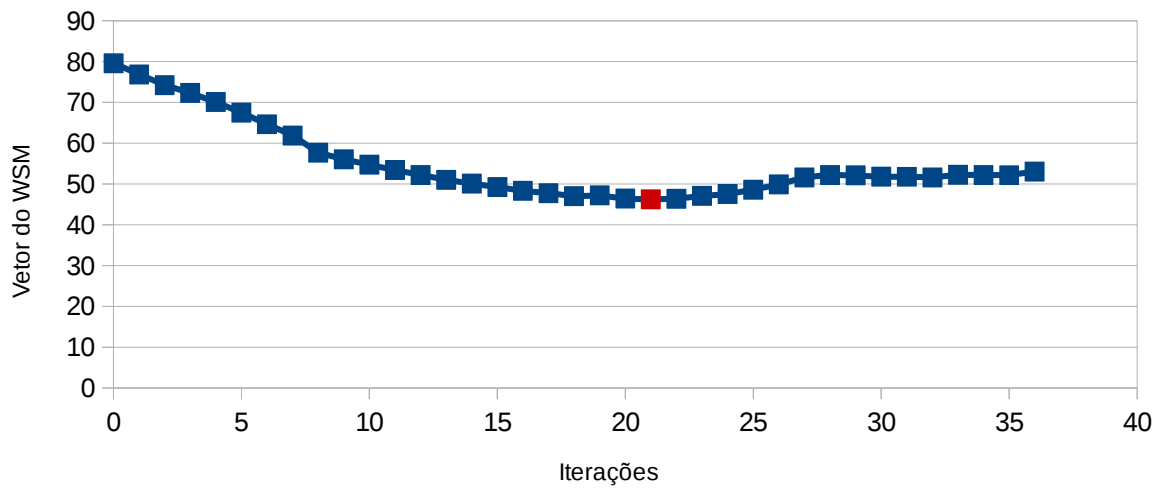


Figura 5.70 – Vetor do WSM x Iterações

5.9 Roda de Trem

Este exemplo procura otimizar a melhor topologia para uma roda maciça, considerando algumas condições normalmente impostas, de acordo com a Fig. 5.71. As dimensões indicadas (dos raios) são dadas em milímetros.

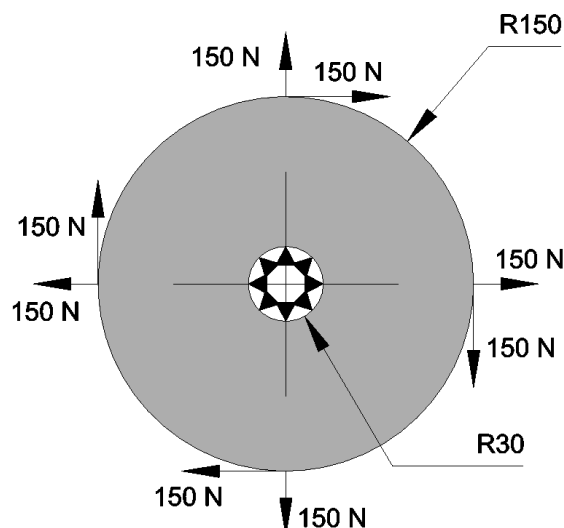


Figura 5.71 – Domínio inicial da roda de trem

Para este problema foi aproveitada a condição de simetria, ou seja, apenas $\frac{1}{4}$ do modelo foi utilizado para a realização da análise. Além disso, os dados pertinentes do modelo são os seguintes:

- Malha utilizada (*mesh*): a malha gerada no Code-Aster® possui 15181 elementos do tipo TETRA4 (Fig. 5.72);
- Módulo de Elasticidade (E): 210 GPa;
- Coeficiente de Poisson (ν): 0,3;
- Espessura (t): 30 mm;
- Volume máximo desejado (VOLMAX) = 20%;
- Remoção máxima de volume mapeado por iteração (RVOL) = 95% (5% dos elementos são devolvidos para a estrutura);
- Fator de raio de rejeição (FRR) = 1,95%;
- Fator evolucionário de rejeição (FER) = 0,95%.

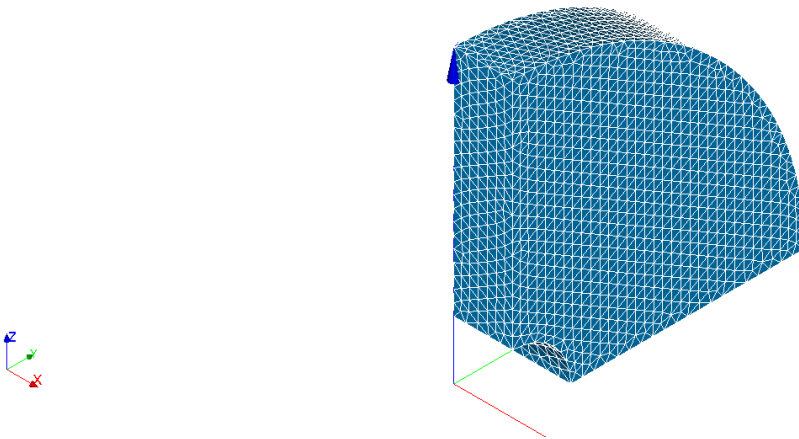


Figura 5.72 – Malha de elementos finitos gerada pelo Salome-Meca®

O processo evolutivo de otimização da roda de trem é mostrado a seguir, nas Figs. 5.73 a 5.75.

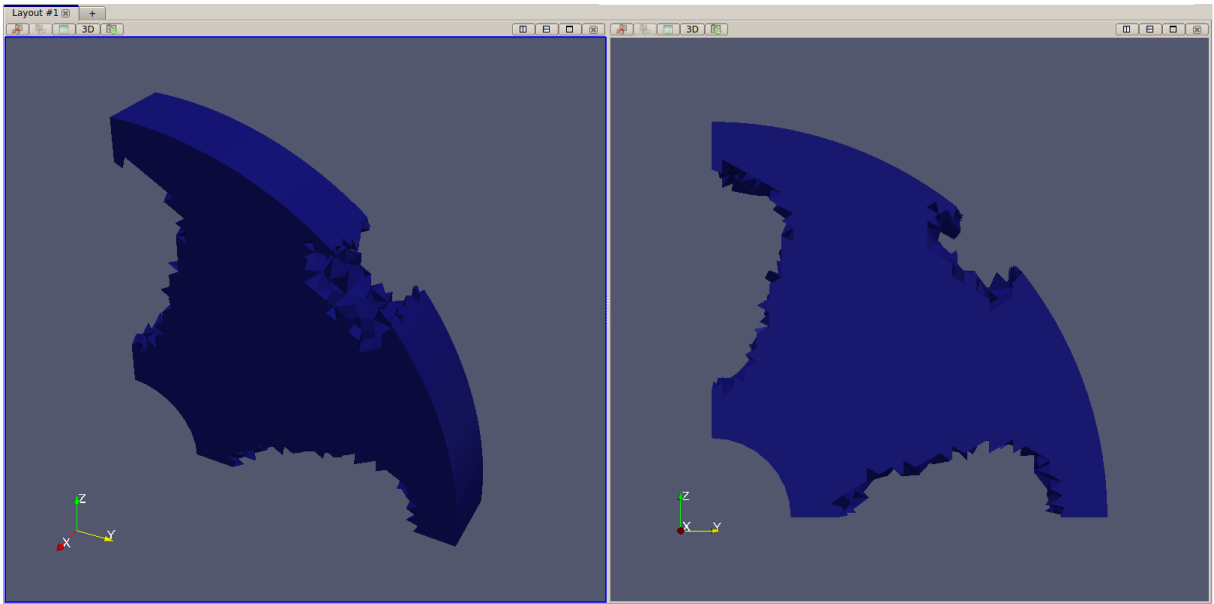


Figura 5.73 – Topologia parcial resultante do passo de otimização 6

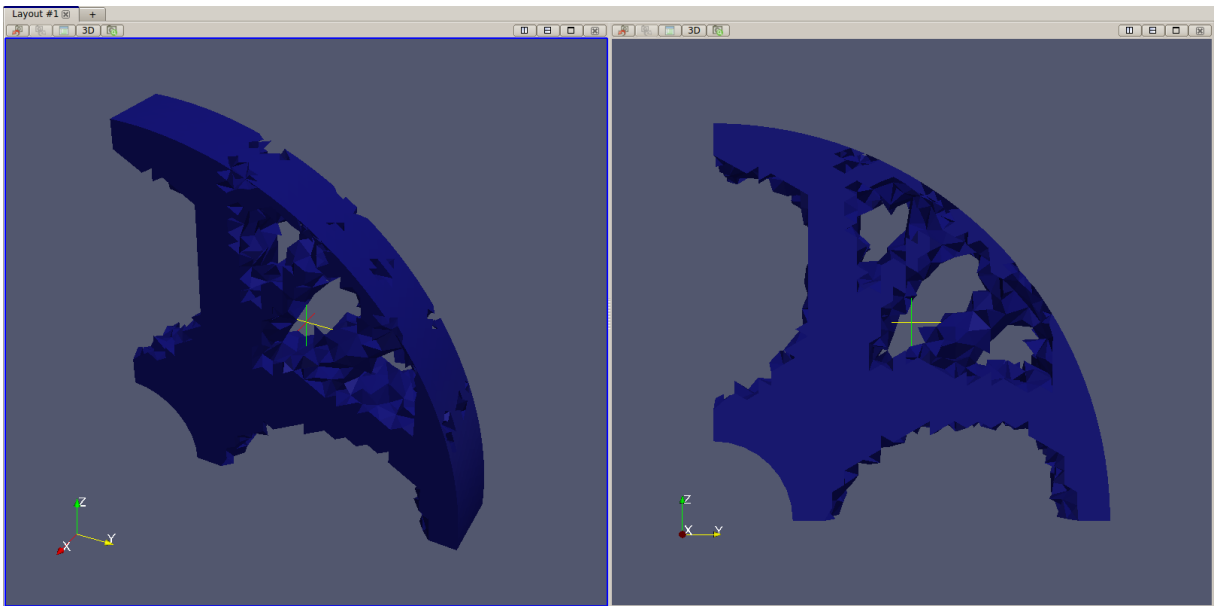


Figura 5.74 – Topologia parcial resultante do passo de otimização 13

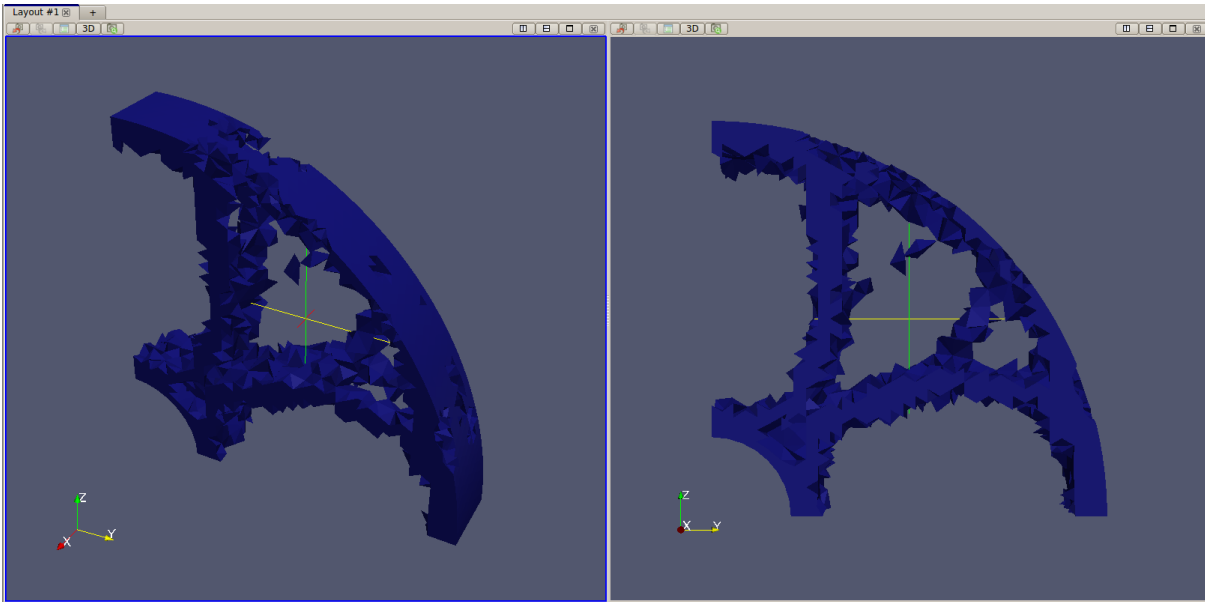


Figura 5.75 – Topologia parcial resultante do passo de otimização 27 (última iteração)

A topologia final ótima obtida para o problema é mostrada na Fig. 5.76.

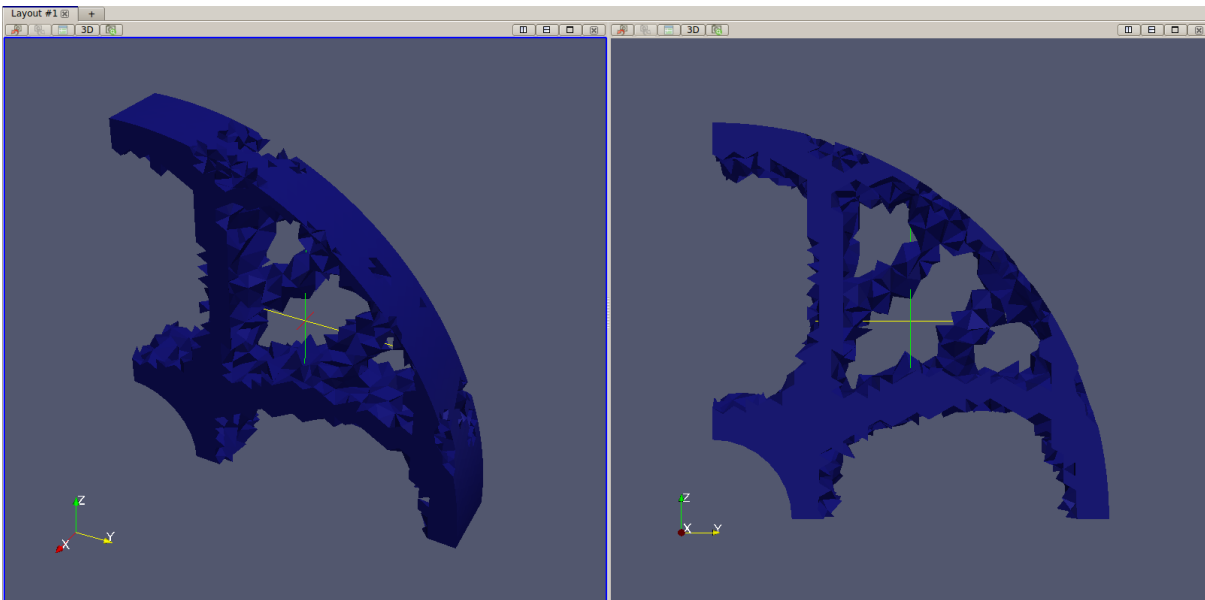


Figura 5.76 – Topologia final ótima resultante do passo de otimização 21

Como foi utilizado a condição de simetria para este exemplo, a topologia ótima final será a combinação de quatro partes da topologia obtida pelo processo de otimização, mostrada na Fig. 5.77.

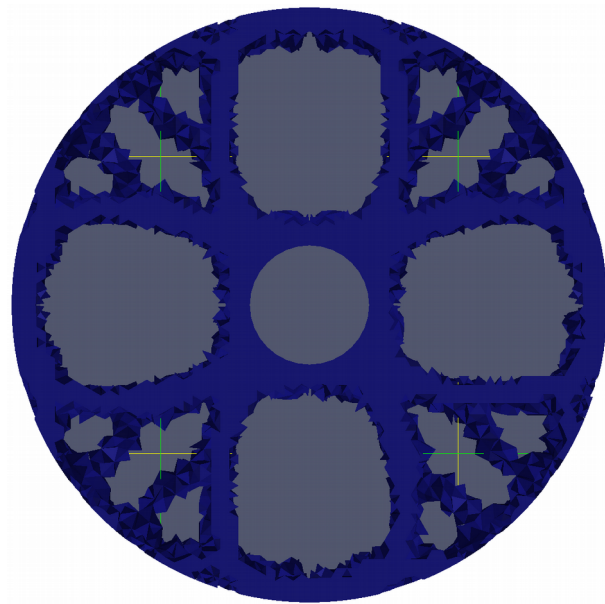


Figura 5.77 – Topologia final ótima completa da roda de trem

A Tabela 5.9 mostra os valores dos parâmetros e do vetor WSM. A linha em destaque nesta tabela marca a iteração e o menor valor do vetor WSM, caracterizando a resposta da melhor topologia obtida pela análise de otimização multiobjetivo. Em seguida, tem-se na Fig. 5.78 um gráfico que descreve o comportamento de cada um desses parâmetros ao longo do processo iterativo.

Tabela 5.9 – Valores dos parâmetros e do Vetor WSM

Iteração	Volume	Deslocamento (cm, %)		Tensão (MPa, %)		IP	WSM
0	100,00	0,28	11,82	205,47	82,33	100,00	75,10
1	97,58	0,28	11,82	205,48	82,33	99,18	73,74
2	92,78	0,28	11,87	205,87	82,48	94,05	70,66
3	87,60	0,28	12,03	207,71	83,22	93,71	68,03
4	83,26	0,31	13,20	209,60	83,98	89,91	65,70
5	75,92	0,29	12,34	208,89	83,70	89,55	61,57
6	70,52	2,36	100,00	211,34	84,67	88,49	81,74
7	66,72	1,65	70,08	213,88	85,69	87,15	71,84
8	62,70	0,83	35,27	210,86	84,48	87,29	60,49
9	58,03	0,62	26,43	206,16	82,60	87,76	55,60
10	55,18	0,77	32,70	203,87	81,68	88,05	55,70
11	50,96	0,74	31,23	200,81	80,46	88,68	53,05
12	48,62	0,82	34,80	198,76	79,64	87,55	52,56
13	46,39	0,84	35,38	196,68	78,80	87,10	51,41
14	43,88	0,84	35,72	196,32	78,66	86,25	50,08
15	41,90	0,90	37,92	197,02	78,94	84,83	49,48
16	39,62	0,91	38,48	194,23	77,82	85,42	48,39
17	38,03	0,96	40,56	193,90	77,69	84,64	48,00
18	36,51	0,96	40,75	195,01	78,13	82,87	47,10
19	34,84	0,99	41,87	195,34	78,26	83,02	46,55
20	33,26	1,02	43,33	195,22	78,22	82,74	46,07
21	31,90	1,05	44,42	199,51	79,93	80,68	45,58
22	30,65	1,26	53,49	204,01	81,74	79,92	47,39
23	29,12	1,31	55,38	208,53	83,55	77,65	47,01
24	27,22	1,34	56,67	212,68	85,21	77,76	46,53
25	25,49	1,70	71,97	213,62	85,59	76,41	49,53
26	23,54	1,81	76,69	223,35	89,49	71,18	49,52
27	19,47	2,16	91,33	249,59	100,00	68,52	51,98

Assim como no exemplo anterior, o volume decresce suavemente ao longo de todo o processo iterativo. Quanto aos deslocamentos, na iteração 6 ocorre um pico anormal, cujo valor é o maior em toda a análise. De acordo com a Fig. 5.73, o processo de otimização tendia para uma direção ao remover elementos da superfície da roda. Porém, estes elementos são devolvidos nas iterações seguintes, modificando o rumo de todo o processo e reduzindo novamente os valores de deslocamento no instante em que os elementos da superfície são devolvidos até a iteração 9. Na Fig. 5.74, que corresponde à iteração 13, estes elementos já foram todos devolvidos. Após esse momento, os valores de deslocamento aumentam gradualmente até a iteração 21, onde os membros internos começam a se desfazer (Fig. 5.75). Com relação a tensão, a análise não apresenta grandes variações. Os valores permanecem praticamente constantes ao longo de todo o processo iterativo. O índice de performance decresce de forma suave, sem grandes variações.

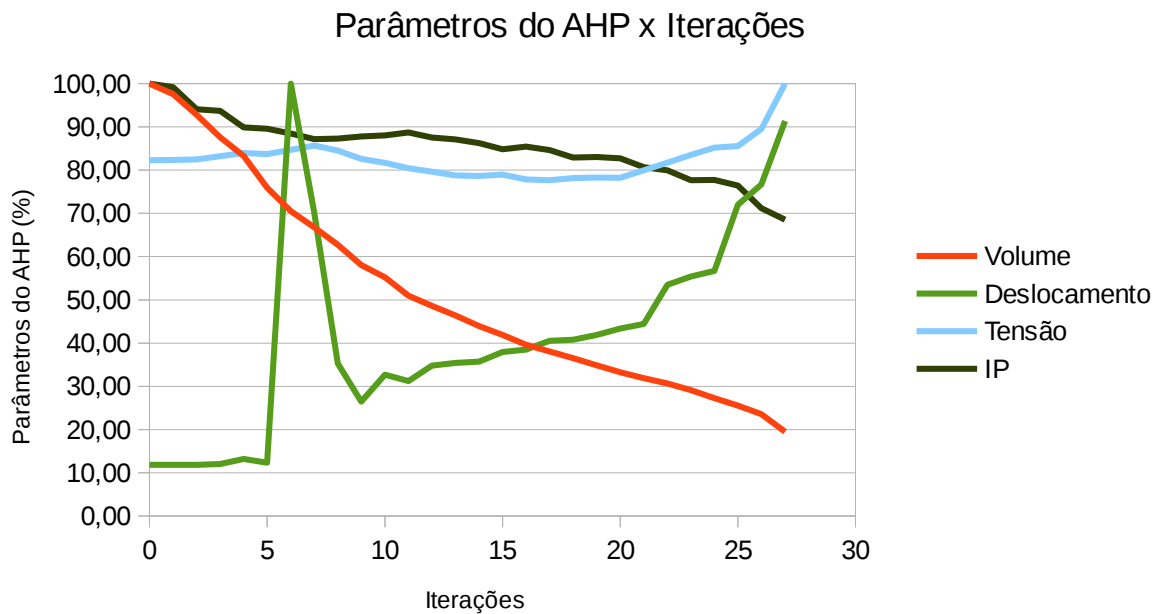


Figura 5.78 – Parâmetros do Vetor WSM (Volume, Deslocamentos, Tensão, Índice de Performance) x Iterações

Para demonstrar a evolução da otimização, a Fig. 5.79 resume o problema, mostrando os valores do vetor WSM ao longo do processo iterativo. O ponto vermelho mostra a iteração com o valor mínimo, que representa a melhor solução para o problema.

Neste exemplo, o vetor WSM possui valor mínimo na iteração 21, configurando esta como a solução para o conjunto de possíveis soluções para a análise. Nessa iteração, o volume é de aproximados 30% em relação ao volume inicial. Nesta iteração, o deslocamento é cerca de 2,25 vezes menor que o deslocamento máximo de toda a análise. A tensão é cerca de 1,25 vezes menor que a tensão máxima e o índice de performance se mantém nos 80%.

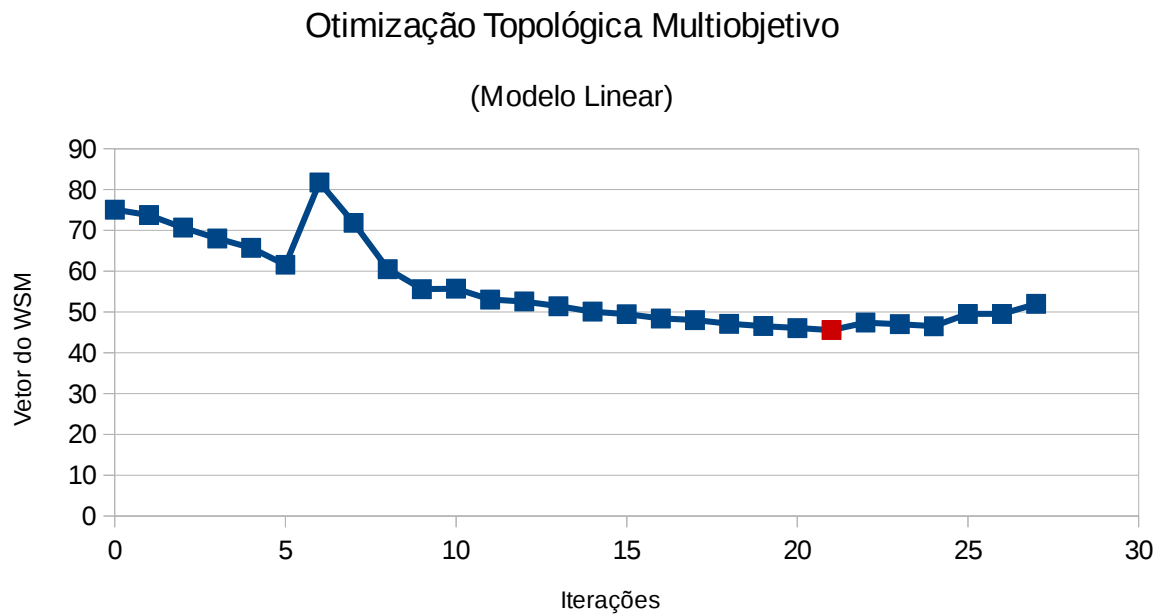


Figura 5.79 – Vetor do WSM x Iterações

5.10 Discussão dos Resultados

Inicialmente, é importante destacar a relação entre as condições de contorno e de carregamento inicial ao qual a estrutura é submetida e a topologia ótima final. Nos exemplos 5.2 e 5.3, por exemplo, o modelo estrutural possui dimensões proporcionais para a placa considerada em cada análise (apesar dos valores reais dessas dimensões serem diferentes), mas como possuem diferentes condições de contorno e carregamento entre os exemplos, as topologias ótimas finais são totalmente distintas entre si. A distribuição de tensões é totalmente dependente do ponto de aplicação das cargas impostas ao problema e das condições de contorno definidas previamente.

De forma geral, o volume dos modelos até então apresentados apresentam uma suavização ao fim do processo iterativo. Como 20% do volume inicial abrange uma perda considerável de material, poucas penalizações são feitas nas últimas iterações comparadas às primeiras. Isso significa que os elementos remanescentes são, de fato, importantes para a estabilidade e desempenho da estrutura e a penalização de elementos, nesta etapa do processo iterativo,

garante um certo custo para a otimização, como a perda de membros fundamentais para o equilíbrio da estrutura (estrutura hipostática), e/ou brusco aumento de tensão nos elementos restantes, e/ou aumento expressivo dos deslocamentos ao se comparar com o modelo inicial. Enquanto há perda de volume por iteração, uma vez que o objetivo é minimizar volume, os deslocamentos aumentam à medida que elementos são penalizados. A perda de material por iteração resulta em estruturas cada vez mais esbeltas e que, conseqüentemente, tendem à instabilidade. Por este motivo, uma ponderação entre o volume e o deslocamento se faz necessária, ressaltando a importância da otimização multiobjetivo deste trabalho.

Com relação ao comportamento da tensão, e assim como o que pode ser visto nos exemplos em Fernandes (2013), quando um membro interno é composto por elementos finitos ineficientes (ou que serão removidos ao longo do processo de otimização) e esse membro é removido completamente, ocorre um pico de tensão devido a um rearranjo da distribuição de tensões ao longo da estrutura. A estrutura se adapta nos passos seguintes de otimização e passa a depender dos membros remanescentes. Isso pode ser visto no exemplo 1, onde não há remoção de membros internos (há pouca variação da tensão, de forma geral), ao contrário dos exemplos 2 e 3, que a tensão tende a aumentar muito a partir dos passos de otimização onde um determinado membro interno é removido. E, assim como os deslocamentos, a tensão máxima também aumenta à medida que os elementos são penalizados. Em modelos como o exemplo 5.5, esse aumento chega a ser tão expressivo que o próprio material não suportaria tamanha tensão, como mostra a Fig. 5.43. Apesar disto, a análise foi considerada até o final justamente para esta comparação em relação à perda de material e aumento de tensão.

Outro ponto importante é a presença de uma ligeira falta de simetria nas respostas obtidas. Isso acontece devido à dependência da malha, já que nenhum filtro foi implementado para minimizar este efeito e não foram utilizadas as condições de simetria do problema, exceto no caso do último exemplo deste capítulo. De acordo com Fernandes (2013), malhas diferentes resultam em topologias finais ótimas distintas, podendo esta diferença ser significativa ou não, ou seja, analisar um mesmo modelo com as mesmas condições iniciais e de contorno a partir de malhas diferentes pode ou apenas melhorar a simetria em malhas mais refinadas ou mudar totalmente a topologia ótima final entre cada análise dessa maneira. O Exemplo 5.5 foi o único modelo cuja malha foi criada a partir de elementos quadriláteros. Devido à sua

simetria inicial, em relação à carga e restrições, era esperado que a topologia ótima final mantivesse essa simetria. Isso, na verdade, não acontece, pois ao passo que o SESO promete uma remoção suave, ele não pondera o retorno dos elementos de forma simétrica, fazendo com que uma assimetria seja formada não pela penalização, mas pela devolução de elementos. Como nenhum filtro foi aplicado para garantir uma devida devolução, a assimetria é perceptível. Nos demais modelos onde elementos triangulares foram utilizados, mesmo com a assimetria natural da malha, os modelos apresentaram uma topologia final ótima bem definida, semelhante às topologias obtidas pela literatura (Exemplos de 5.1 a 5.4).

É interessante também ressaltar que, exceto para o primeiro exemplo no qual a tensão se manteve praticamente constante durante toda a análise, em todos os gráficos relativos aos parâmetros do AHP, a reta paralela ao eixo das ordenadas (porcentagem dos atributos) que corta o eixo das abcissas (iterações) no exato ponto da iteração ótima, possui um ponto pertencente a uma área delimitada pelas quatro curvas em cada análise. Ao se observar cada gráfico, uma única área é formada, englobando todas as quatro curvas simultaneamente. A ocorrência deste efeito não significa que o ótimo esteja necessariamente dentro daquele espaço, mas sim que aquela área corresponde a um equilíbrio entre os parâmetros e esse equilíbrio está diretamente ligado com a obtenção da topologia ótima, de acordo com a ideia da análise de otimização multiobjetivo via AHP e WSM.

Finalmente, os exemplos estáticos lineares apresentados neste capítulo mostraram topologias finais ótimas adequadas para cada problema imposto ao serem comparadas com os resultados encontrados na literatura. Isso, de certa forma, garante o bom funcionamento do algoritmo SESO implementado e sua interação com os programas necessários que foram descritos anteriormente. Já os exemplos 5.5, 5.6 e 5.7 mantêm o foco em estruturas tridimensionais para o qual o código, de fato, foi desenvolvido, e os exemplos 5.8 e 5.9 procuram, respectivamente, abordar um problema clássico da otimização estrutural e a condição de simetria para a obtenção da topologia ótima.

6

RESULTADOS NÃO LINEARES

Este capítulo se destina a tratar especificamente de três resultados não lineares obtidos pela implementação do código de otimização SESO. As estruturas foram modeladas no *software* Salome-Meca® versão 2014.1, descrito no capítulo anterior. Todos os modelos aqui apresentados foram analisados em 3D, sendo que os modelos bidimensionais retirados da literatura foram aproximados com uma espessura consideravelmente menor em relação aos outros parâmetros. Além disso, em cada um dos três exemplos foram obtidas as respostas linear, não linear geométrica e não linear física e geométrica, usando a mesma malha e parâmetros para estas três análises. Assim como nas análises lineares, os valores de deslocamento considerados para todas as análises deste capítulo referem-se sempre ao maior deslocamento ao qual a estrutura for submetida na iteração correspondente. O modelo constitutivo elastoplástico do material para todas as análises que incluem a não linearidade física é representado de acordo com o diagrama tensão x deformação apresentado na Fig. 6.1.

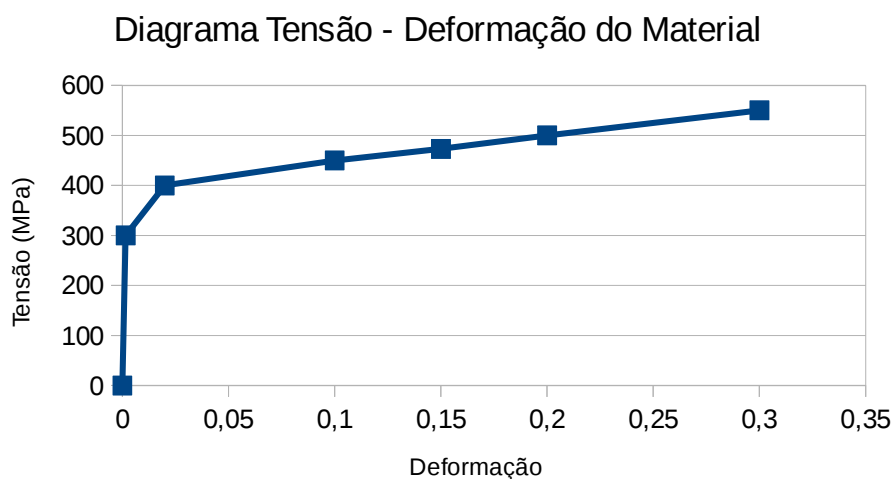


Figura 6.1 – Diagrama tensão x deformação do material nas análises fisicamente não lineares

As diferenças e/ou semelhanças entre as topologias ótimas encontradas são também mostradas. Para as análises não lineares foram usados 50 passos de carga e uma tolerância de 10^{-4} para a convergência dos resultados. As tabelas que contém os valores dos parâmetros se encontram no Apêndice A.

6.1 Chapa Apoiada pelas Extremidades nos Nós Centrais

Este modelo consiste em uma chapa retangular apoiada nas extremidades, nos nós centrais dos lados. Uma força vertical é aplicada para cima, no centro da extremidade superior, e possui as dimensões mostradas na Fig. 6.2.

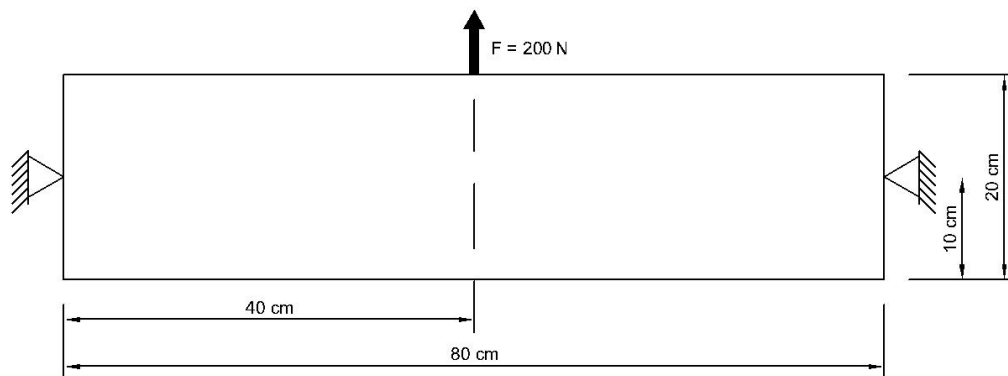


Figura 6.2 – Domínio inicial da chapa apoiada pelas extremidades nos nós centrais

Para este problema, os dados pertinentes do modelo são os seguintes:

- Malha utilizada (*mesh*): a malha gerada no Code-Aster® possui 13584 elementos do tipo TETRA4 (Fig. 6.3);
- Módulo de Elasticidade (E): 210 GPa;
- Tensão de Escoamento (σ_{esc}): 300MPa;
- Módulo de Encruamento Isotrópico (E_{ISO}): 536 MPa;
- Coeficiente de Poisson (ν): 0,3;
- Espessura (t): 4 cm;
- Volume máximo desejado (VOLMAX) = 20%;
- Remoção máxima de volume mapeado por iteração (RVOL) = 95% (5% dos

elementos são devolvidos para a estrutura);

- Fator de raio de rejeição (FRR) = 1,5%;
- Fator evolucionário de rejeição (FER) = 0,85%.

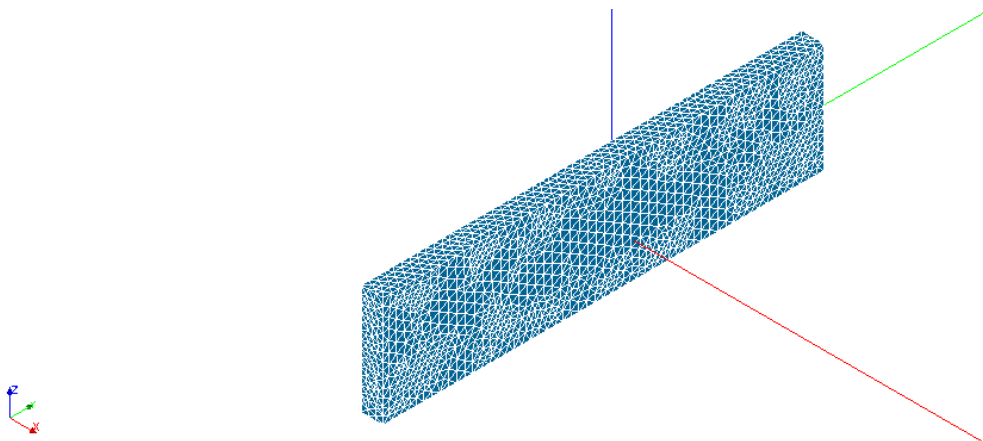


Figura 6.3 – Malha de elementos finitos gerada pelo Salome-Meca®

O processo evolutivo de otimização do problema da chapa apoiada nas extremidades nos nós centrais, para a análise linear, é mostrado nas Figs. 6.4 a 6.6.

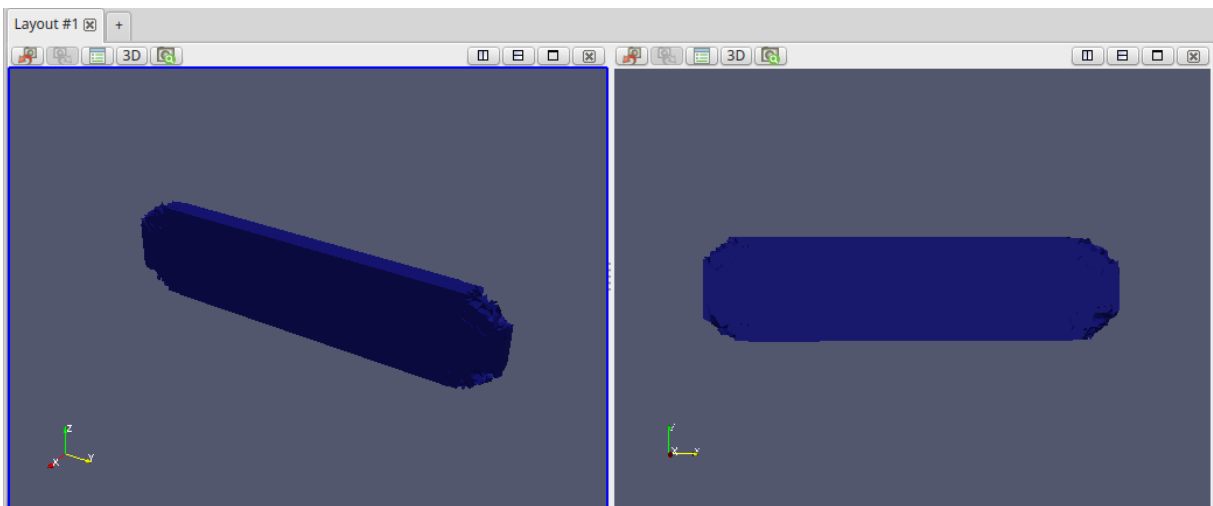


Figura 6.4 – Topologia parcial linear resultante do passo de otimização 5

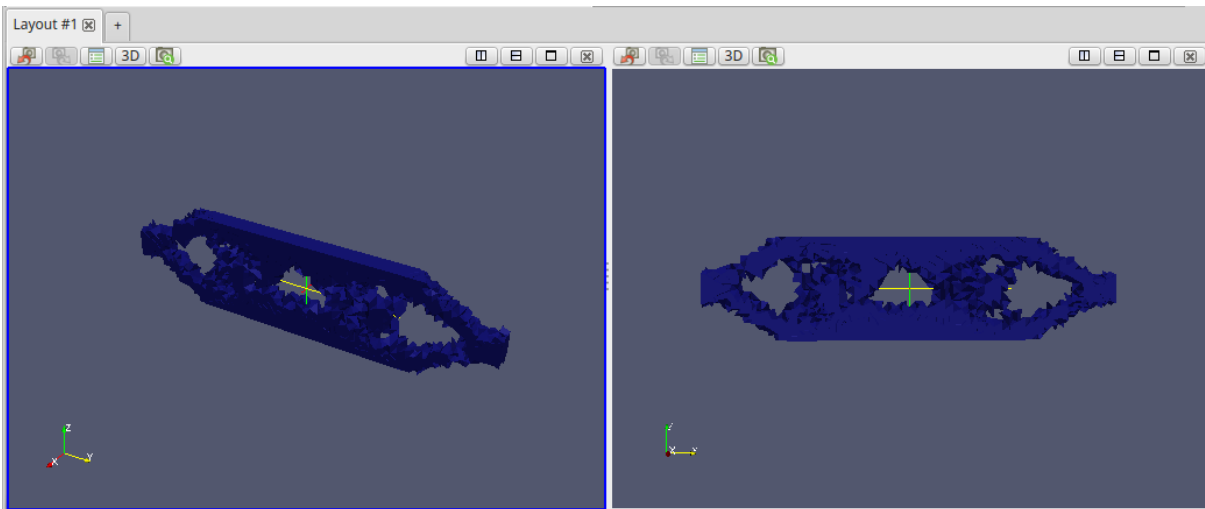


Figura 6.5 – Topologia parcial linear resultante do passo de otimização 14

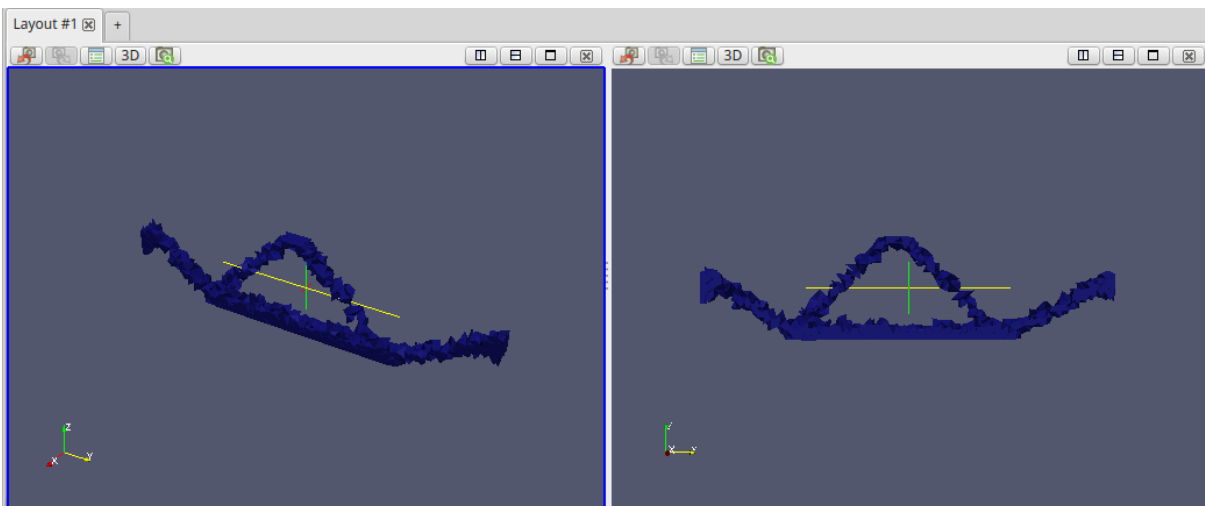


Figura 6.6 – Topologia parcial linear resultante do passo de otimização 27 (última iteração)

A topologia final ótima obtida para o problema considerando a análise linear é mostrada na Fig. 6.7.

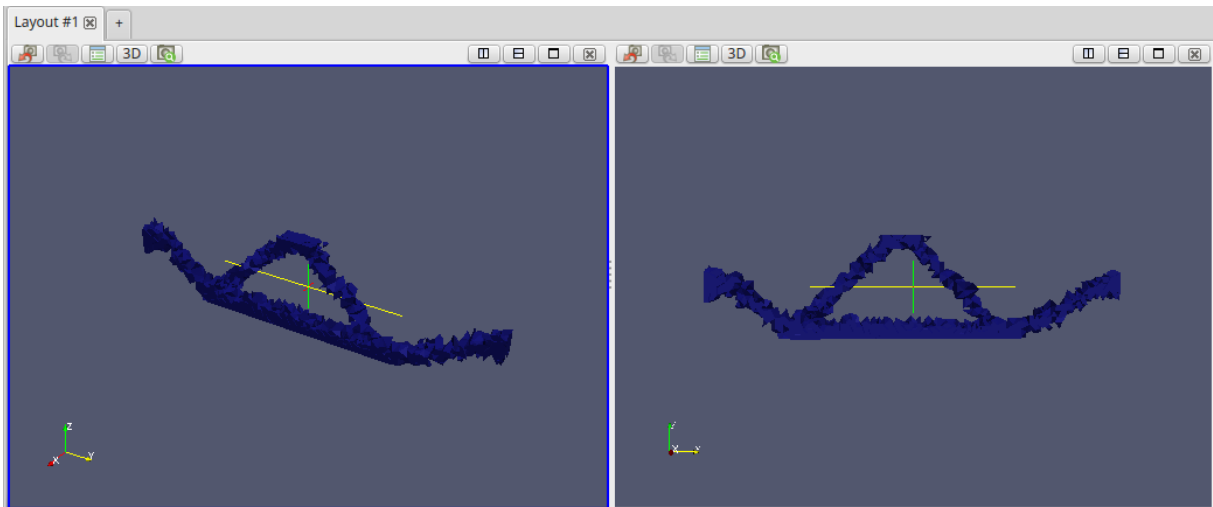


Figura 6.7 – Topologia final ótima linear resultante do passo de otimização 21

O processo evolutivo de otimização do problema da chapa apoiada nas extremidades nos nós centrais, para a análise não linear geométrica, é mostrado nas Figs. 6.8 a 6.10.

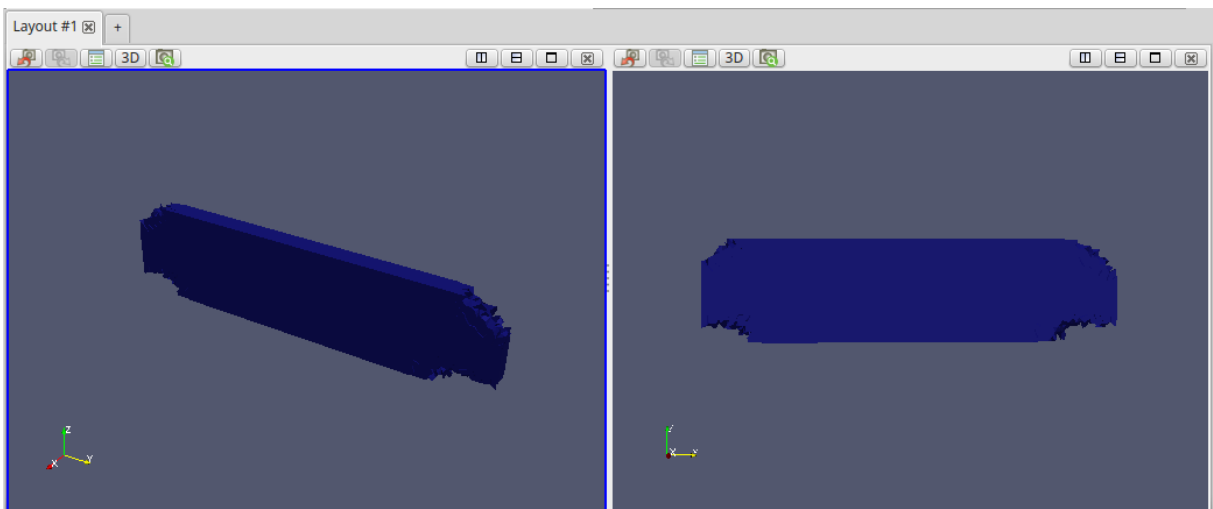


Figura 6.8 – Topologia parcial não linear geométrica resultante do passo de otimização 5

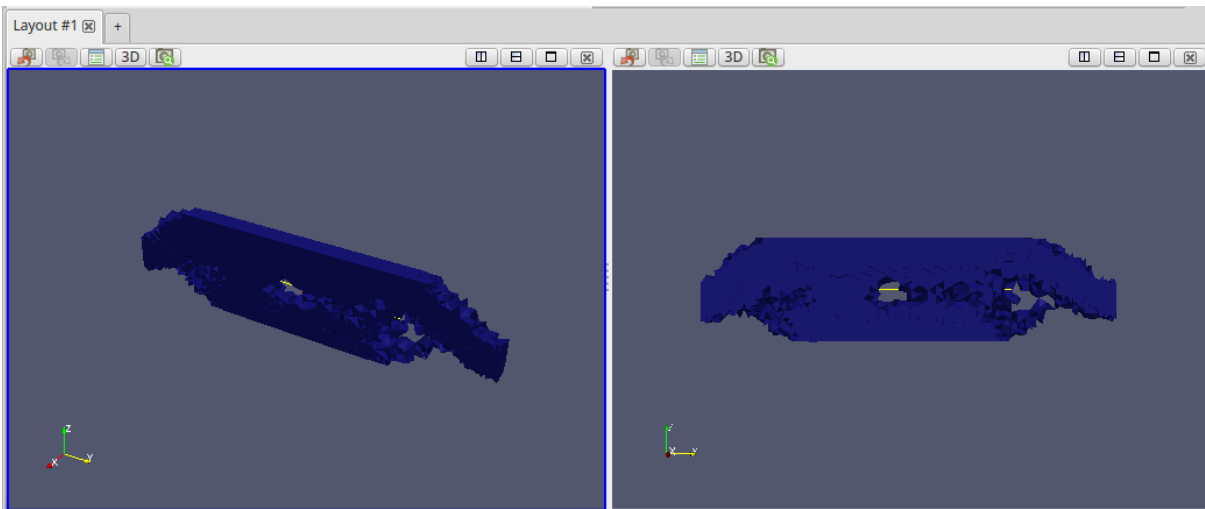


Figura 6.9 – Topologia parcial não linear geométrica resultante do passo de otimização 11

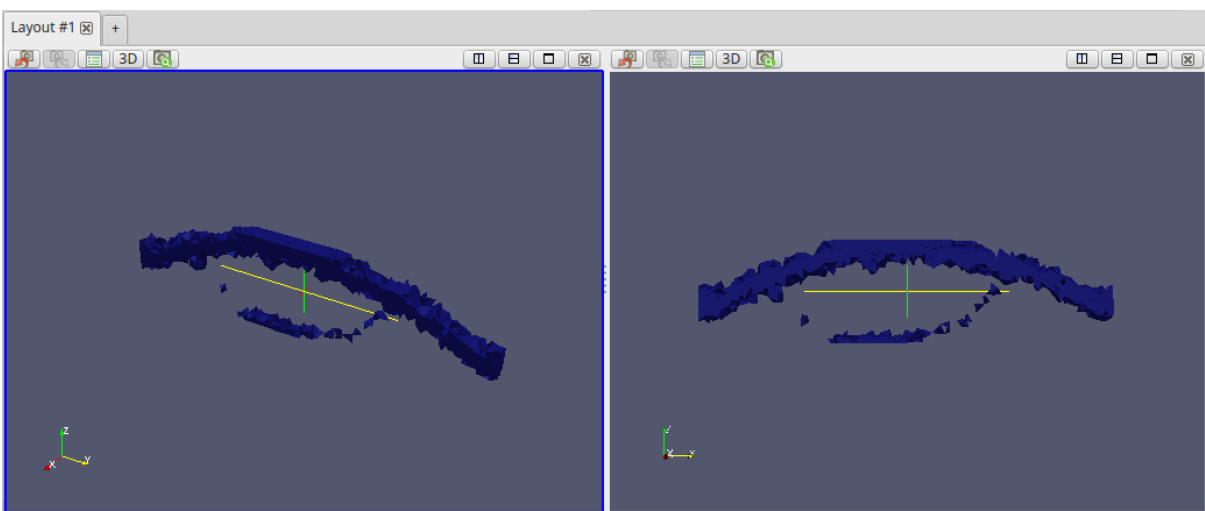


Figura 6.10 – Topologia parcial não linear geométrica resultante do passo de otimização 21
(última iteração)

A topologia final ótima obtida para o problema considerando a análise não linear geométrica é mostrada na Fig. 6.11.

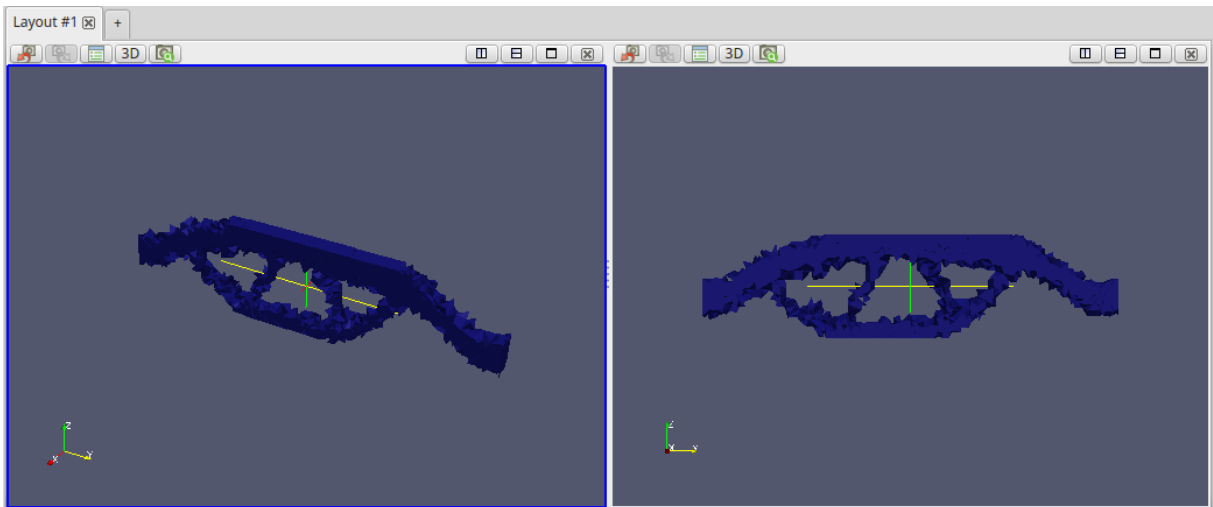


Figura 6.11 – Topologia final ótima não linear geométrica resultante do passo de otimização

18

O processo evolutivo de otimização do problema da chapa apoiada nas extremidades nos nós centrais, para a análise não linear física e geométrica, é mostrado nas Figs. 6.12 a 6.14.

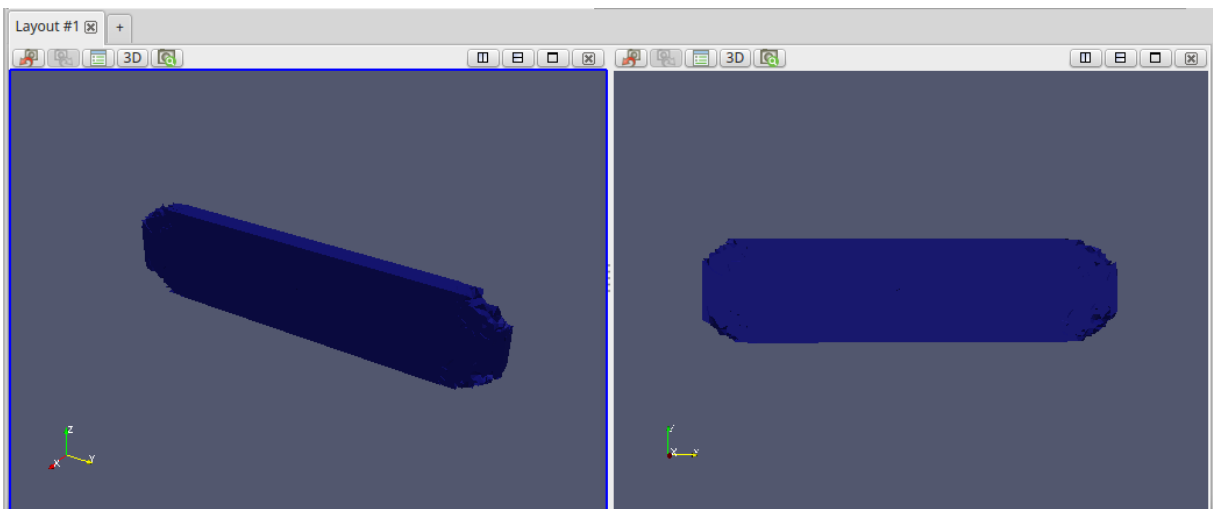


Figura 6.12 – Topologia parcial não linear física e geométrica resultante do passo de otimização 6

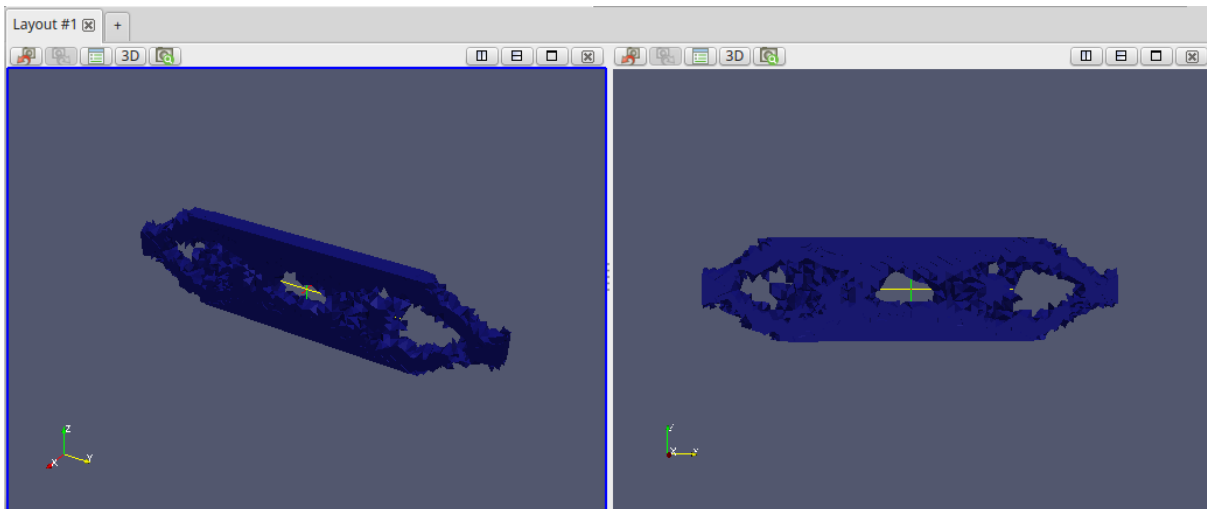


Figura 6.13 – Topologia parcial não linear física e geométrica resultante do passo de otimização 15

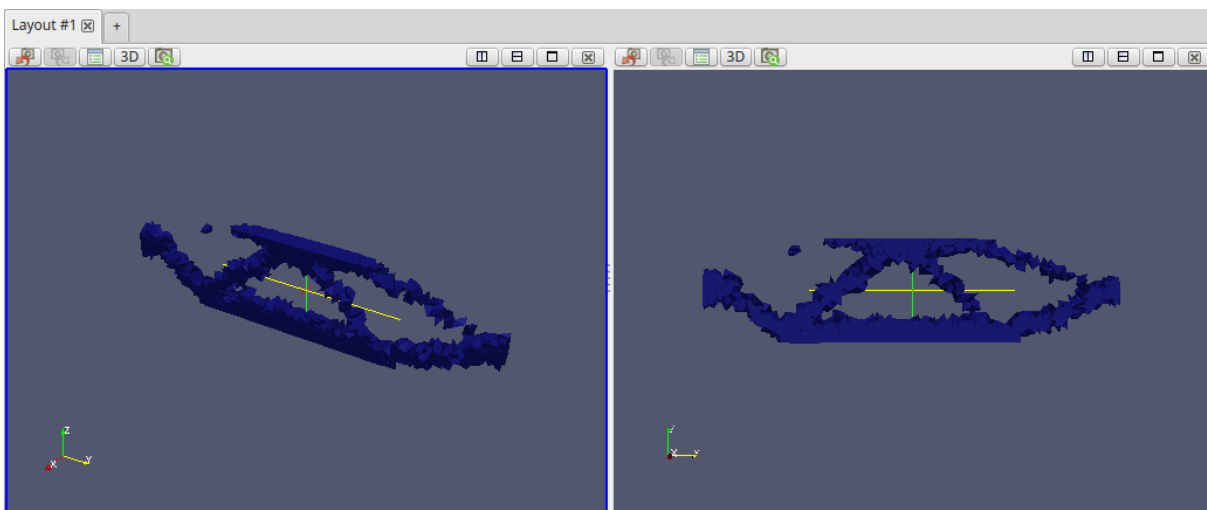


Figura 6.14 – Topologia parcial não linear física e geométrica resultante do passo de otimização 28 (última iteração)

A topologia final ótima obtida para o problema considerando a análise não linear física e geométrica é mostrada na Fig. 6.15.

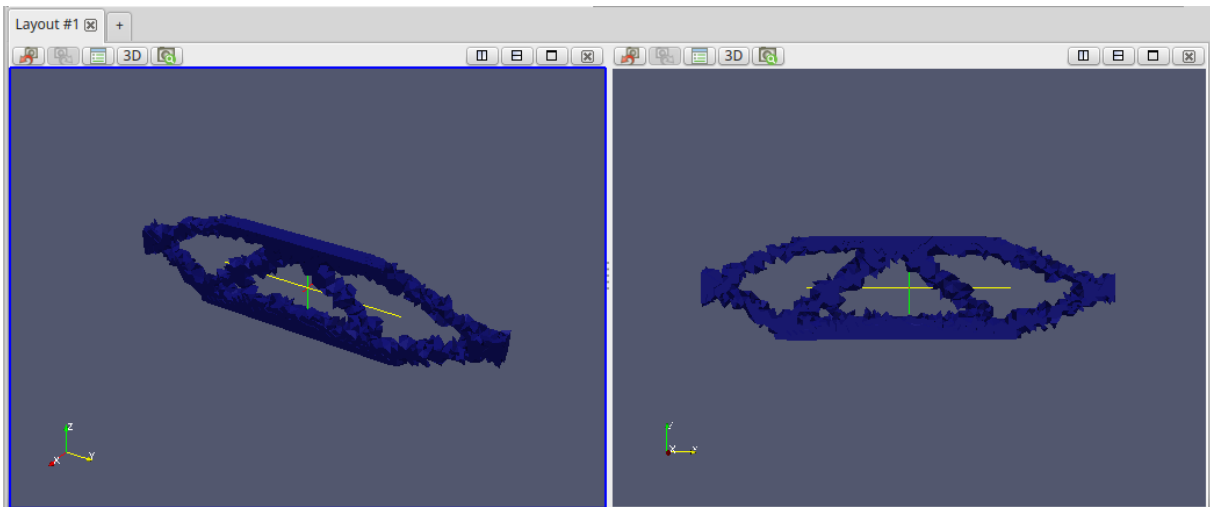


Figura 6.15 – Topologia final ótima não linear física e geométrica resultante do passo de otimização 24

A Fig. 6.16 mostra um comparativo entre as topologias finais ótimas obtidas do modelo proposto pelas encontradas por Gea e Luo (2001).

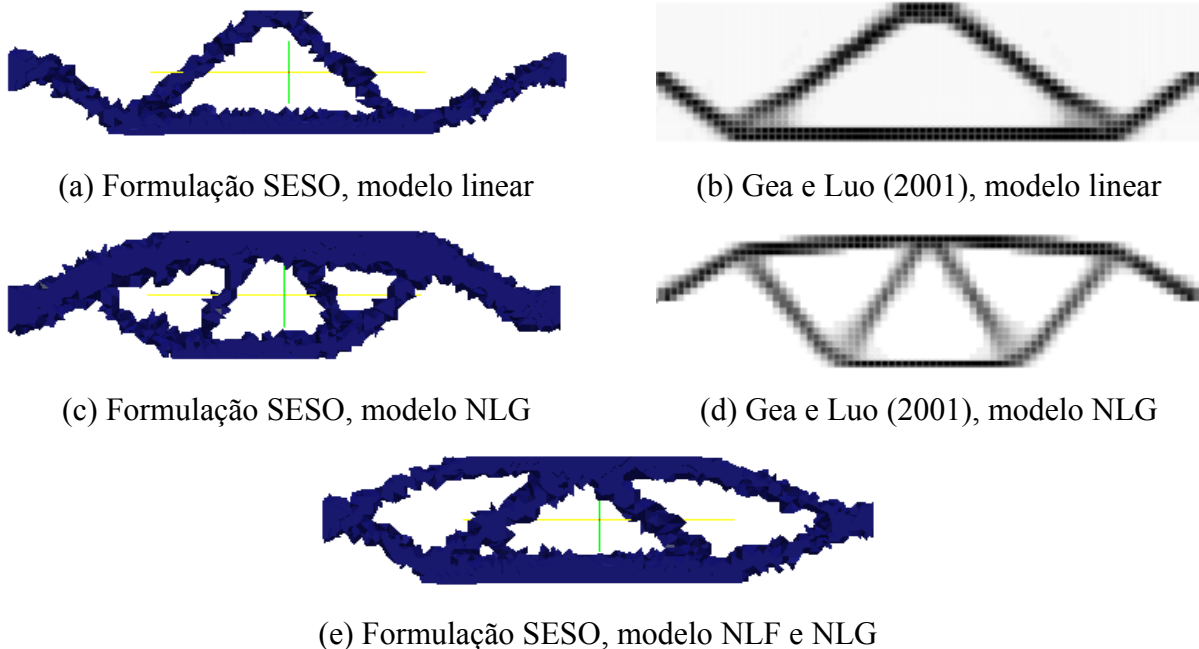


Figura 6.16 – Topologias finais ótimas para o problema da chapa retangular apoiada nas extremidades nos nós centrais

Apesar de Gea e Luo (2001) não terem investigado a não linearidade física, o resultado obtido será comparado com as topologias ótimas linear e não linear geométrica. Além disso, as Fig.

6.17 a 6.19 mostram gráficos que descrevem o comportamento de cada um destes parâmetros ao longo dos processos iterativos das três análises.

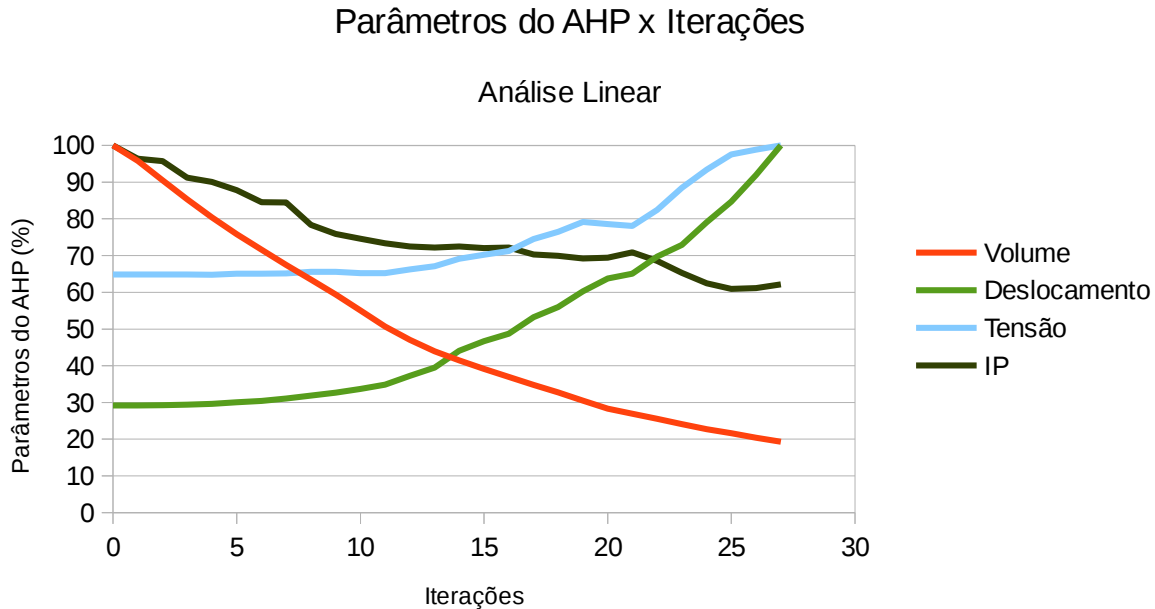


Figura 6.17 – Parâmetros do Vetor WSM (Volume, Deslocamentos, Tensão, Índice de Performance) x Iterações: Análise Linear

Na análise linear, o volume decresce praticamente de forma linear, com uma suavização a partir do meio da análise que é mantida até o fim. Isso é refletido inicialmente nos deslocamentos e na tensão, já que ambos se mantêm praticamente constantes até próximo da iteração 10, onde os primeiros vazios começam a se formar no interior do modelo. A Fig 6.5, que representa a iteração 14, já mostra um estágio mais avançado da formação de membros internos, o que configura um aumento relativo dos deslocamentos e uma leve elevação dos valores da tensão. Na iteração que corresponde ao ótimo (Fig. 6.7), há um leve decréscimo da tensão, onde são definidos exatamente os membros que suportam o carregamento indicado. O índice de performance, em contrapartida, mostra um ligeiro aumento, justificando a considerável perda de material da estrutura relativo aos membros superiores ainda visíveis na iteração 14 (Fig. 6.5). É importante ressaltar que a topologia referente à última iteração (Fig. 6.6) mostra um início de ruptura do membro, até então, importante para a estabilidade, tornando esta topologia inadequada por ser quase hipostática.

O aumento de deslocamento correspondente a toda a análise é da ordem de 3,5 vezes em relação ao inicial e a tensão é cerca de 1,55 vezes maior que a inicial.

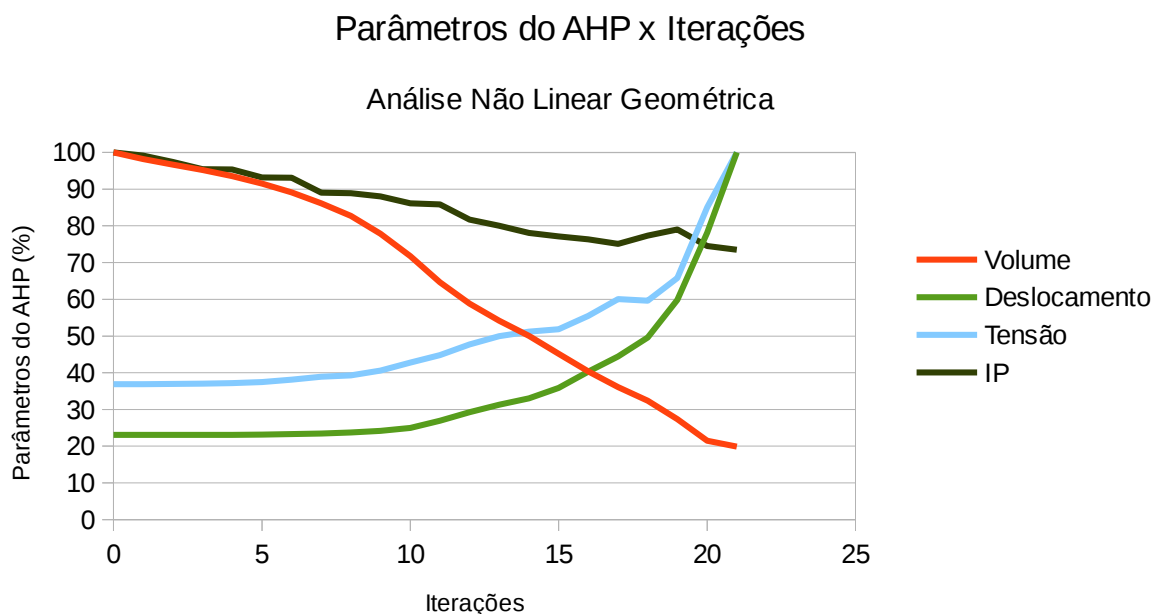


Figura 6.18 – Parâmetros do Vetor WSM (Volume, Deslocamentos, Tensão, Índice de Performance) x Iterações: Análise não Linear Geométrica

Na análise não linear geométrica, o volume, ao contrário da análise linear, é penalizado suavemente no início do processo de otimização e, em seguida, há uma queda mais acentuada, que se inicia a partir da iteração 7. Este comportamento é mantido até o fim, praticamente. Tanto deslocamento quanto tensão começam praticamente constantes até que, próximo à iteração 11 começam a aumentar (Fig. 6.9). Novamente, nesta iteração é possível notar a formação de vazios internos e, assim como em exemplos anteriores, com o início do surgimento destes vazios ocorre o aumento de deslocamento e tensão. Também, a topologia referente à última iteração (Fig. 6.10) mostra os membros inferiores totalmente removidos, fazendo com que a carga seja suportada apenas pelos membros principais superiores. Alguns elementos esparsos podem ser notados, representando uma dependência da malha. Apesar da ausência destes membros inferiores não tornar a estrutura hipostática e, de certa forma, contribui muito para um aumento exagerado da tensão e dos deslocamentos, o que torna inadequada a resposta. A tensão nesta última iteração, 21, é da ordem de 2,7 vezes a tensão inicial e os deslocamentos são cerca de 4,5 vezes maior que o inicial. O índice de performance

mostra um leve aumento na iteração 19, justamente quando ocorre a remoção dos membros inferiores.

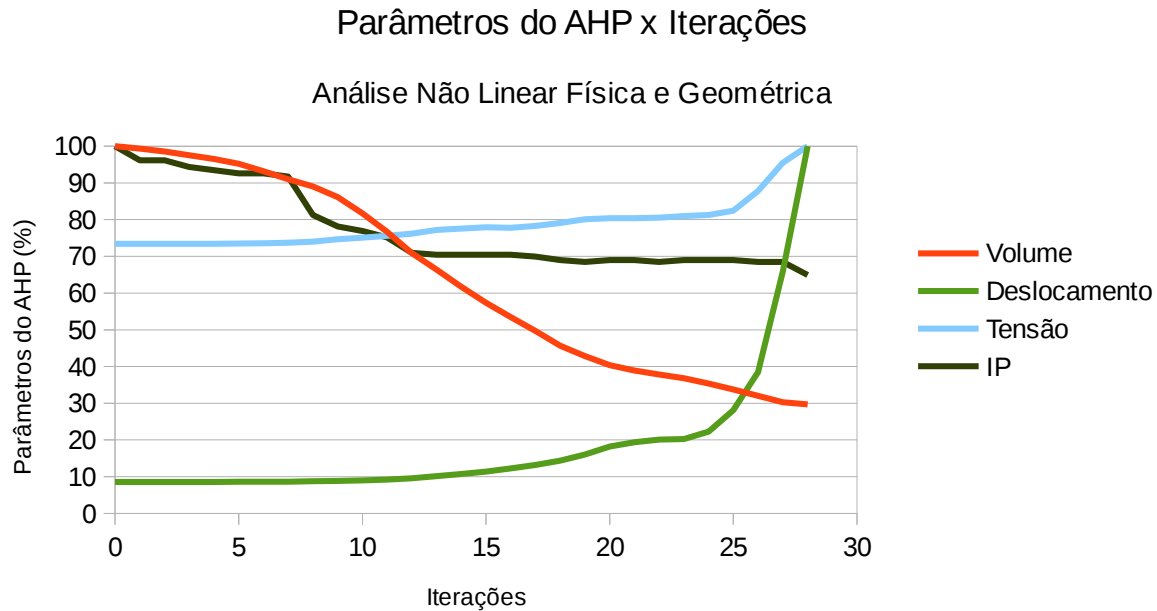


Figura 6.19 – Parâmetros do Vetor WSM (Volume, Deslocamentos, Tensão, Índice de Performance) x Iterações: Análise não Linear Física e Geométrica

Na análise não linear física e geométrica, o comportamento do volume é similar ao da análise não linear geométrica, ou seja, também é penalizado suavemente no início do processo de otimização e, em seguida, há uma queda mais acentuada, que se inicia a partir da iteração 9. Ao fim da análise, perto da iteração 20, esta suavização volta a aparecer. Os deslocamentos pouco variam até a iteração 15, onde ocorre um leve aumento e, logo em seguida, um repentino e acentuado aumento pode ser notado a partir da iteração 24. A tensão também experimenta um aumento expressivo a partir desta mesma iteração, uma vez que se manteve sem significativas variações até então. Inclusive, convém ressaltar que esta iteração corresponde exatamente ao ótimo do problema (Fig. 6.15). Nas iterações seguintes, onde o aumento da tensão e deslocamento se tornam evidentes, percebe-se a ruptura dos membros superiores (Fig. 6.14) e a análise é então interrompida por não convergir adequadamente, sendo este o único exemplo onde o processo de otimização não é concluído aos 20% de volume inicial (Fig. 6.20). A análise, então, termina com o volume da última iteração com cerca de 29,7% do volume inicial.

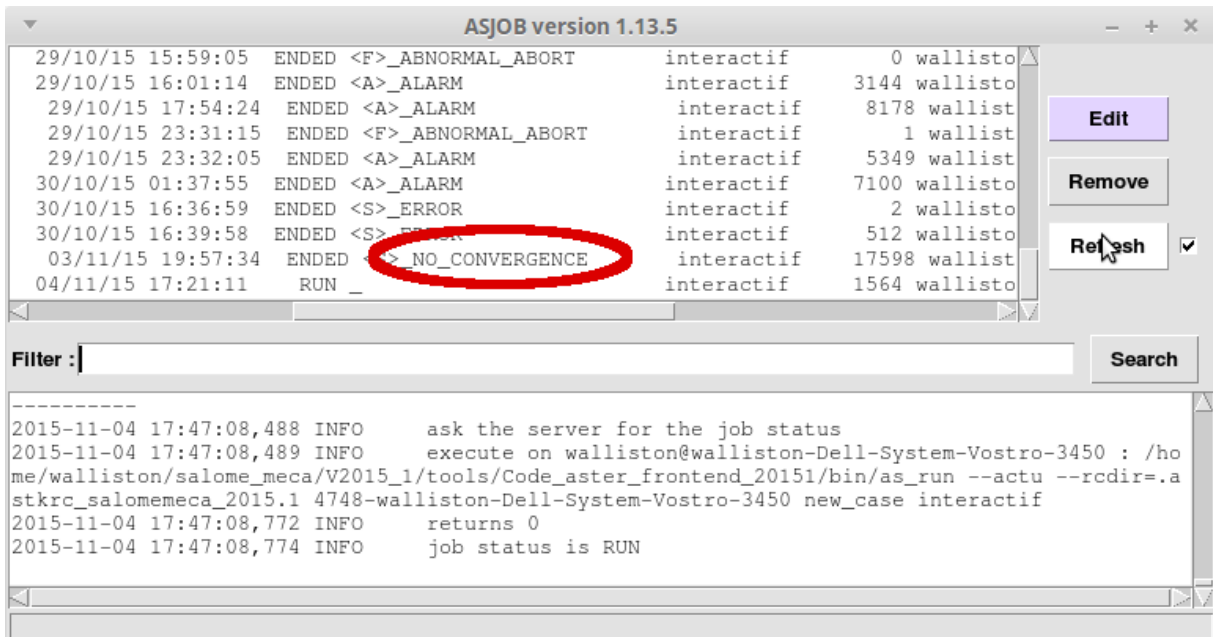


Figura 6.20 – Falha da convergência da análise não linear física e geométrica

Uma comparação entre as três análises é resumida na Fig. 6.21, mostrando os valores dos respectivos vetores WSM ao longo do processo iterativo. O ponto preto mostra a iteração com o valor mínimo de cada análise, que representa a melhor solução para cada problema.

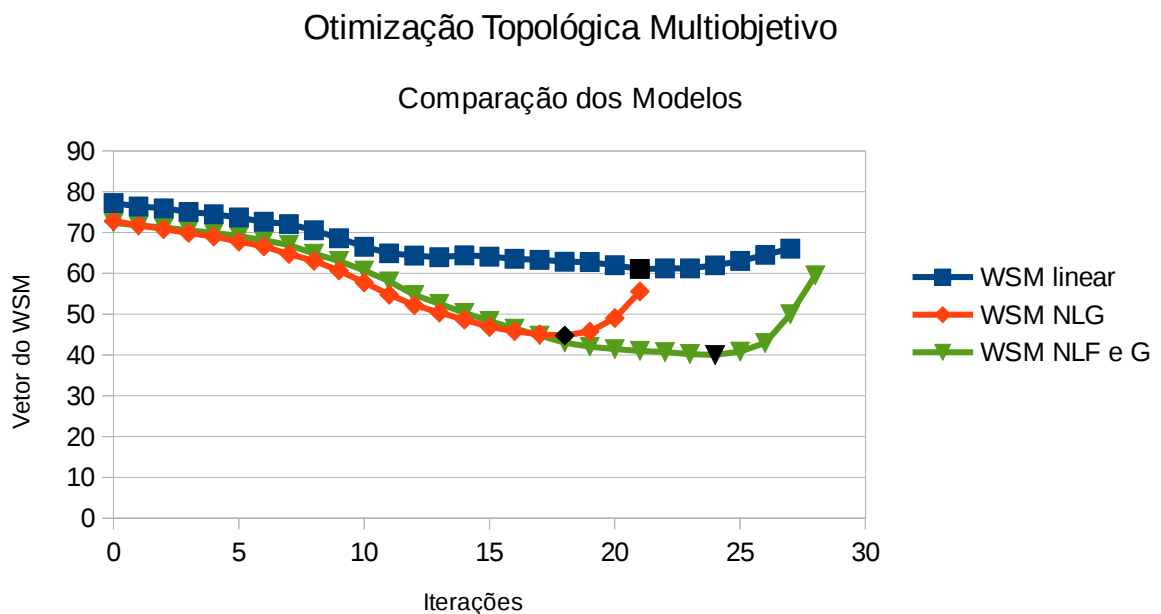


Figura 6.21 – Comparação dos Vetores do WSM x Iterações

O vetor WSM linear possui valor mínimo na iteração 21, configurando esta como a solução para o conjunto de possíveis soluções para a análise. Nesta iteração, o volume é de aproximados 27% em relação ao volume inicial, os deslocamentos cerca de 2,2 vezes o inicial e a tensão é da ordem de 1,2 vezes a tensão máxima de toda a análise. O índice de performance reside na faixa dos 70%.

O vetor WSM não linear geométrico possui valor mínimo na iteração 18, configurando esta como a solução para o conjunto de possíveis soluções para a análise. Nesta iteração, o volume é de aproximados 36% em relação ao volume inicial, os deslocamentos cerca de duas vezes o inicial e a tensão é da ordem de 1,6 vezes a tensão máxima de toda a análise. O índice de performance reside na faixa dos 75%.

O vetor WSM não linear físico e geométrico possui valor mínimo na iteração 24, configurando esta como a solução para o conjunto de possíveis soluções para a análise. Nesta iteração, o volume é de aproximados 35% em relação ao volume inicial, os deslocamentos cerca de 2,6 vezes o inicial e a tensão é da ordem de 1,1 vezes a tensão máxima de toda a análise. O índice de performance reside na faixa dos 69%.

6.2 Viga em Balanço

Este modelo consiste em uma viga curta em balanço, cujos apoios são configurados como em um tripé. A carga é aplicada na extremidade do balanço e possui as seguintes dimensões, de acordo com a Fig. 6.22.

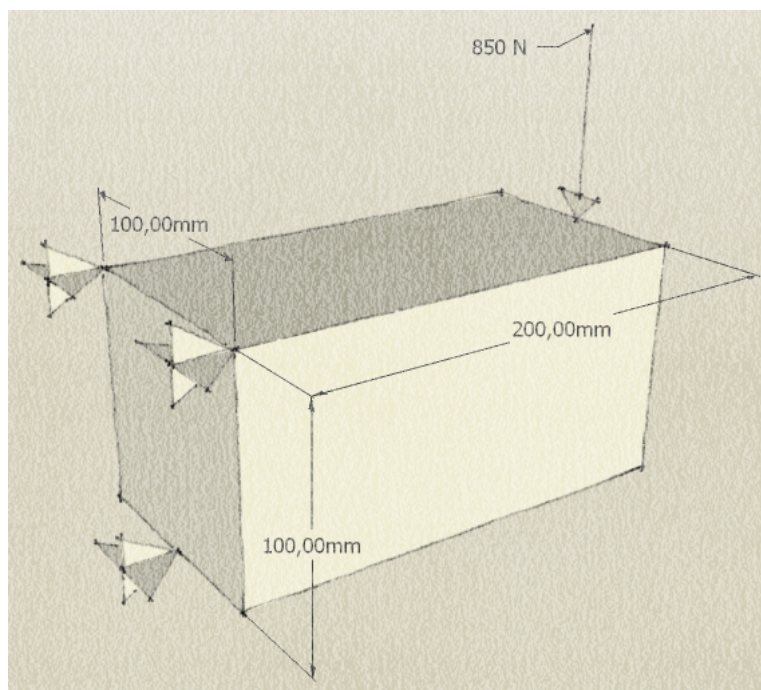


Figura 6.22 – Domínio inicial da viga em balanço

Para este problema, os dados pertinentes do modelo são os seguintes:

- Malha utilizada (*mesh*): a malha gerada no Code-Aster ® possui 10577 elementos do tipo TETRA4 (Fig. 6.23);
- Módulo de Elasticidade (E): 210 GPa;
- Tensão de Escoamento (σ_{esc}): 300MPa;
- Módulo de Encruamento Isotrópico (E_{ISO}): 536 MPa;
- Coeficiente de Poisson (ν): 0,3;
- Volume máximo desejado (VOLMAX) = 20%;
- Remoção máxima de volume mapeado por iteração (RVOL) = 95% (5% dos elementos são devolvidos para a estrutura);
- Fator de raio de rejeição (FRR) = 1,25%;
- Fator evolucionário de rejeição (FER) = 0,75%.

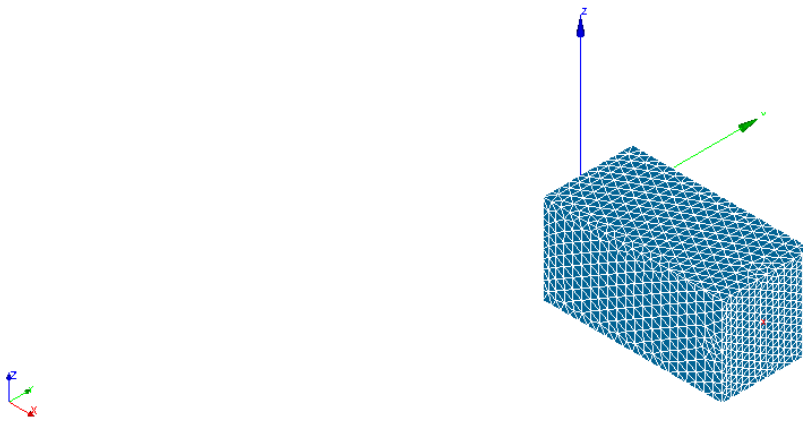


Figura 6.23 – Malha de elementos finitos gerada pelo Salome-Meca®

O processo evolutivo de otimização do problema da viga em balanço, para a análise linear, é mostrado nas Figs. 6.24 a 6.26.

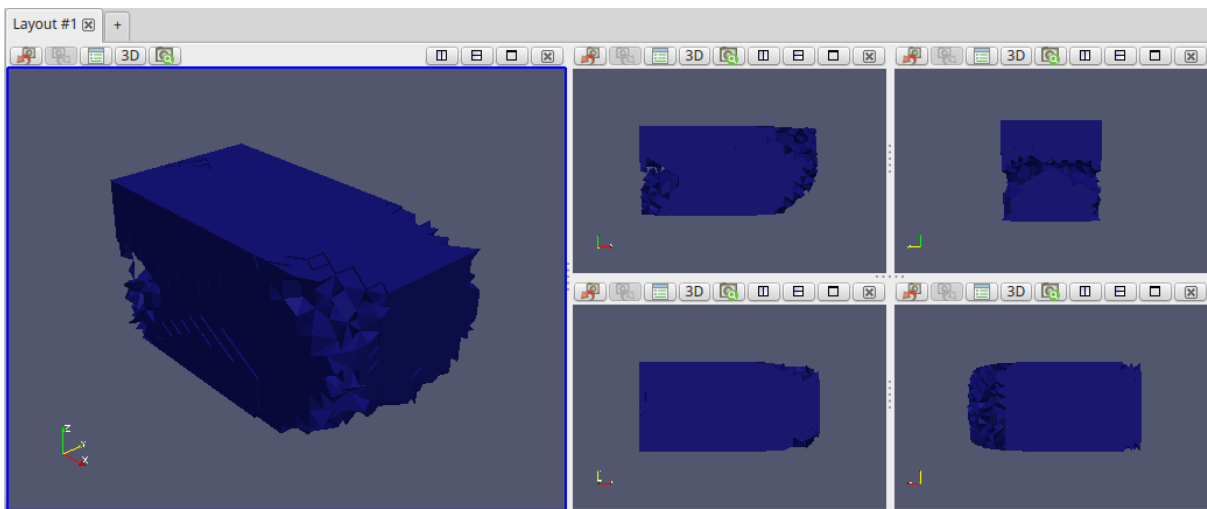


Figura 6.24 – Topologia parcial linear resultante do passo de otimização 3

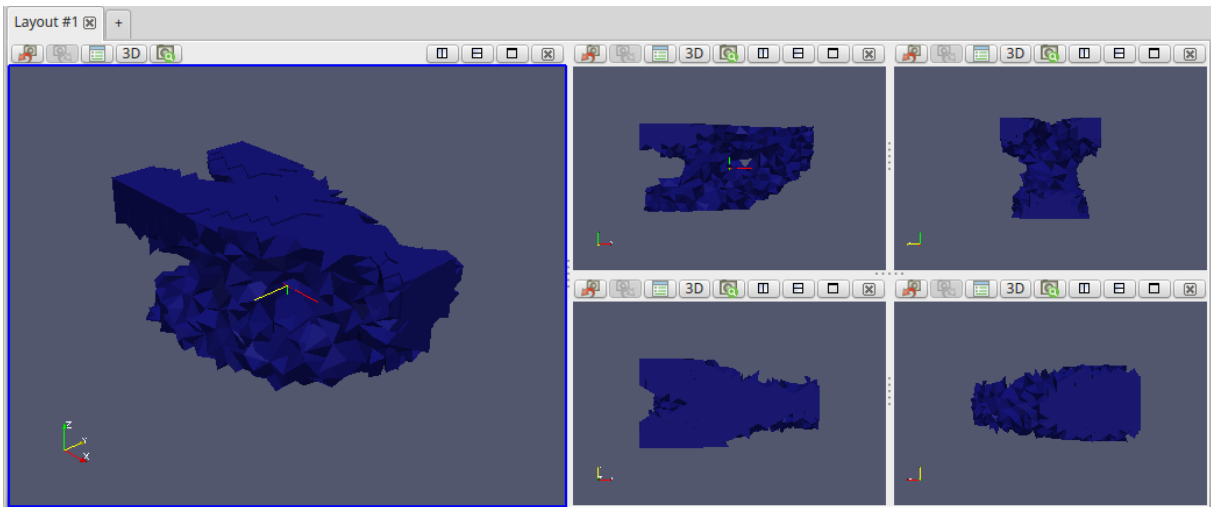


Figura 6.25 – Topologia parcial linear resultante do passo de otimização 10

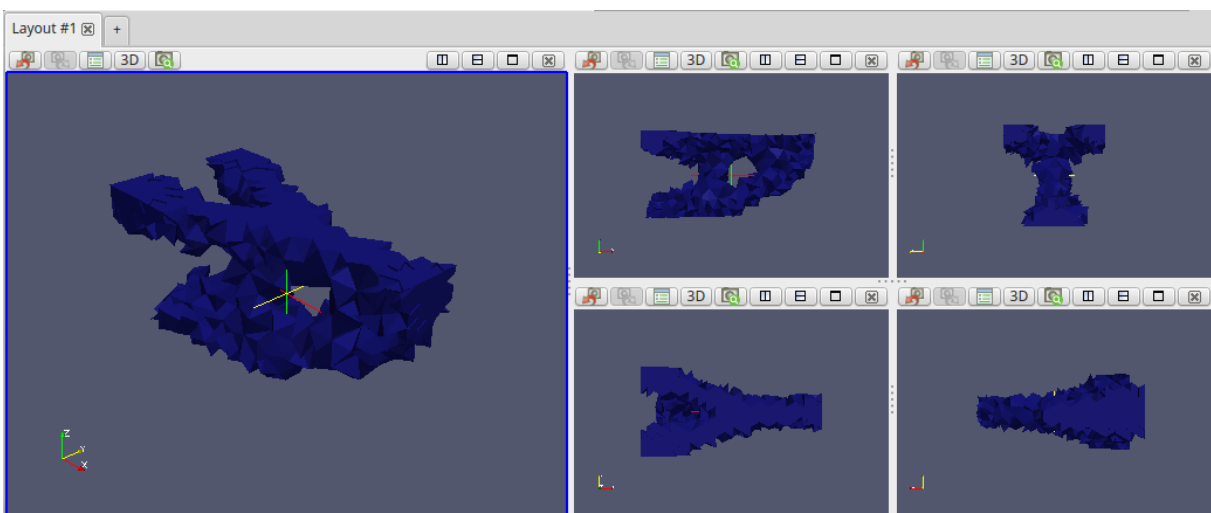


Figura 6.26 – Topologia parcial linear resultante do passo de otimização 21 (última iteração)

A topologia final ótima obtida para o problema considerando a análise linear é mostrada na Fig. 6.27.

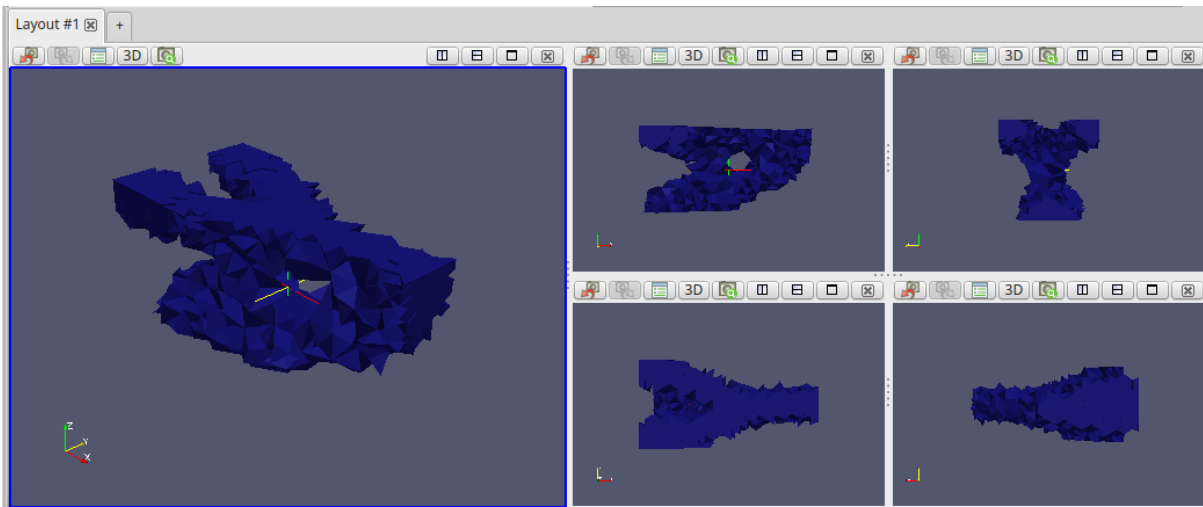


Figura 6.27 – Topologia final ótima linear resultante do passo de otimização 17

O processo evolutivo de otimização do problema da viga em balanço, para a análise não linear geométrica, é mostrado nas Figs. 6.28 a 6.30.

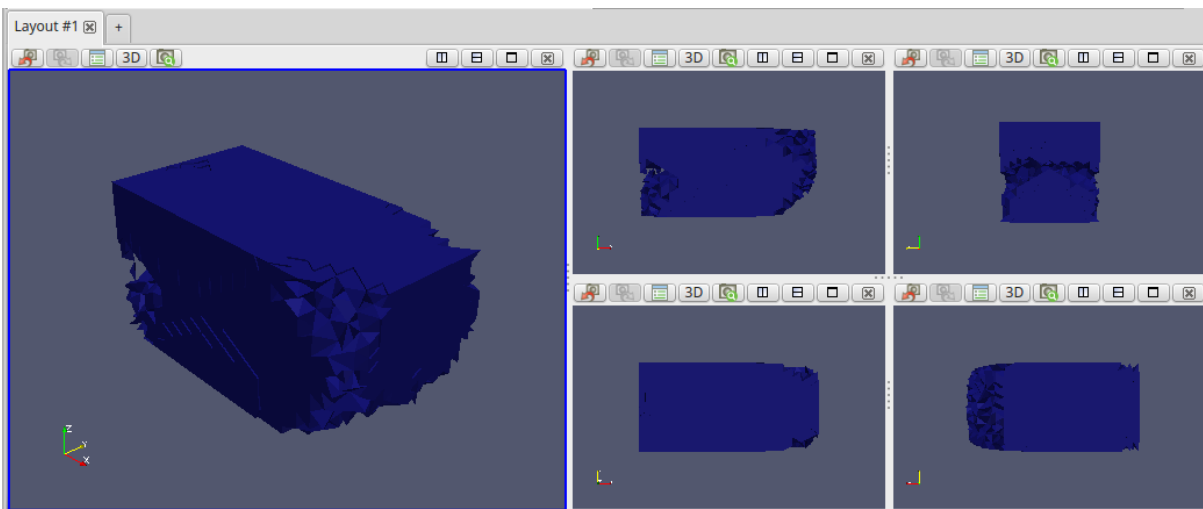


Figura 6.28 – Topologia parcial não linear geométrica resultante do passo de otimização 3

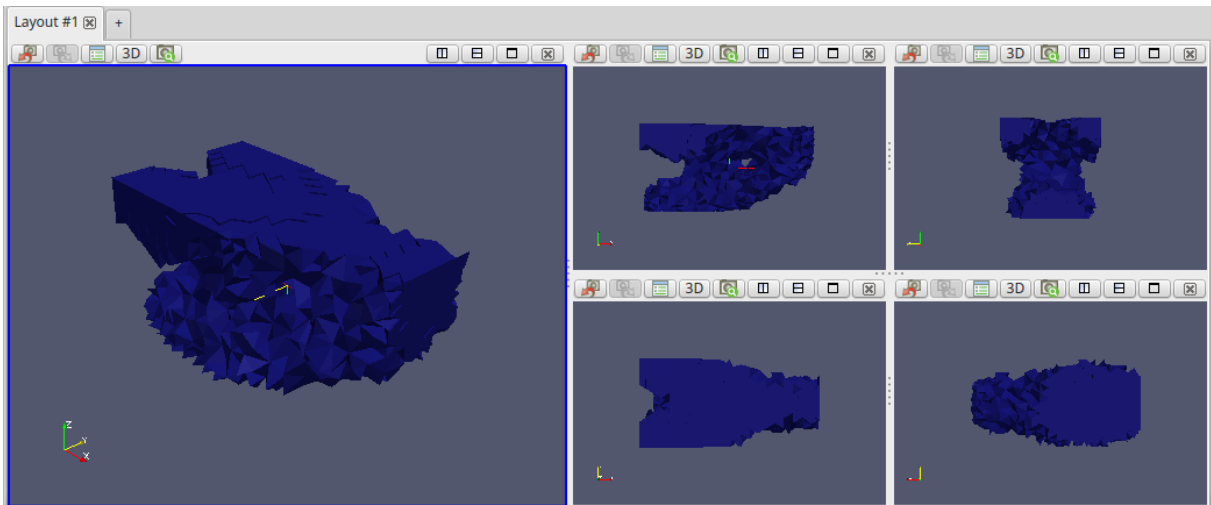


Figura 6.29 – Topologia parcial não linear geométrica resultante do passo de otimização 8

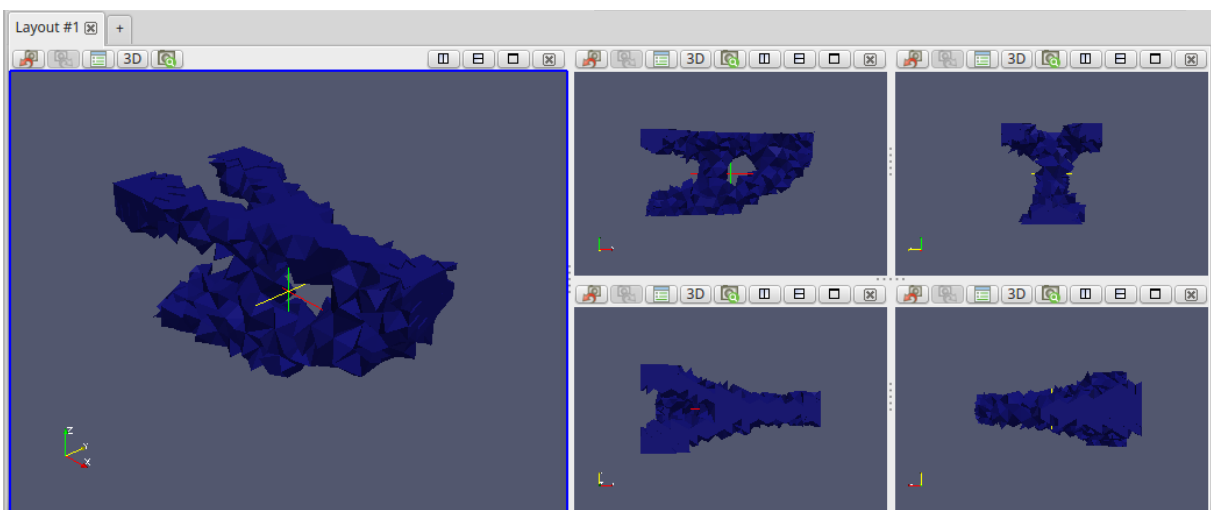


Figura 6.30 – Topologia parcial não linear geométrica resultante do passo de otimização 22
(última iteração)

A topologia final ótima obtida para o problema considerando a análise não linear geométrica é mostrada na Fig. 6.31.

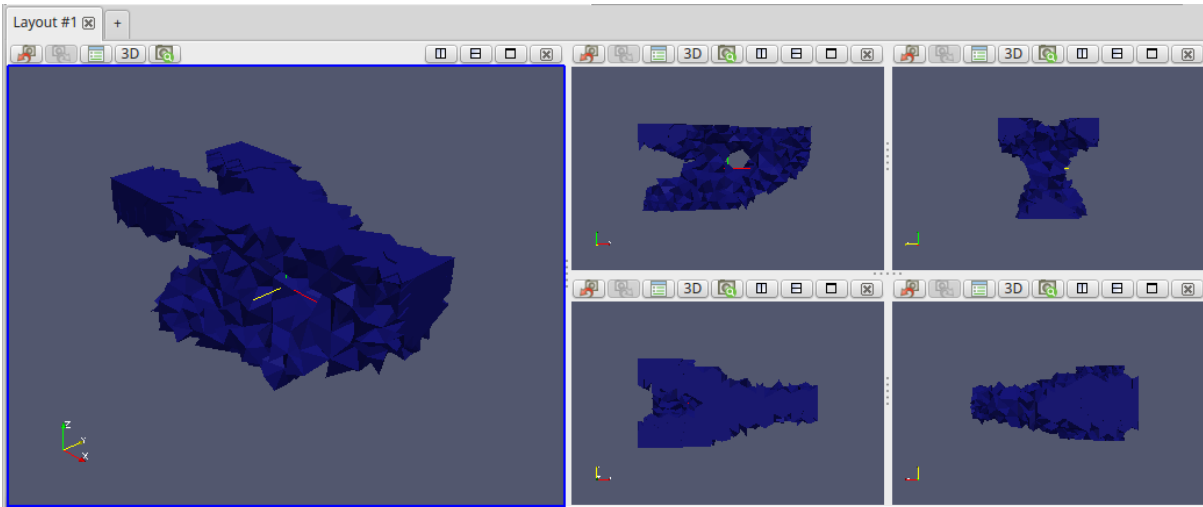


Figura 6.31 – Topologia final ótima não linear geométrica resultante do passo de otimização

15

O processo evolutivo de otimização do problema da viga em balanço, para a análise não linear física e geométrica, é mostrado nas Figs. 6.32 a 6.34.

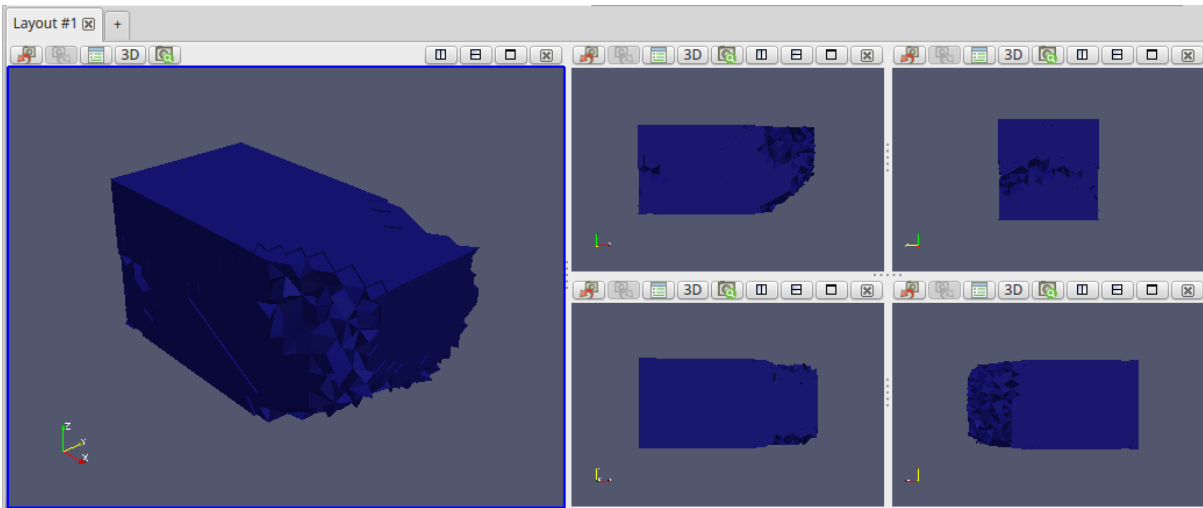


Figura 6.32 – Topologia parcial não linear física e geométrica resultante do passo de otimização 7

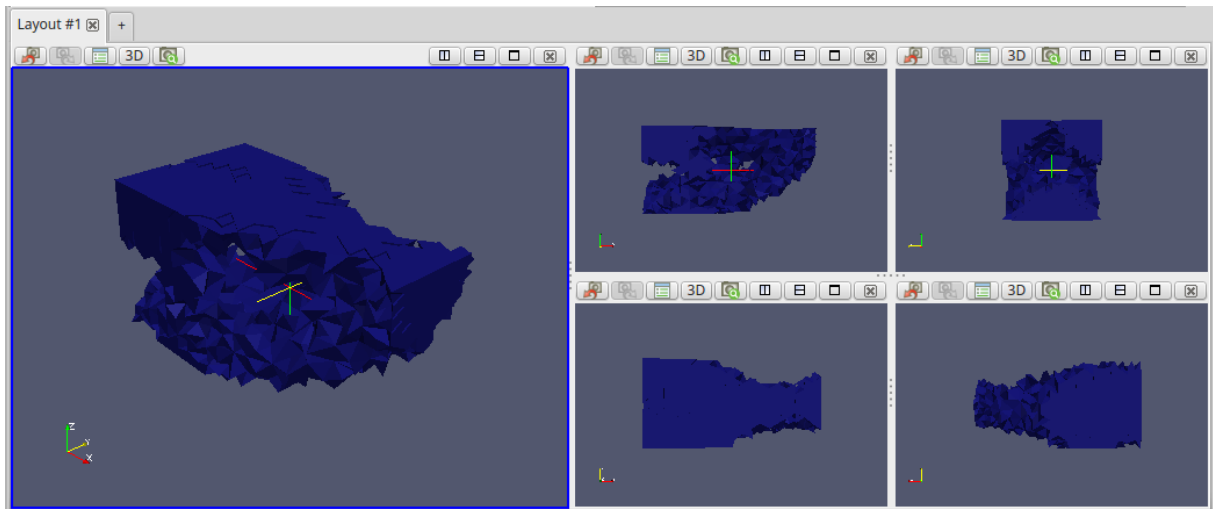


Figura 6.33 – Topologia parcial não linear física e geométrica resultante do passo de otimização 19

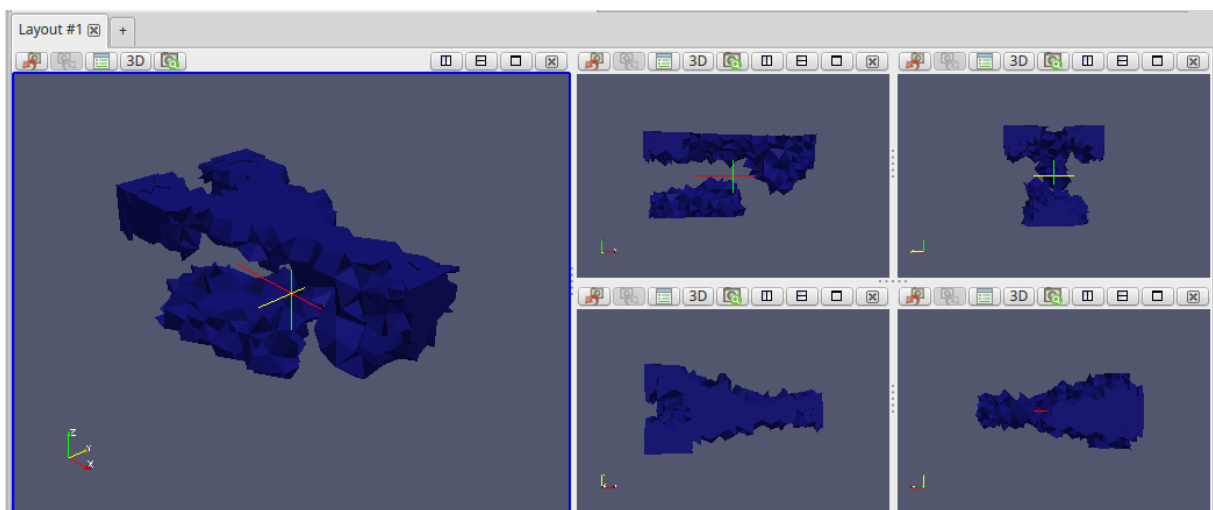


Figura 6.34 – Topologia parcial não linear física e geométrica resultante do passo de otimização 52 (última iteração)

A topologia final ótima obtida para o problema considerando a análise não linear física e geométrica é mostrada na Fig. 6.35.

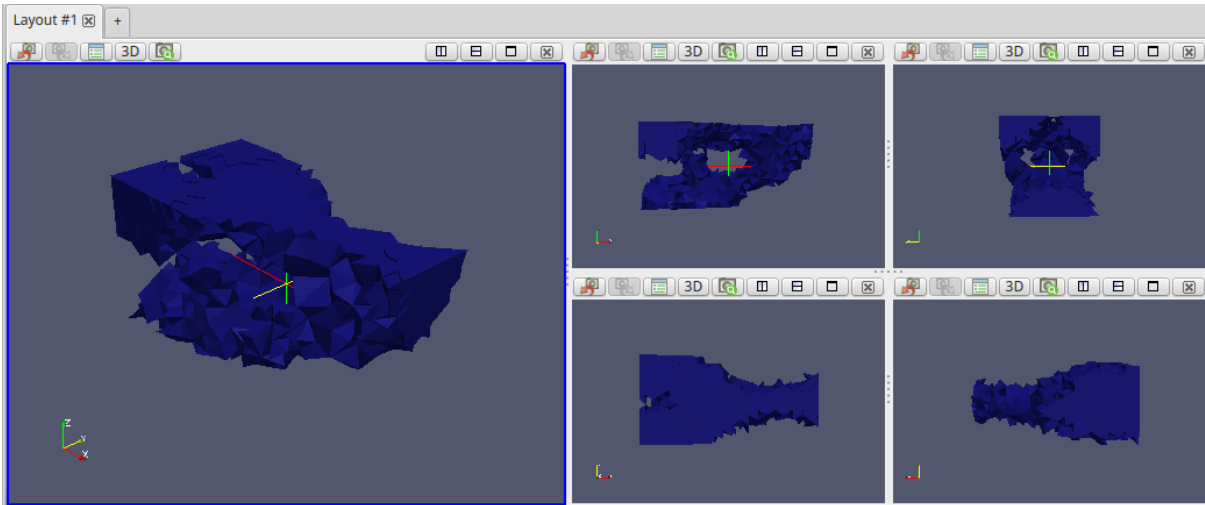


Figura 6.35 – Topologia final ótima não linear física e geométrica resultante do passo de otimização 34

As Fig. 6.36 a 6.38 mostram gráficos que descrevem o comportamento de cada um destes parâmetros ao longo dos processos iterativos das três análises.

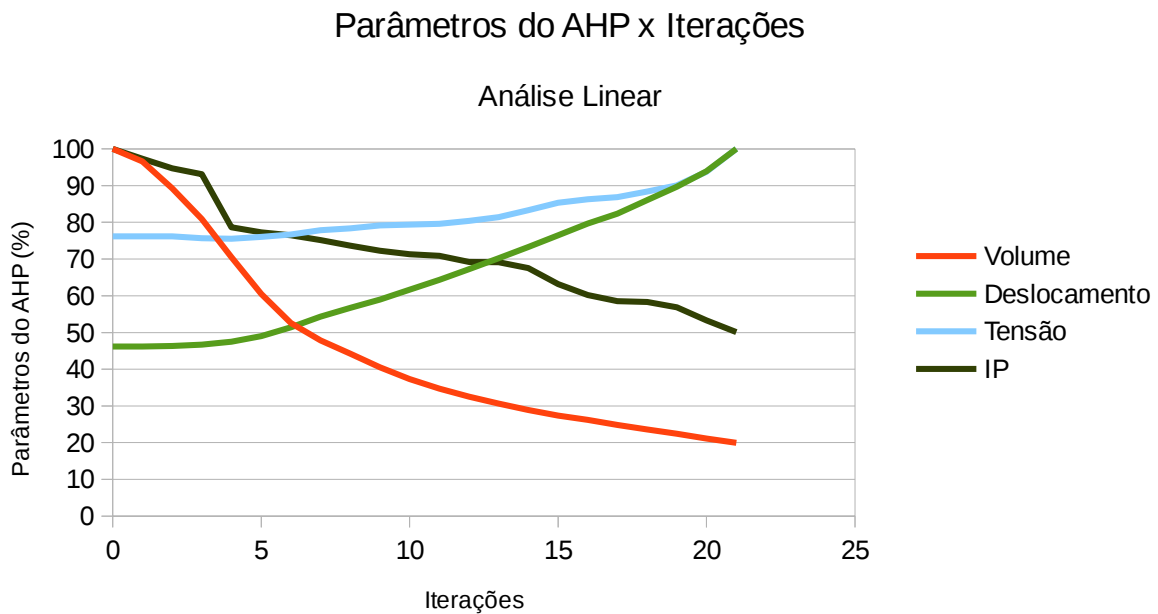


Figura 6.36 – Parâmetros do Vetor WSM (Volume, Deslocamentos, Tensão, Índice de Performance) x Iterações: Análise Linear

Na análise linear, o volume decresce rapidamente até a iteração 7 e, a partir de então, a

penalização ocorre de forma mais suavizada até o fim do processo. Os deslocamentos passam a aumentar de forma praticamente linear a partir da iteração 5 até o fim da análise. A variação considerando o maior deslocamento de toda a análise comparado ao inicial é da ordem de 2,15 vezes. Comparada aos outros parâmetros, a tensão não apresenta grandes variações ao longo do processo iterativo, apesar de aumentar gradualmente e quase linearmente a partir da iteração 5. O aumento total considerando a primeira e a última iterações para a tensão foi da ordem de 1,3 vezes. O índice de performance decresce a cada iteração, apenas com um destaque na iteração 4, onde há uma queda acentuada. Um fato interessante deste exemplo é que embora haja a formação de vazios internos no modelo (formando membros nos apoios), a tensão não aumenta tanto uma vez que a viga é curta (o comprimento do balanço corresponde a duas vezes as outras dimensões). Há, portanto, a formação de um membro ao centro do modelo (próximo ao vazio formado) que contribui para a rigidez e estabilidade do modelo ao fazer a ligação entre os membros superiores (relativos aos apoios) e o inferior.

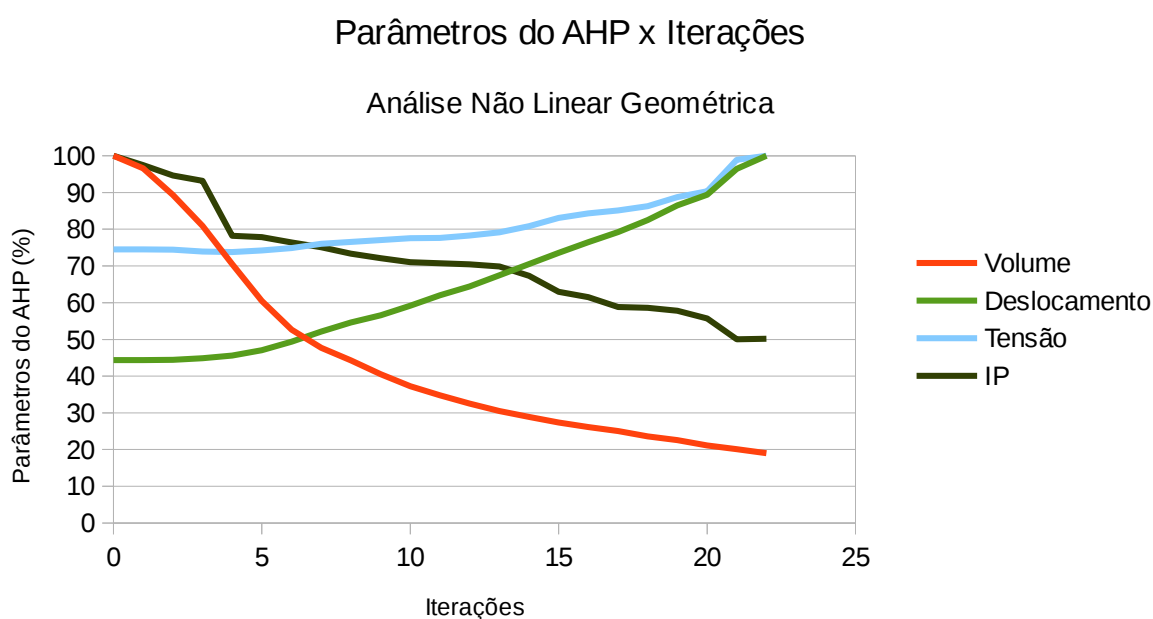


Figura 6.37 – Parâmetros do Vetor WSM (Volume, Deslocamentos, Tensão, Índice de Performance) x Iterações: Análise não Linear Geométrica

Na análise não linear geométrica, o comportamento dos parâmetros é semelhante ao linear, com pequenas diferenças em relação aos valores dos deslocamentos ao longo do processo. Além da análise não linear geométrica necessitar de uma iteração a mais que a linear (o

processo de penalização ocorreu de forma mais gradual), sua respectiva topologia ótima possui mais elementos comparada à resposta ótima da análise linear. Em outras palavras, mesmo com uma iteração a mais em relação a análise linear para o término do processo iterativo, a topologia ótima final corresponde à iteração 15 enquanto que na análise linear a topologia ótima final se encontra na iteração 17. Assim, embora ambas topologias sejam semelhantes, a estrutura obtida pela análise não linear se mantém mais rígida.

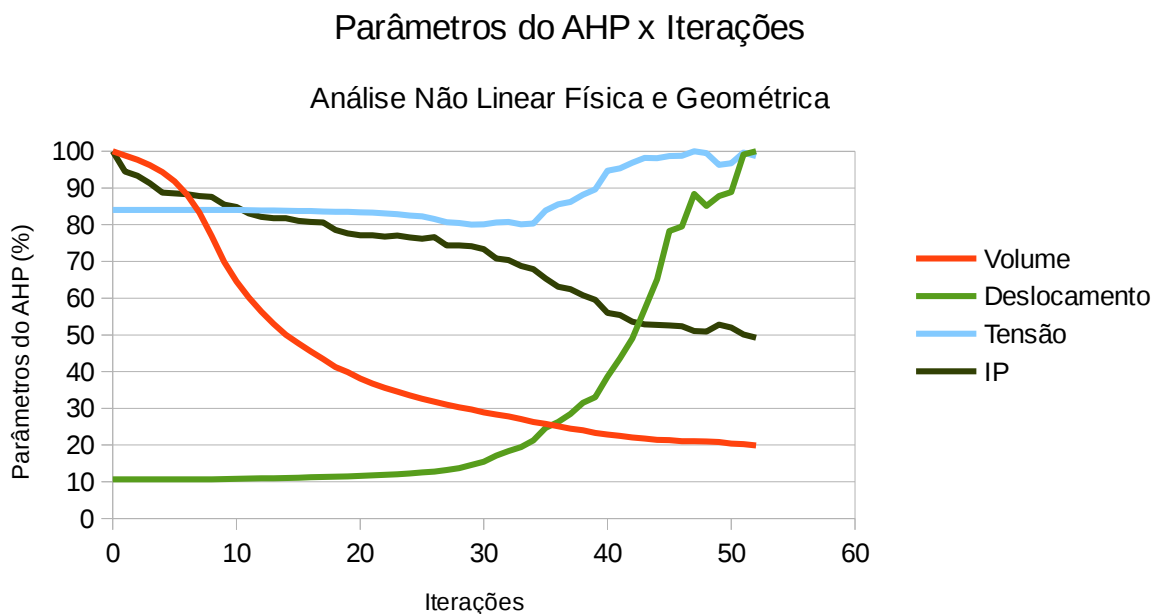


Figura 6.38 – Parâmetros do Vetor WSM (Volume, Deslocamentos, Tensão, Índice de Performance) x Iterações: Análise não Linear Física e Geométrica

Na análise não linear física e geométrica, o comportamento do volume é similar ao das outras análises, embora a topologia ótima final não seja. Os deslocamentos passam a ter uma variação significativa a partir da iteração 30, aproximadamente. Deste ponto em diante, o aumento é expressivo, chegando a mais de 10 vezes mais se comparando a primeira com a última iteração. O comportamento da tensão é interessante, mostrando uma sutil queda a partir da iteração 22 para, logo em seguida, mostrar um aumento gradual da última iteração da ordem de 1,17 vezes em relação à primeira. O índice de performance tem um comportamento decrescente durante quase toda a análise, sem oscilações relevantes. Neste exemplo, quando há a ruptura do modelo nas últimas iterações, os deslocamentos passam a ser grandes o suficiente para a estrutura tender a se tornar hipostática. Neste caso, a topologia ótima reside

na iteração 33, que corresponde aproximadamente ao início do brusco aumento dos deslocamentos.

Uma comparação entre as três análises é resumida na Fig. 6.39, mostrando os valores dos respectivos vetores WSM ao longo do processo iterativo. O ponto preto indica a iteração com o valor mínimo de cada análise, que representa a melhor solução para cada problema.

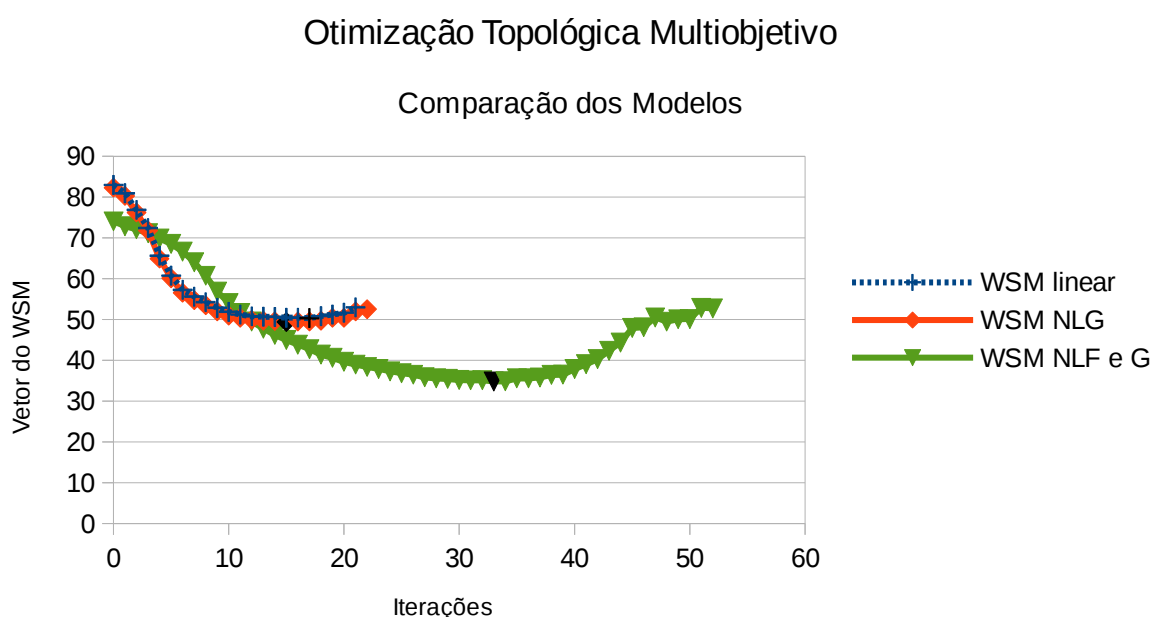


Figura 6.39 – Comparação dos Vetores do WSM x Iterações

O vetor WSM linear possui valor mínimo na iteração 17, configurando esta como a solução para o conjunto de possíveis soluções para a análise. Nesta iteração, o volume é de aproximados 25% em relação ao volume inicial, os deslocamentos cerca de 1,8 vezes o inicial e a tensão é da ordem de 1,15 vezes a tensão máxima de toda a análise. O índice de performance reside na faixa dos 58,5%.

O vetor WSM não linear geométrico possui valor mínimo na iteração 15, configurando esta como a solução para o conjunto de possíveis soluções para a análise. Nesta iteração, o volume é de aproximados 27% em relação ao volume inicial, os deslocamentos cerca de 1,65 vezes o inicial e a tensão é da ordem de 1,12 vezes a tensão máxima de toda a análise. O índice de performance reside na faixa dos 63%.

O vetor WSM não linear físico e geométrico possui valor mínimo na iteração 34, configurando esta como a solução para o conjunto de possíveis soluções para a análise. Nesta iteração, o volume é de aproximados 27% em relação ao volume inicial, os deslocamentos cerca de 2,6 vezes o inicial e a tensão é da ordem de 1,8 vezes a tensão máxima de toda a análise. O índice de performance reside na faixa dos 68,75%.

6.3 Viga em formato de “L”

Este modelo consiste em uma viga em formato de “L”, cujos apoios estão nas extremidades abertas, na parte inferior. Para impedir a rotação, foram introduzidos apoios na face superior do modelo, localizado a 1/3 da distância total. A carga é aplicada na junção entre as vigas e possui as seguintes dimensões, de acordo com a Fig. 6.40.

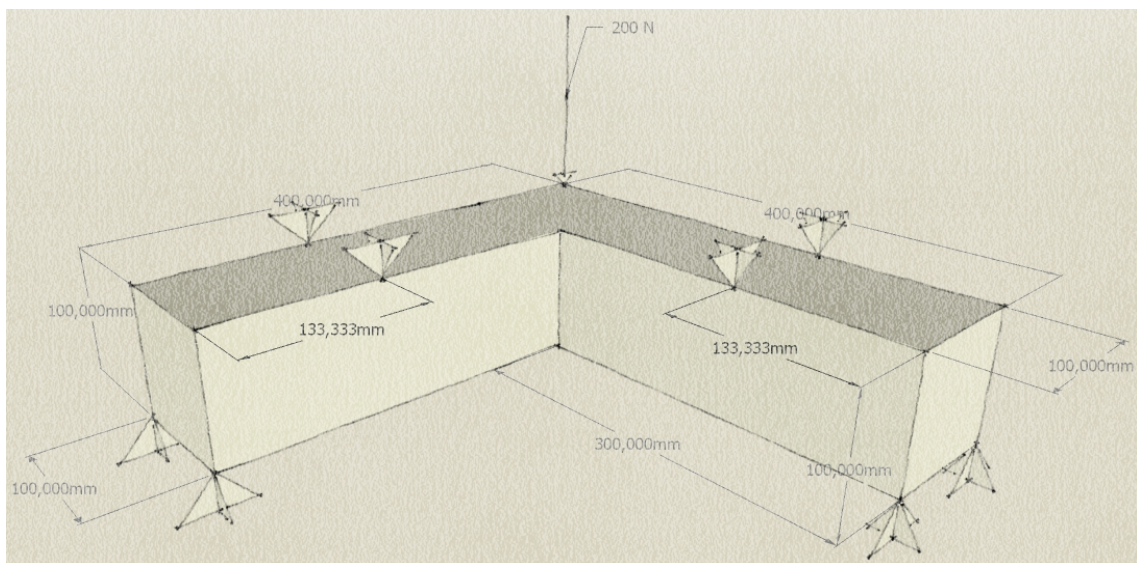


Figura 6.40 – Domínio inicial da viga em formato de “L”

Para este problema, os dados pertinentes do modelo são os seguintes:

- Malha utilizada (*mesh*): a malha gerada no Code-Aster® possui 9834 elementos do tipo TETRA4 (Fig. 6.41);
- Módulo de Elasticidade (E): 210 GPa;
- Tensão de Escoamento (σ_{esc}): 300MPa;

- Módulo de Encruamento Isotrópico (E_{ISO}): 536 MPa;
- Coeficiente de Poisson (ν): 0,3;
- Volume máximo desejado (VOLMAX) = 20%;
- Remoção máxima de volume mapeado por iteração (RVOL) = 95% (5% dos elementos são devolvidos para a estrutura);
- Fator de raio de rejeição (FRR) = 1,75%;
- Fator evolucionário de rejeição (FER) = 0,75%.

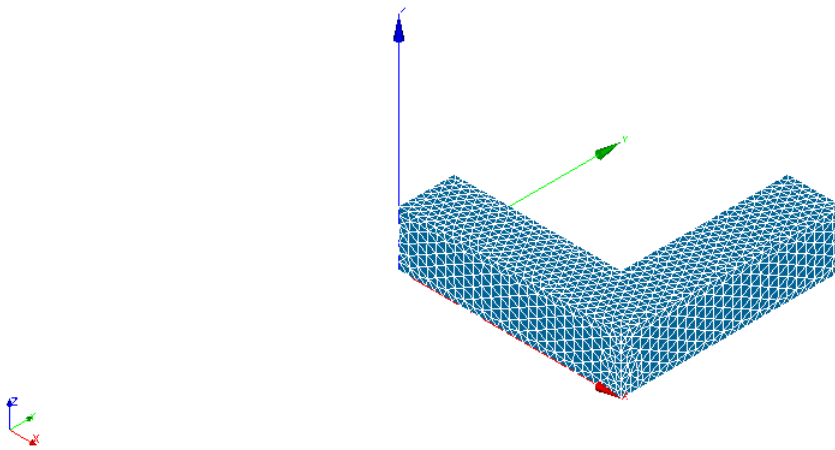


Figura 6.41 – Malha de elementos finitos gerada pelo Salome-Meca®

O processo evolutivo de otimização do problema da viga em formato de “L”, para a análise linear, é mostrado nas Figs. 6.42 a 6.44.

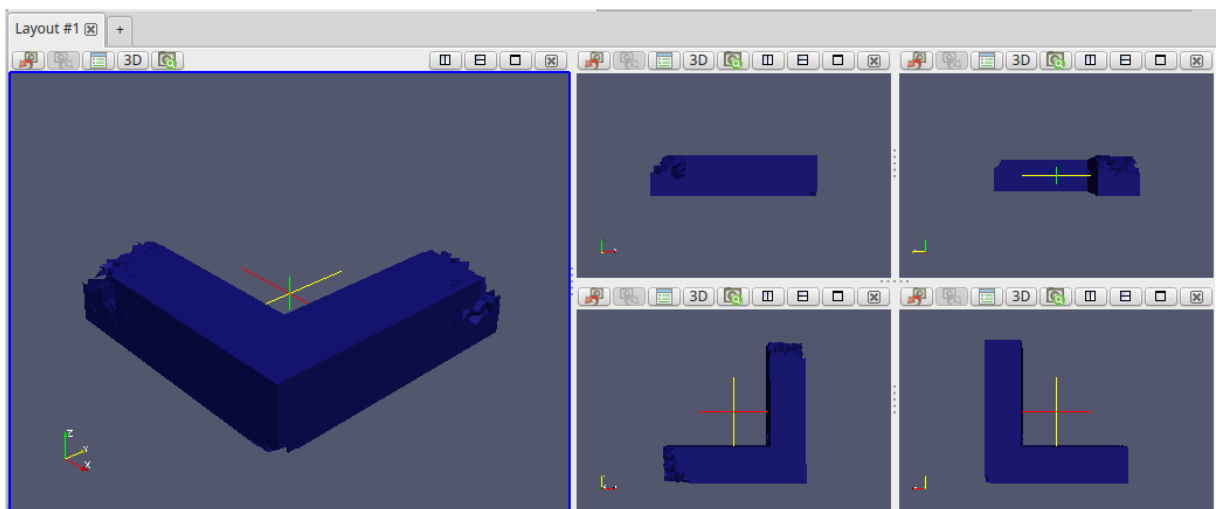


Figura 6.42 – Topologia parcial linear resultante do passo de otimização 4

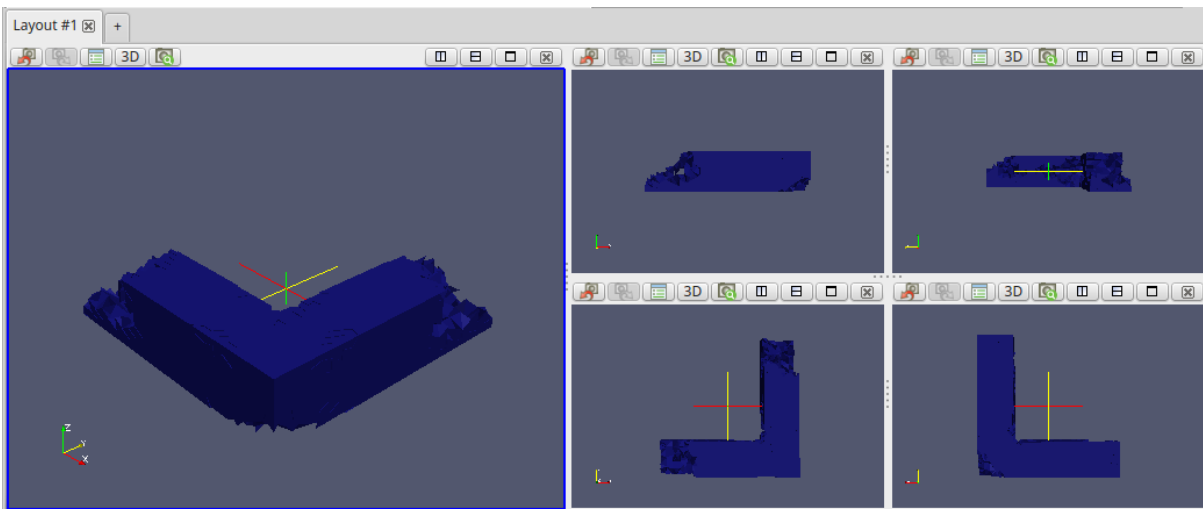


Figura 6.43 – Topologia parcial linear resultante do passo de otimização 10

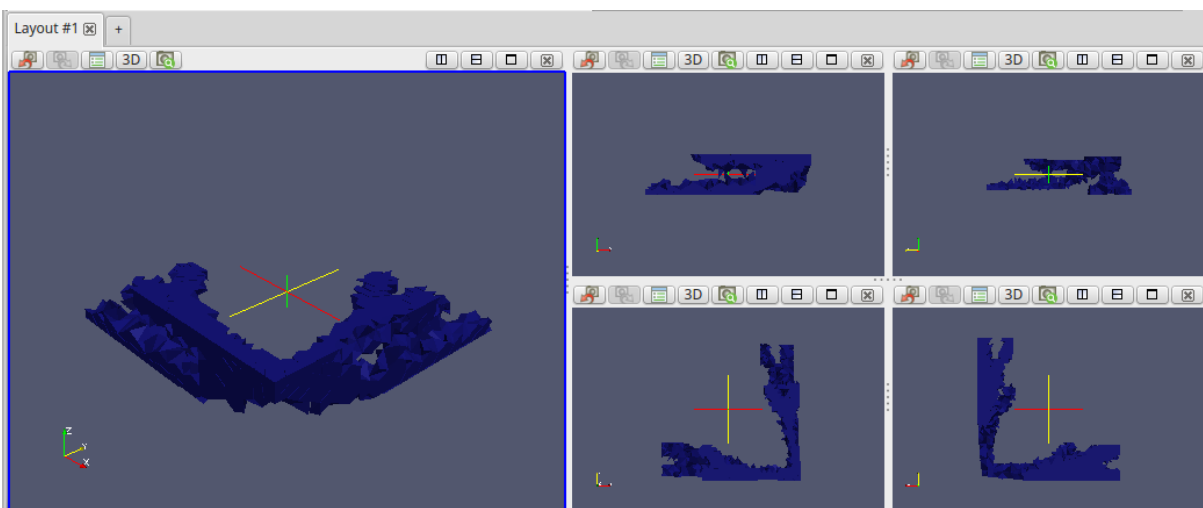


Figura 6.44 – Topologia parcial linear resultante do passo de otimização 22 (última iteração)

A topologia final ótima obtida para o problema considerando a análise linear é mostrada na Fig. 6.45.

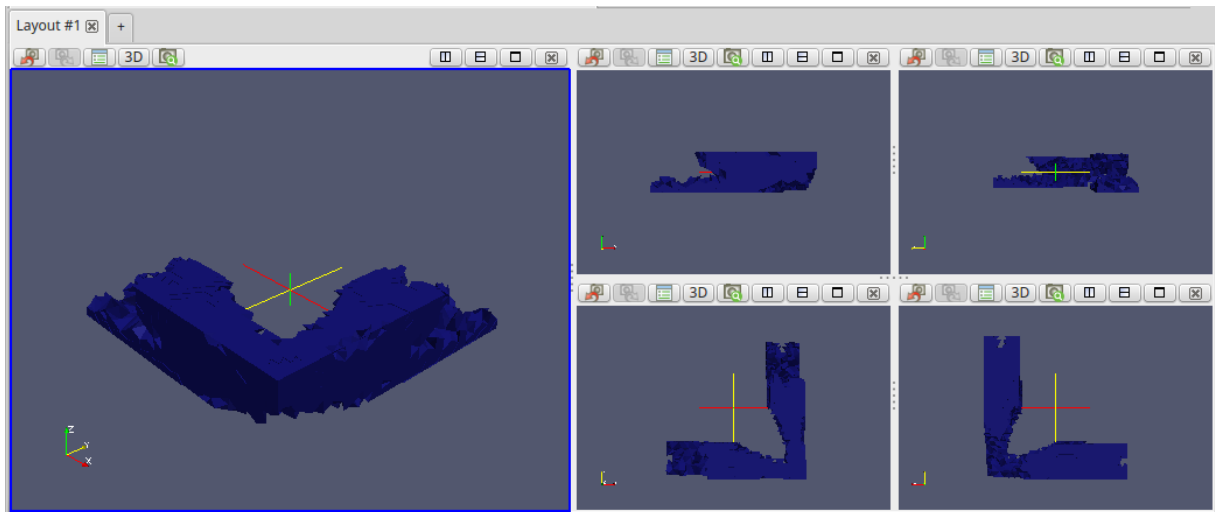


Figura 6.45 – Topologia final ótima linear resultante do passo de otimização 14

O processo evolutivo de otimização do problema da viga em formato de “L”, para a análise não linear geométrica, é mostrado nas Figs. 6.46 a 6.48.

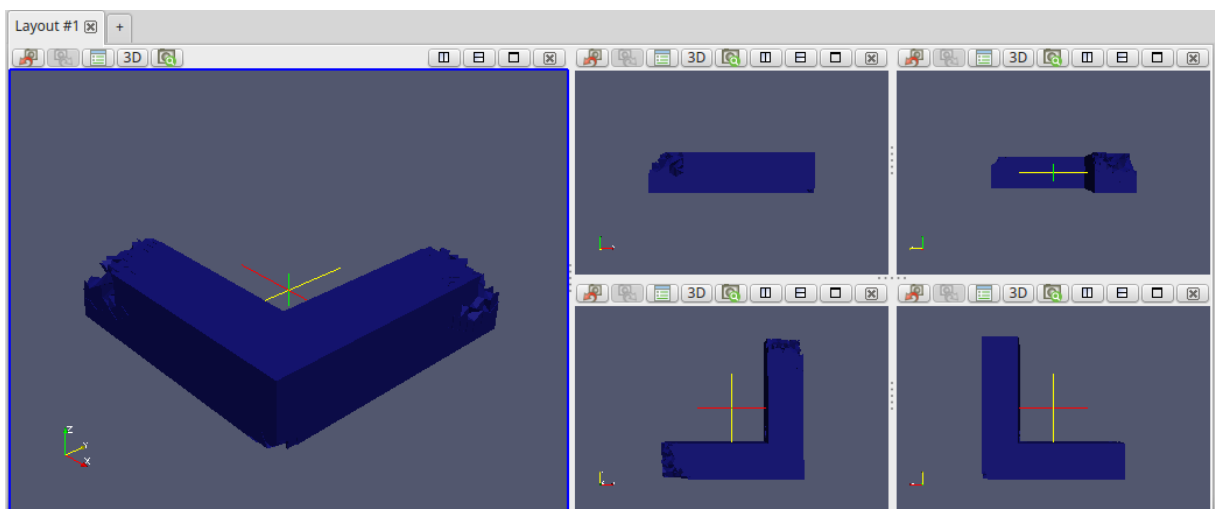


Figura 6.46 – Topologia parcial não linear geométrica resultante do passo de otimização 4

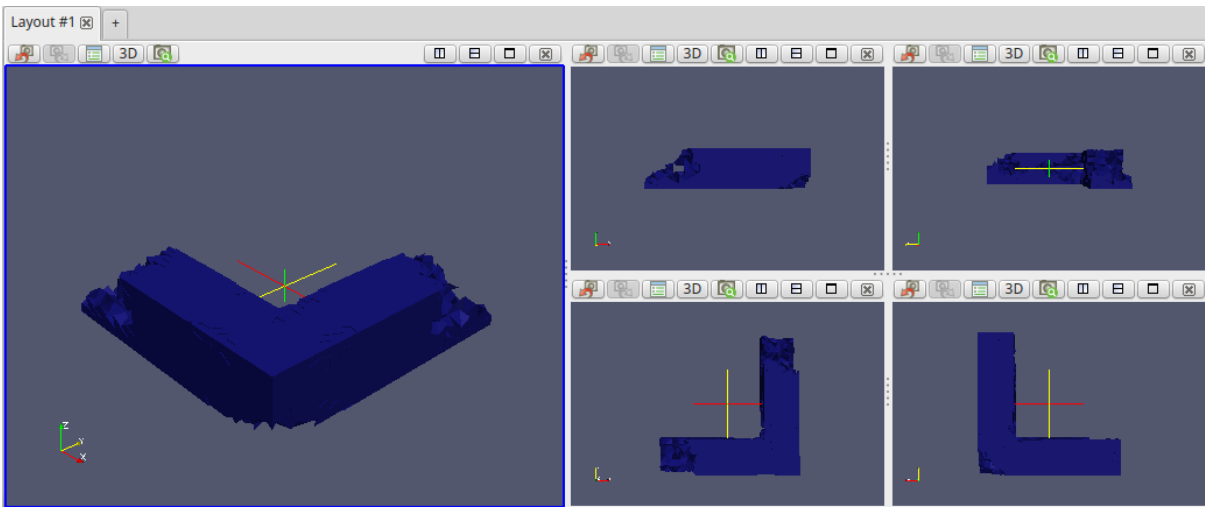


Figura 6.47 – Topologia parcial não linear geométrica resultante do passo de otimização 10

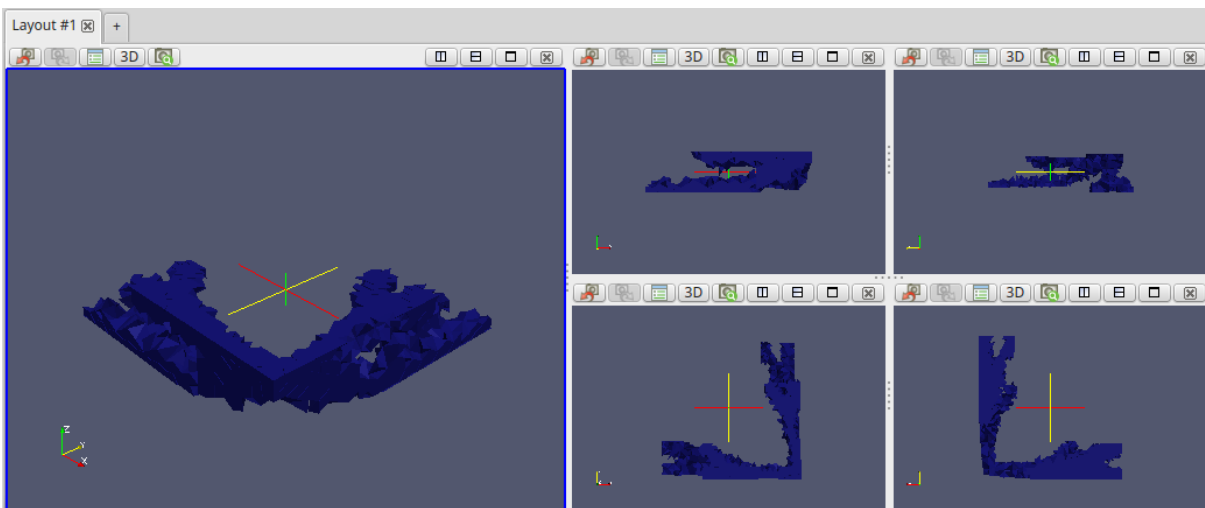


Figura 6.48 – Topologia parcial não linear geométrica resultante do passo de otimização 22
(última iteração)

A topologia final ótima obtida para o problema considerando a análise não linear geométrica é mostrada na Fig. 6.49.

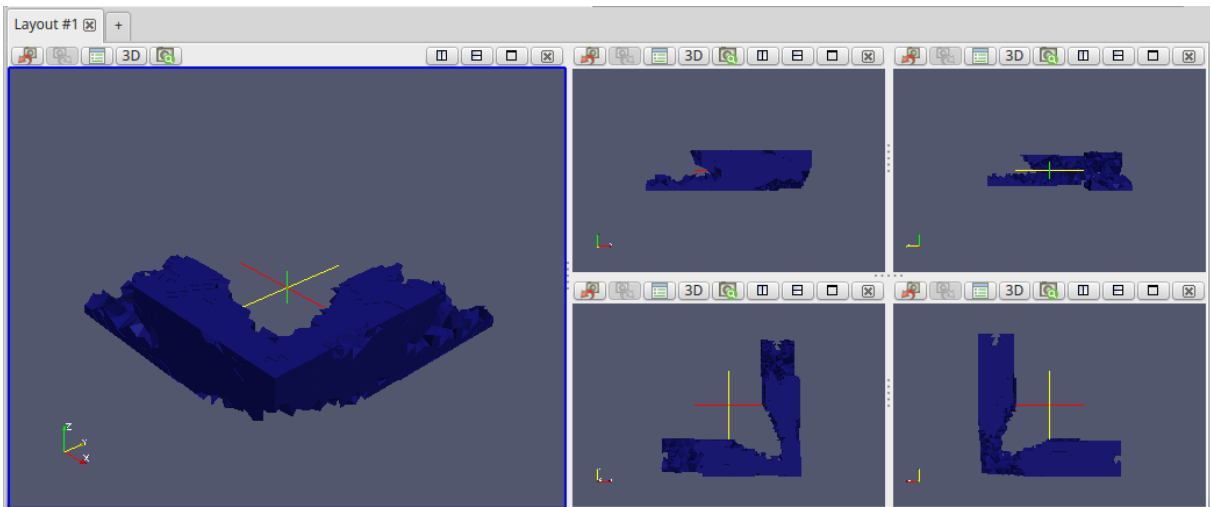


Figura 6.49 – Topologia final ótima não linear geométrica resultante do passo de otimização

14

O processo evolutivo de otimização do problema da viga em formato de “L”, para a análise não linear física e geométrica, é mostrado nas Figs. 6.50 a 6.52.

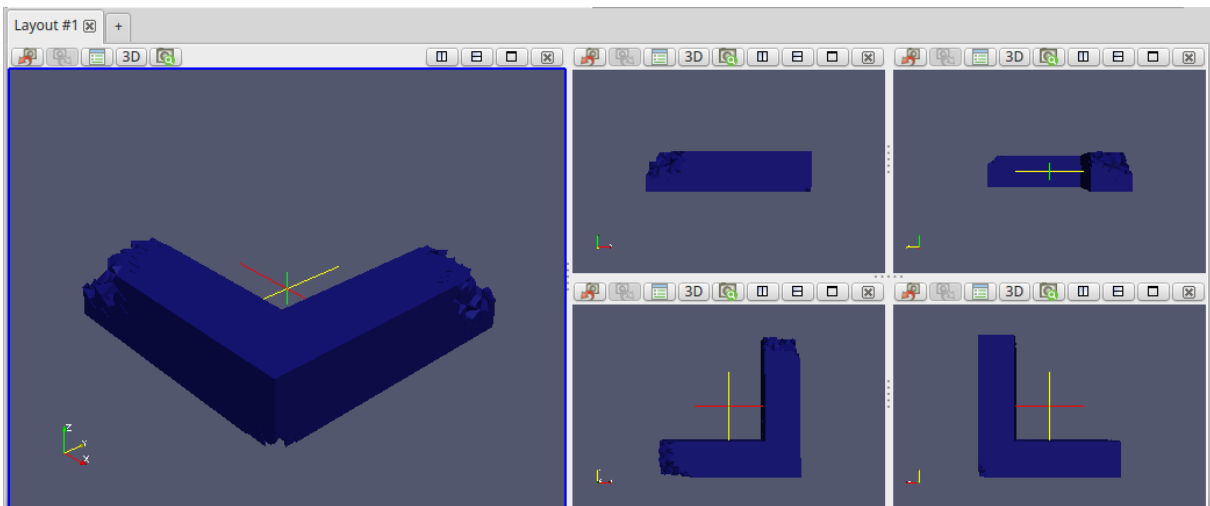


Figura 6.50 – Topologia parcial não linear física e geométrica resultante do passo de otimização 5

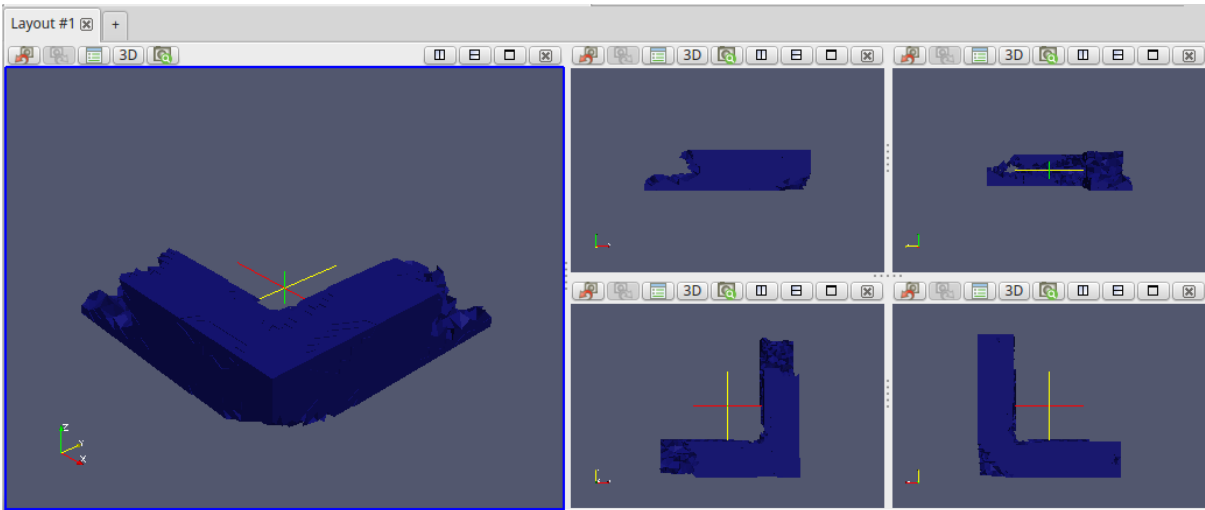


Figura 6.51 – Topologia parcial não linear física e geométrica resultante do passo de otimização 12

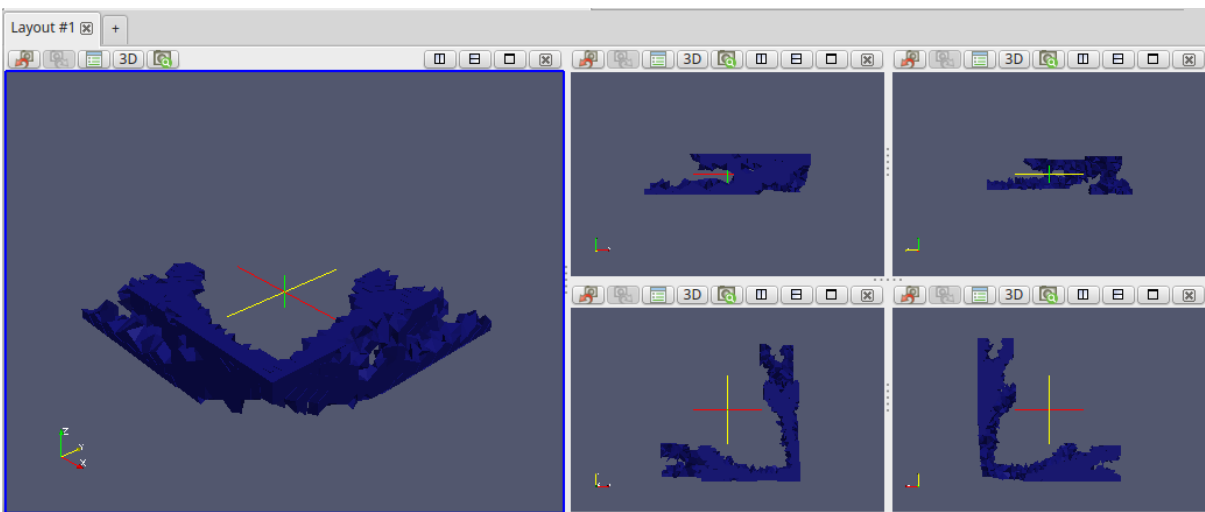


Figura 6.52 – Topologia parcial não linear física e geométrica resultante do passo de otimização 28 (última iteração)

A topologia final ótima obtida para o problema considerando a análise não linear física e geométrica é mostrada na Fig. 6.53.

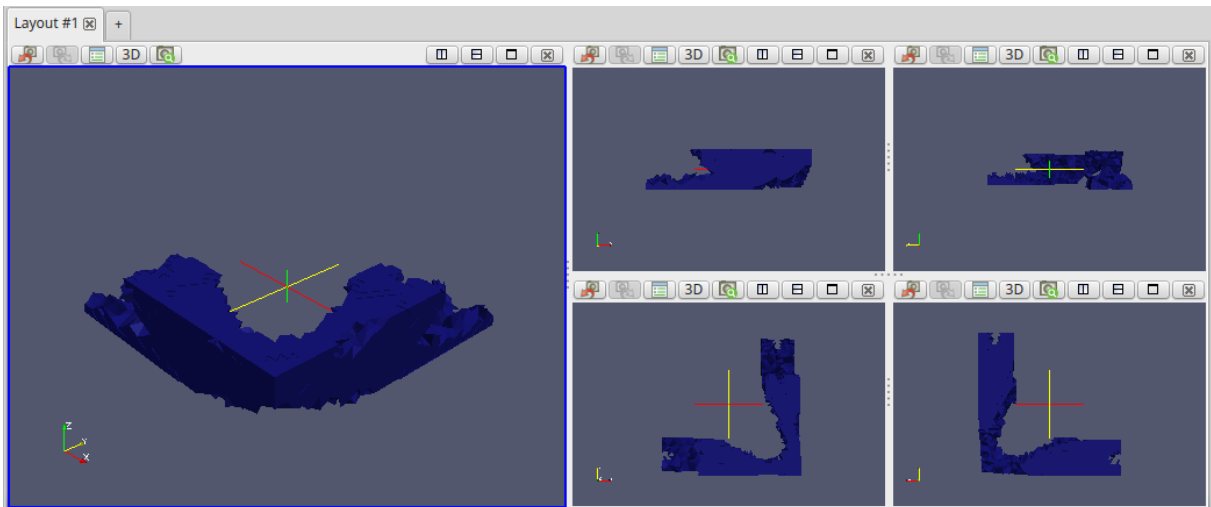


Figura 6.53 – Topologia final ótima não linear física e geométrica resultante do passo de otimização 19

As Fig. 6.54 a 6.56 mostram gráficos que descrevem o comportamento de cada um destes parâmetros ao longo dos processos iterativos das três análises.

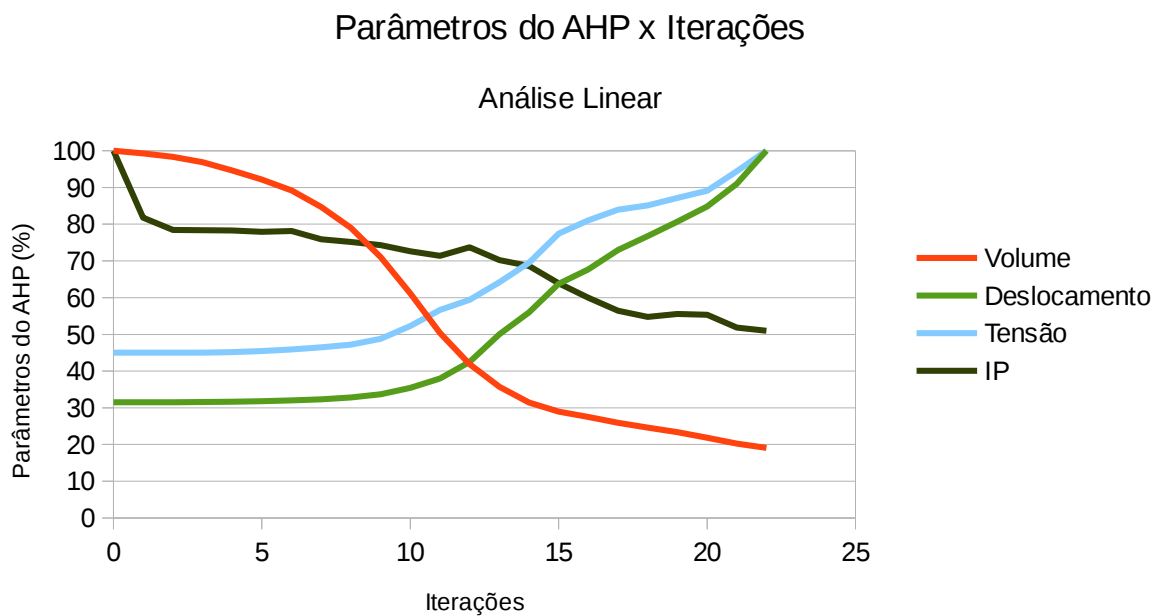


Figura 6.54 – Parâmetros do Vetor WSM (Volume, Deslocamentos, Tensão, Índice de Performance) x Iterações: Análise Linear

Na análise linear, o volume começa a decrescer lentamente no início do processo iterativo. A

partir da iteração 8, aproximadamente, a penalização de elementos se acentua até novamente ocorrer uma suavização ao fim da análise. Os deslocamentos e a tensão possuem comportamento semelhante, aumentando suavemente a partir da iteração 10. As respectivas variações de ambos atributos, deslocamento e tensão, são de aproximados 3,15 e 2,22 vezes, comparando o aumento entre o final e o inicial. O índice de performance apresenta uma significativa queda inicial, mas, a seguir, permanece sem grandes oscilações ao longo da análise. Neste modelo não há a formação de vazios nem a ruptura de membros que se formaram ao longo do processo iterativo e, por isso, não há picos de tensão, de acordo com a Fig. 6.54. A topologia ótima final é obtida na iteração 14, mostrada na Fig. 6.45.

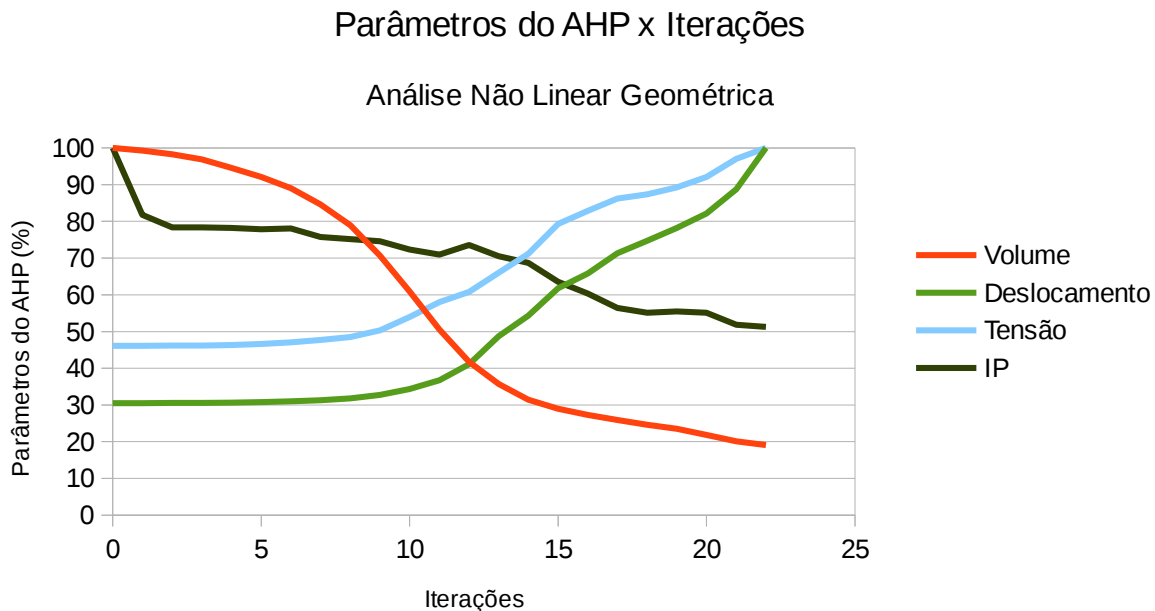


Figura 6.55 – Parâmetros do Vetor WSM (Volume, Deslocamentos, Tensão, Índice de Performance) x Iterações: Análise não Linear Geométrica

Na análise não linear geométrica, o comportamento de todos os atributos é muito similar ao visto na análise linear, exceto por uma sutil diferença nos valores da tensão e deslocamento principalmente no final do processo iterativo. De toda forma, a topologia ótima foi obtida na mesma iteração do modelo linear (iteração 14), mantendo o mesmo aspecto anterior. Novamente, não há a formação de vazios nem a ruptura de membros que se formaram ao longo do processo iterativo e, por isso, não há picos de tensão.

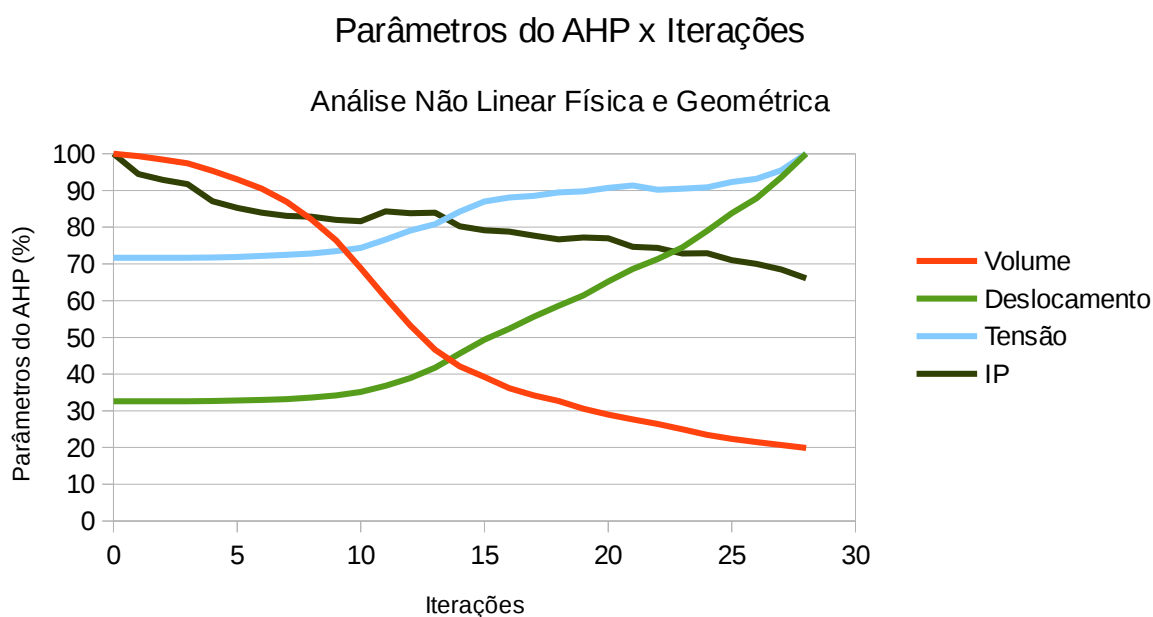


Figura 6.56 – Parâmetros do Vetor WSM (Volume, Deslocamentos, Tensão, Índice de Performance) x Iterações: Análise não Linear Física e Geométrica

Na análise não linear física e geométrica, o volume decresce de forma similar às análises anteriores. Há uma diferença nos valores do deslocamento e tensão, apesar do comportamento em si destes atributos ser similar ao das análises anteriores. As respectivas variações de ambos atributos, deslocamento e tensão, são de aproximados 3,1 e 1,4 vezes, comparando o aumento entre o final e o inicial. Como a topologia ótima possui uma aparência similar às outras análises, embora obtida em uma iteração diferente (iteração 19, mostrada na Fig. 6.53), igualmente para os modelos anteriores, não há a formação de vazios nem a ruptura de membros que se formaram ao longo do processo iterativo e, por isso, também não há picos de tensão.

Uma comparação entre as três análises é resumida na Fig. 6.57, mostrando os valores dos respectivos vetores WSM ao longo do processo iterativo. O ponto preto indica a iteração com o valor mínimo de cada análise, que representa a melhor solução para cada problema.

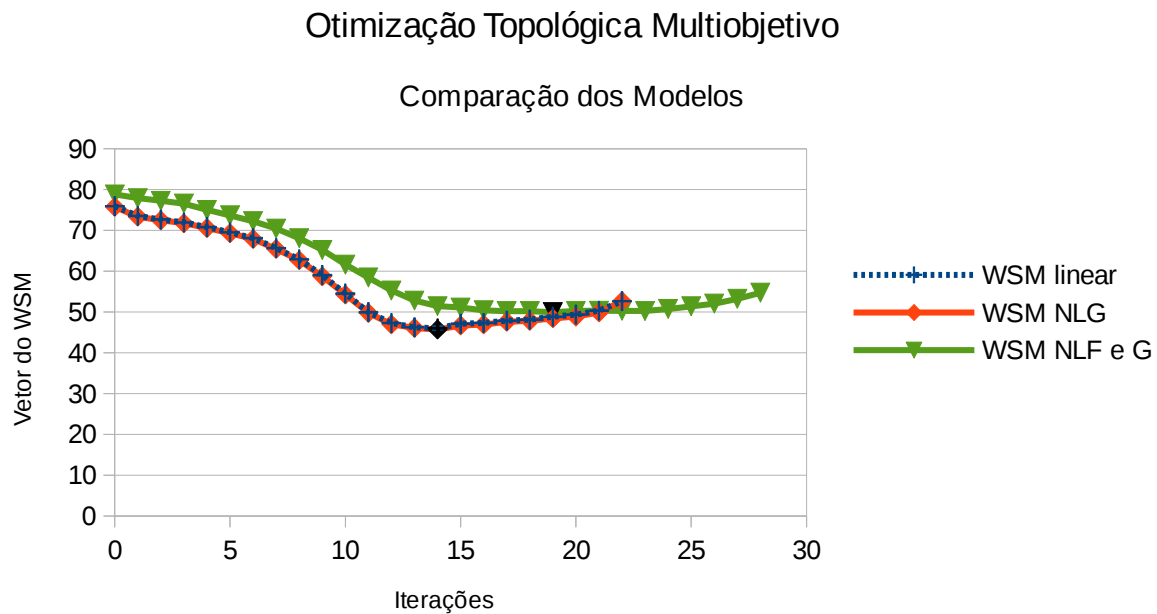


Figura 6.57 – Comparação dos Vetores do WSM x Iterações

O vetor WSM linear possui valor mínimo na iteração 14, configurando esta como a solução para o conjunto de possíveis soluções para a análise. Nesta iteração, o volume é de aproximados 31% em relação ao volume inicial, os deslocamentos cerca de 1,8 vezes o inicial e a tensão é da ordem de 1,55 vezes a tensão máxima de toda a análise. O índice de performance reside na faixa dos 69%.

O vetor WSM não linear geométrico possui valor mínimo na iteração 14, configurando esta como a solução para o conjunto de possíveis soluções para a análise. Nesta iteração, o volume é de aproximados 31% em relação ao volume inicial, os deslocamentos cerca de 1,8 vezes o inicial e a tensão é da ordem de 1,55 vezes a tensão máxima de toda a análise. O índice de performance reside na faixa dos 69%.

O vetor WSM não linear físico e geométrico possui valor mínimo na iteração 19, configurando esta como a solução para o conjunto de possíveis soluções para a análise. Nesta iteração, o volume é de aproximados 30,5% em relação ao volume inicial, os deslocamentos cerca de 1,9 vezes o inicial e a tensão é da ordem de 1,25 vezes a tensão máxima de toda a análise. O índice de performance reside na faixa dos 77%.

6.4 Discussão dos Resultados

Além das considerações apresentadas no capítulo anterior sobre os modelos otimizados a partir da análise linear, este capítulo procurou mostrar as diferenças e semelhanças entre as topologias ótimas resultantes dos tipos de análise estáticas.

Como não há uma garantia matemática de que o ótimo global vai ser obtido ao se usar a técnica SESO de otimização, uma investigação adequada da topologia resultante pode ser necessária ao se realizar apenas um tipo de análise. No primeiro exemplo, todas as topologias ótimas foram diferentes entre si e possuem características distintas, como o número de membros internos formados ao longo do processo iterativo. Apesar da topologia obtida para a análise não linear física e geométrica tender a ser a mesma da análise linear, quando os membros superiores são removidos, a tensão aumenta de tal forma que não há a convergência do modelo, encerrando a análise e garantindo uma diferença entre ambas topologias. Para este caso, ao se considerar pequenos deslocamentos e pequenas deformações, a topologia ótima obtida pela análise linear pode ser adequada para as condições impostas; mas, ao se considerar grandes deslocamentos e/ou grandes deformações, a resposta mais apropriada já seria outra.

No segundo exemplo, as topologias ótimas linear e não linear geométrica se mostraram semelhantes, exceto pelos valores de alguns parâmetros. A topologia correspondente à análise não linear geométrica manteve uma porcentagem maior de volume final do que a linear, mostrando-se ligeiramente mais rígida. Consequentemente, esta topologia apresenta valores menores para o deslocamento e tensão, embora não tão significativos, considerando a premissa que quanto menor o volume, maiores serão os deslocamentos e a tensão (fato verificado nas análises lineares do capítulo anterior). Em contrapartida, a topologia referente à análise não linear física e geométrica se mostrou diferente das demais, com um volume significativamente maior e a presença de um membro adicional na parte anterior que faz a ligação entre os braços superiores com o inferior (ambos modelos, linear e não linear geométrico, apresentaram um membro apenas).

Já no terceiro exemplo, todas as topologias ótimas foram semelhantes entre si, de acordo com

as condições iniciais impostas ao problema. A remoção (penalização) de elementos ao longo do processo iterativo para todas as análises foi praticamente a mesma, de acordo com a curva de volume mostrada nas Figs. 6.53, 6.54 e 6.55. Embora os valores tenham sutis diferenças entre si, decorrentes das aproximações do MEF, as topologias ainda assim foram parecidas. Considerando a magnitude da carga e as condições de contorno, a semelhança de todos os modelos é fundamentada no regime elástico linear do problema proposto.

Assim como pode ser notado nos exemplos lineares, há também para os não lineares a falta de simetria devido à dependência da malha. Porém, de acordo com o Exemplo 1, mesmo com esta falta de simetria, as topologias ótimas comparadas com a literatura apresentaram certo grau de semelhança, validando as diferenças entre os modelos lineares e não lineares.

Como demonstra este capítulo, é importante uma análise minuciosa de otimização ao se obter a topologia ótima. Para determinados modelos, apenas a realização de uma análise linear pode ser insuficiente, já que uma topologia inadequada pode ser obtida ao se considerar certas condições iniciais. Em outras palavras, se o material ultrapassa o regime linear elástico, por exemplo, a topologia obtida pela análise linear pode ser imprópria para a concepção da peça estrutural. Logo, faz-se necessária uma verificação das condições do material e da magnitude dos deslocamentos para realizar as devidas análises a fim de se obter a melhor topologia ótima. A otimização em regime não linear é especialmente importante para projetos que necessitem da análise do comportamento estrutural em estado limite último.

Finalmente, os exemplos estáticos não lineares apresentados neste capítulo mostraram topologias finais ótimas adequadas para cada problema imposto, sendo o primeiro exemplo comparado com os resultados obtidos na literatura. Isso, de certa forma, garante o bom funcionamento do algoritmo SESO implementado e sua interação com os programas necessários que foram descritos anteriormente.

7

CONSIDERAÇÕES FINAIS

De acordo com as análises realizadas na presente Tese, algumas considerações importantes podem ser feitas.

A importância de se otimizar está diretamente ligada ao fato de se aproveitar o máximo desempenho estrutural para determinadas condições iniciais e de contorno. Assim, de acordo com as especificações de projeto e do ponto de vista econômico, é vantajoso otimizar um modelo para se ter uma boa relação entre o volume final da estrutura (peso) e o tipo de carregamento aplicado. Além disso, outras variáveis, como custo, podem ser relativamente melhoradas ao fim do processo. Mais do que minimizar volume, é importante uma ponderação entre a tensão máxima atuante no modelo e o deslocamento máximo ao qual a nova topologia está submetida. De acordo com os exemplos, fica claro que quanto menor o volume da topologia, maiores serão as tensões e os deslocamentos impostos a ela. Assim, é necessário verificar quanto de volume pode ser removido sem comprometer a integridade e os limites de utilização, fixando a importância da otimização multiobjetivo deste trabalho;

De forma geral, as técnicas heurísticas de otimização necessitam de um critério de parada bem definido. Em muitos casos, esse critério de parada é o volume final, fazendo com que a topologia ótima seja a obtida na última iteração. Em outras palavras, o projetista informa qual o volume final da topologia ótima desejado, e o resultado terá aquele volume informado. O verdadeiro problema deste tipo de procedimento é a garantia de que o volume informado corresponde, de fato, a melhor topologia para o problema, ou seja, o ótimo. O método de otimização multiobjetivo utilizado neste trabalho contribui para a resolução desta questão,

uma vez que a topologia ótima não terá, obrigatoriamente, o volume previamente arbitrado pelo projetista. Convém lembrar que este volume informado ainda corresponde ao critério de parada, mas a ponderação do método de otimização multiobjetivo leva em conta outros parâmetros e faz com que o volume seja maior do que o mínimo (20% ou 10%) adotado para os exemplos aqui apresentados;

Uma das principais vantagens da técnica de otimização topológica SESO é a não exigência do cálculo de gradientes da função objetivo. Segundo (GOURNEY *et al*, 2008), de fato a função objetivo não está normalmente disponível para grande parte dos problemas práticos de otimização topológica, encontrados em aplicações reais de engenharia. Ainda, o cálculo do gradiente pode ser computacionalmente caro e para os casos em que a função objetivo apresenta descontinuidade ou funções com certo grau de complexidade, torna-se mais complicado de se obter uma resposta (DAS *et al*, 2011). Para estes casos, estudos que envolvem o cálculo de subgradientes podem ser uma alternativa adequada e viável para a resolução desse tipo de problema (VENCESLAU, 2013). De forma geral, o fato da teoria que envolve a técnica de otimização SESO, heurística, ser simplificada em comparação com outras técnicas determinísticas e/ou probabilísticas traz uma grande vantagem no que diz respeito à implementação computacional. Ainda, a possibilidade de inserção de filtros diversos para minimizar problemas relacionados à otimização junto ao MEF, como o tabuleiro de xadrez e dependência da malha, abre um atrativo campo de estudo que dispensa relações matemáticas complexas e que garante resultados satisfatórios;

A otimização multiobjetivo para este trabalho procurou garantir a topologia ótima de acordo com atributos preestabelecidos, mostrando que a topologia ótima nem sempre corresponde ao volume indicado no critério de parada. A grande vantagem do Processo Analítico Hierárquico (AHP) está na estruturação de problemas complexos com vários atributos de forma hierárquica ao mesmo tempo em que permite a resolução através da investigação de cada nível da hierarquia separadamente, combinando os resultados ao longo do progresso da análise. Estes níveis de hierarquia representam um sistema que se inicia do nível mais baixo (conjuntos de alternativas), passa por níveis intermédios (subcritérios e/ou critérios) quando for o caso, até chegar ao nível mais alto (objetivo geral). Com isso, as prioridades das alternativas são estimadas de forma independente para cada critério em cada nível para,

finalmente, o peso (ou prioridade) de cada critério ser definido (LIU e HAI, 2005). Convém também ressaltar que esse processo analítico baseado em uma metodologia de vários critérios de tomada de decisão permite medir fatores qualitativos e quantitativos em dado conjunto de atributos, cuja eficiência é obtida através da razão de consistência. Como para este trabalho este valor é inferior a 10% (8,9%), isto significa que os valores adotados para os atributos são adequados, garantindo a eficácia da resposta dos problemas multiobjetivos;

Com o MEF aplicado juntamente às técnicas de otimização foram gerados resultados significativamente satisfatórios. A literatura, em geral, abrange muito a utilização do MEF para otimização, apesar de também explorar, de forma mais recente, a utilização do Método dos Elementos de Contorno (MEC) para a otimização de modelos estruturais. Apesar de o MEF possibilitar o surgimento de alguns limitantes (como a dependência da malha, por exemplo), a utilização de filtros para a técnica SESO pode minimizar alguns desses problemas. Considerando-se que muitas das topologias ótimas extraídas da literatura, a fim de serem comparadas com os resultados deste trabalho, foram obtidas por diferentes técnicas de otimização, a técnica SESO mostrou notável eficiência de acordo com o nível de aproximação das topologias ótimas. Desta forma, vale a pena uma mais ampla e melhor investigação e aprofundamento da técnica, inclusive com outros métodos matemáticos (como Elementos de Contorno), com o intuito de melhorar o SESO, com a inserção de outros tipos de filtros, e obter resultados cada vez mais aprimorados;

Diferentes condições iniciais e de contorno considerando um mesmo modelo estrutural inicial (mesma dimensão da peça) geram topologias finais ótimas diferentes. Os maiores valores de tensão em uma análise inicial se concentram, pode-se dizer, na menor distância entre a carga e o apoio. Isto implica dizer que tanto os elementos nas proximidades do carregamento quanto os elementos na região da restrição devem estar presentes na topologia ótima final. Se, ao longo do processo iterativo (geralmente no fim), essa ligação entre carga e apoio for rompida (quando há remoção de membros internos, por exemplo), a estabilidade estrutural pode ficar comprometida. Para a avaliação da estabilidade, alguns autores sugerem a análise de robustez da solução otimizada obtida que, segundo Dill (2013), está diretamente associada à garantia de estabilidade. Zhao *et al.* 2015 mencionam que a otimização topológica robusta minimiza a sensibilidade a incertezas, que são condições não consideradas no projeto inicial e importantes

para o desempenho estrutural. Tudo isso respeitando os Estados Limites aos quais a peça estrutural venha a ser projetada. Embora esta Tese não tenha trabalhado diretamente neste quesito, é um importante tema a ser estudado em trabalhos futuros, no âmbito da otimização topológica multiobjetivo.

Como o ótimo global nem sempre pode ser obtido em uma análise utilizando a técnica heurística SESO, a realização de uma análise não linear, além da linear, pode ser adequada a fim de investigar possíveis diferentes soluções, face ao problema proposto. Em outras palavras, quando a carga considerada na análise de um modelo estrutural qualquer sob certas condições de contorno é pequena o suficiente para que não provoque o escoamento do material e/ou grandes deslocamentos, apenas uma análise linear pode ser suficiente para a determinação da topologia ótima, pois as análises não lineares resultarão em topologias ótimas similares a linear. Do contrário, uma falsa topologia, ou uma topologia inadequada, ou ainda uma topologia não totalmente otimizada pode ser gerada a partir de uma análise linear para problemas desse gênero, uma vez que as respostas podem ser diferentes para cada tipo de análise.

Finalmente, a partir destas considerações, é visto que vale a pena otimizar uma estrutura para se poder utilizar o máximo da capacidade estrutural de acordo com as condições iniciais impostas ao modelo. Além da estrutura, outros parâmetros também podem ser otimizados, como peso e custo final. Convém ressaltar que a topologia ótima obtida pela técnica SESO não deve ser imediatamente utilizada para a produção, ou seja, é necessário um tratamento no que diz respeito à forma da topologia e padrões de reforços estruturais, incluindo as simetrias e espessuras dos membros formados. Logo, uma nova otimização pode ser feita sendo, neste caso, a otimização de forma e/ou paramétrica. E, novamente, uma análise não linear é importante para que não se obtenha uma topologia ótima possivelmente inadequada em uma análise mais detalhada de uma estrutura, já que a distribuição de tensão no modelo pode variar para os casos onde o material se comporta de maneira não linear, ou quando grandes deslocamentos e giros estão presentes na análise estrutural.

BIBLIOGRAFIA

- ALLAIRE, G.; JOUVE, F.; TOADER, A. M.; *Structural Optimization Using Sensitivity Analysis And A Level-Set Method*. Journal of Computational Physics, 194(1), 363–393. 2004.
- ALLAIRE, G.; DE GOURNAY, F.; JOUVE, F.; TOADER, A. M. *Structural Optimization Using Topological And Shape Sensitivity Via A Level Set Method*. Control and Cybernetics 34(1), 59–80. 2005.
- ALMEIDA, V. S. *Otimização Topológica De Estruturas via MEF e MEC*. Projeto de Pesquisa PQ, CNPq. 2009.
- ALMEIDA, V. S.; FERNANDES, W. S.; SIMONETTI, H. L.; NEVES, F. A., GRECO, M. *Avaliação de Diferentes Elementos Finitos em Problemas Elásticos 2D em Otimização Topológica*. CILAMCE, Brasil. 2011.
- ALMEIDA, V. S.; SIMONETTI, H. L.; NEVES, F. A. *Seleção de Topologias Ótimas de Estruturas Elásticas 2D com Restrição de Tensão – Via Smooth Evolutionary Structural Optimization*. Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería. 30(2):69–76. 2014.
- ADELI, H.; SARMA, K. C. *Cost Optimization of Structures*. Chichester, UK. 2006.
- ANDERSON, T. W. *Estimating linear statistical relationships*. The Annals of Statistics, vol. 12, no. 1, pp. 1-45. 1984.
- ARORA, J. S. *Introduction to Optimum Design, 3rd Edition*. Elsevier, Oxford, USA. 2012.
- BABY, S. *AHP modeling for multicriteria decision-making and to optimise strategies for protecting coastal landscape resources*. IJIIMT; 4: 218–227. 2013.
- BATHE, K. *Finite element procedures*. Prentice-Hall, Inc. New Jersey, USA. 1996.
- BEINAT, E.; NIJKAMP P. *Multicriteria analysis for land–use management*. Dordrecht:

Kluwer Academic Publishers. 1998.

BELYTSCHKO, T.; XIAO, S. P.; PARIMI, C. *Topology Optimization With Implicit Functions And Regularization*. International Journal for Numerical Methods in Engineering 57(8), 1177–1196. 2003.

BENDSØE, M. P. *Optimization Of Structural Topology, Shape And Material*. Berlin, Heidelberg, New York: Springer. 1995.

BENDSØE, M. P. *Optimal Shape Design as a Material Distribution Problem*. Structural Optimization, vol. 1, pp. 193-202. 1989.

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

BENDSØE, M.; SIGMUND, O. *Topology Optimization: Theory, Methods And Applications*. Springer. New York, USA. 2003.

BOUCHITTÉ, G.; GANGBO, W.; SEPPECHER, P. *Michell Trusses And Lines Of Principal Action*. Mathematical Models and Methods in Applied Sciences, vol. 18, No. 9, 1571–1603. 2008.

CERVERA, E. *Evolutionary Structural Optimization based on Boundary Representation of B-spline Geometry*. PhD Thesis. University of Durham, Durham, UK. 2003.

CERVERA, E.; TREVELYAN, J. *Evolutionary Structural Optimization based on Boundary Representation of NURBS. Part I: 2D Algorithms*. Computers and Structures, 83(23-24), 1902–1916. 2005.

CHABOCHE, J. L. *A Review of Some Plasticity and Viscoplasticity Constitutive Theories*. International Journal of Plasticity, vol 24 (10): 1642–1693. 2008.

CHANG, D. H.; YOO, K. S.; PARK, J. Y.; HAN, S. Y. *Optimum Design for Nonlinear Problems Using Modified Ant Colony Optimization*. International Conference on Software and Computer Applications (ICSCA), Singapore. 2012.

CHENG, T. K.; OLHOFF, N. *An Investigation Concerning Optimal Design of Solid Elastic Plates*. International Journal of Solids Structures, v.17, pp. 305-323. 1981.

- CHENG, T. K.; OLHOFF, N. *Regularized Formulation for Optimal Design of Axisymmetric Plates*. International Journal of Solids Structures, 18-2, pp. 153-169. 1982.
- CHU, D. N.; XIE, Y. M.; HIRA, A.; STEVEN, G. P. *Evolutionary Structural Optimization for Problems with Stiffness Constraints*. Finite Elements in Analysis and Design, Vol. 21, pp. 239-251. 1996.
- COX, H. L. *The Theory of Design*, Aeronaut. Res. Council Rep. No. 19791. 1958.
- CRISFIELD, M. A. *Non Linear Finite Element Analysis of Solids and Structures*. John Wiley & Sons, Inc., 605 Third Avenue, New York, USA. 1997.
- DAS, R.; JONES, R.; XIE, Y. M. *Optimal Topology Design of Industrial Structures Using an Evolutionary Algorithm*. Optimization and Engineering, v. 12, p. 681-717. 2011.
- DÍAZ, A. R.; SIGMUND, O. *Checkerboard patterns in layout optimization*. Structural and Multidisciplinary Optimization, 10, 40-45. 1995.
- DILL, G. K. *Projeto de Controladores Robustos para Sistemas de Potencia Baseado em Otimização Paramétrica*. Tese de Doutorado do Programa de Pós Graduação em Engenharia Elétrica da Universidade Federal de Santa Catarina. 2013.
- DOWNEY, A. *Think Python. How to Think Like a Computer Scientist*. Green Tea Press. Needham, Massachusetts, USA. 2012.
- DUTTA D. *Climate Perturbation and Coastal Zone Systems in Asia Pacific Region: Holistic Approaches and Tools for Vulnerability Assessment and Sustainable Management Strategy*. Final Report submitted to Asia-Pacific Network for Global Change Research: 52. 2007.
- FERNANDES, W. S. *Estudo de otimização topológica em estruturas 2D considerando a não linearidade geométrica*. Dissertação de Mestrado do Programa de Pós Graduação do Departamento de Engenharia Civil da UFOP, Ouro Preto. 2013.
- FERNANDES, W. S.; ALMEIDA, V. S.; NEVES, F. A.; GRECO, M. *Topology optimization applied to 2D elasticity problems considering the geometrical nonlinearity*. Engineering Structures 100, p. 116–127. 2015.
- FRIEDMAN, J. H.; STUETZLE, W. *Projection Pursuit Regression*. J. Amer. Statist. Assoc. 76, 817. 1981.

- GEA, H.; LUO, J. *Topology optimization of structures with geometrical nonlinearities*. Computers & Structures, 79, 1977–1985. 2001.
- GUEDES, J. M.; TAYLOR, J. E. *On The Prediction Of Material Properties And Topology For Optimal Continuum Structures*. Structural and Multidisciplinary Optimization 14, 183-192. 1997.
- GUEST, J. K.; PRÉVOST, J. H. *Optimizing Multifunctional Materials: Design of Microstructures for Maximized Stiffness and Fluid Permeability*. International Journal of Solids and Structures, vol. 43, pp. 7028–7047. 2006.
- GOLDBERG, D. E. *Genetic Algorithms In Search, Optimization And Machine Learning*. Addison-Wesley Publishing, Reading, MA. 1989.
- GOMES, F.; SENNE, T. *An algorithm for the topology optimization of geometrically nonlinear structures*. International Journal for Numerical Methods in Engineering. Volume 99, Issue 6, pages 391–409. 2013.
- GOODMAN, J.; KOHN, R. V.; REYNA, L. *Numerical Study of Relaxed Variational Problem from Optimal Design*, Computer Methods in Applied Mechanics and Engineering, 57, pp. 107-127. 1986.
- GOURNAY, F.; ALLAIRE, G.; JOUVE, F. *Shape and Topology Optimization of the Robust Compliance via the Level Set Method*. ESAIM: Control, Optimisation and Calculus of Variations, v. 14, 2008, p. 43–70. 2008.
- HABER, R. B.; BENDSØE, M. P.; JOG, C. *A New Approach To Variable-Topology Shape Design Using A Constraint On The Perimeter*. Structural and Multidisciplinary Optimization 11, 1-12. 1996.
- HAFTKA, R.; GRANDHI, R. *Structural Shape Optimization – A survey*, Computer Methods in Applied Mechanics and Engineering, v. 57, pp. 91-106. 1986.
- HAFTKA, R. T.; GÜRDAL, Z. *Elements of structural optimization, 3rd Edition*. Kluwer Academic Publishers. Dordrecht, The Netherlands. 1992.
- HÄRDLE, W.; STOKER, T. *Investigating smooth multiple regression by the method of average derivatives*. Journ. of the american statist. association (jasa), vol. 84, no.408, pp.986-995. 1989.

- HASTIE, T.; TIBSHIRANI, R. *Generalized Linear Models (with Discussion)*. Statistical Science, vol. 1, no. 3, pp. 297-318. 1986.
- HECKMAN, N. *Spline Smoothing in Partly Linear Models*. Journal of the Royal Statistical Society B, vol. 48, no. 2, pp. 244-248. 1986.
- HEMP, W. S. *Optimum Structures*, Oxford: Claredon , p. 123. 1973.
- HIBBELER, R. C. *Resistência dos Materiais, 7ª Edição*. Prentice Hall, São Paulo. 2010.
- HU, Z.; RAUCH, E. F.; TEODOSIU, C. (1992). *Work-Hardening Behavior of Mild Steel under Stress Reversal at Large Strains*. International Journal of Plasticity, v.8, p. 839-859.
- HUANG, X.; XIE, M. *Evolutionary Topology Optimization of Continuum Structures: Methods and Applications*. United Kingdom: Wiley. 2010.
- IZMAILOV, A., SOLODOV, M. *Otimização Volume 2*. 1 ed. Rio de Janeiro, IMPA. 2007.
- IVANČO, V. *Nonlinear Finite Element Analysis : Script Of Lectures*. PhD Thesis of Faculty of Mechanical Engineering, Technical University of Košice, Slovakia. 2011.
- 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, 130, 203-226. 1996.
- JUNG, D.; GEA, C. *Topology Optimization of Nonlinear Structures*, Finite Elements in Analysis and Design, v. 40, p. 1417–1427. 2004.
- KAVEH, A.; HASSANI, B.; SHOJAEI, S.; TAVAKKOLI, S. M. *Structural topology optimization using ant colony methodology*. Engineering Structures, 30(9), 2559–2565. 2008.
- KIKUCHI, N.; ODEN, J. T.; SONG, Y. J. *Convergence of Modified Penalty Methods and Smoothing Schemes of Pressure for Stoke's Flow Problems*. In Gallagher, R. et al. (eds), "Finite Elements in Fluids", vol. 5, John Wiley & Sons, pp. 107-126. 1984.
- KOHN, R. V.; STRANG, G. *Optimal-design and relaxation of variational problems*. Communications on Pure and Applied Mathematics, v.39, n.1, pp.112-137. 1986.
- KROG, L.; OLHOFF, N. *Optimum topology and reinforcement design of disk and plate*

structures with multiple stiffness and eigenfrequency objectives. Computers & Structures, 72, 535-563. 1999.

LABANOWSKI, A. J.; FANCELLO, E. A.; NOVOTNY, A. A. *Simp, Eso e Tsa: Uma Análise Comparativa de Métodos de Otimização Topológica para Elasticidade 2D e 3D*, CILAMCE, Brasil. 2004.

LAHUERTA, R. D.; ROJAS, P. A. M.; SILVA, E. C. N. *Projeto de Estruturas em Regime Não-Linear Geométrico Utilizando o Método da Otimização Topológica*. Asociación Argentina de Mecánica Computacional (AMCA), Argentina. 2010.

LAI, T. L.; ROBBINS, H.; WEI, C. Z. *Strong Consistency of Least Squares Estimates in Multiple Regression*. Proc. Natl. Acad. Sci. USA, vol. 75, No. 7, pp. 3034-3036. 1978.

LANCZOS, C. *The variational principles of mechanics*. University of Toronto, Canada. 1952.

LANES, R.; GRECO, M. *Aplicação De Um Método De Otimização Topológica Evolucionária Desenvolvido Em Script Python*. Ciência & Engenharia, 22(1), 1–11. 2013.

LEVINE, I. N., *Quantum Chemistry*. Ed. 5. New Jersey: Prentice Hall, 739 p. 2000.

LI, K. C. *Sliced Inverse Regression for Dimension Reduction*. Journal of the American Statistical Association (JASA), vol. 86, no. 414, pp. 316-342. 1991.

LI, Q.; STEVEN, G. P.; XIE, Y. M. *Optimal Shape Design for Steady Heat Conduction by the Evolutionary Procedure*. Inverse Problems in Heat Transfer and Fluid Flow, Heat Transfer Div. American Society of Mechanical Engineers, New York, 340, 159-164. 1997.

LI, R.; ZHANG, Y. TONG, L. *Numerical Study of the Cyclic Load Behavior of AISI 316L Stainless Steel Shear Links for Seismic Fuse Device*. Frontiers of Structural and Civil Engineering, vol 8(4): 414-426. 2014.

LIANG, Q. Q. *Performance-Based Optimization Of Structures: Theory And Applications*. Spon Press, London. 2005.

LIANG, Q. Q.; XIE, Y. M.; STEVEN, G. P. *Topology Optimization Of Strut-And-Tie Models In Reinforced Concrete Structures Using An Evolutionary Procedure*. ACI Structural Journal, Vol. 97, No. 2, pp. 322-330. 2000.

- LIN, C. Y.; HSU, F. M. *An Efficient Algorithm to obtain the optimal topology of industrial structural design problems*. Journal of the Chinese institute of Engineers, vol. 31, n.1, pp. 17-30. 2008.
- LIU, F. F.; HAI, H. L. *The Voting Analytic Hierarchy Process Method for Selecting Supplier*. Int. J. Production Economics 97, p. 308–317. 2005.
- LOPES, W. *Estudo do Encruamento dos aços AISI 304 E AISI 409 com uso da Técnica de Cisalhamento Planar Simples*. Tese de Doutorado da Escola de Engenharia Metalúrgica da Universidade Federal de Minas Gerais. 2009.
- LURIE, K. A.; FEDEROV, A. V.; CHERKAEV, A. V. *Regularization of Optimal Design Problems for Bar and Plates, Parts I e II*, Journal of optimizations and Theory and Applications, 37-4, pp. 499-521,523-543. 1982.
- MANICKARAJAH, D.; XIE, Y. M.; STEVEN, G. P. *An evolutionary method for optimization of plate buckling resistance*. Finite Elements in Analysis and Design, v. 29, pp. 205-230. 1998.
- MARCZAK, R. J. *Optimal Topology Identification of 2D Elastic Structures with the Boundary Element Method*. Mecánica Computacional, Argentina, Vol XXV, pp. 2769-2784. 2006.
- MARCZAK, R. J. *Optimization of Elastic Structures Using Boundary Elements and a Topological-Shape Sensitivity Formulation*. Latin American Journal of Solids and Structures, 5, 99–117. 2008.
- MARLER, R. T.; ARORA, J. S. *Survey of Multiobjective Optimization Methods for Engineering*. Struct Multidiscip O; 26: 369–395. 2004.
- MOAVENI, S. *Finite Element Analysis: Theory and Application with ANSYS*. Prentice-Hall, Inc. New Jersey, USA. 2003.
- ODU, G. O.; CHARLES-OWABA, O. E. *Review of Multi-criteria Optimization Methods – Theory and Applications*. IOSR Journal of Engineering (IOSRJEN). e-ISSN: 2250-3021, p-ISSN: 2278-8719 Vol. 3, Issue 10 ||V2|| PP 01-14. 2013.
- OLIVEIRA, R. S. *Análise de Pavimentos de Edifícios de Concreto Armado com a Consideração da Não-Linearidade Física – Modelagem e Metodologia de Aplicação a*

- Projetos*. Tese de Doutorado da Escola de Engenharia da Universidade de São Paulo, São Carlos. 2001.
- OSHER, S.; SETHIAN, J. *Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations*. Journal of Computational Physics, 12–49. 1988.
- PARKINSON, M. B.; HOWELL, L. L.; COX, J. J. *A parametric Approach to the Optimization-Based Design of Compliant Mechanisms*. Proceedings of the 23rd Design Automation Conference, DETC97/DAC-3763. 1997.
- PEREIRA, C. E. L. *Análise de Sensibilidade Topológica em Problemas de Não-Linearidade Geométrica e Hiperelasticidade Não-Linear Quasi-Incompressível*. Tese de Doutorado da Faculdade de Engenharia Mecânica da Universidade Estadual de Campinas, Campinas. 2006.
- PIMENTA, P. M. *Fundamentos da Mecânica dos Sólidos e das Estruturas*. Departamento de Estruturas e Fundações da Escola Politécnica da Universidade de São Paulo, São Paulo. 2006.
- PRAGER, W. *A Note On Discretized Michell Structures*. Comput. Meth. Appl. Mech., v. 3,p. 349–355. 1974.
- PROENÇA, S. P. B. *Sobre modelos matemáticos do comportamento não linear do concreto: análise crítica e contribuições*. São Carlos. Tese (Doutorado) - Departamento de Engenharia de Estruturas, EESC-USP. 1988.
- PROOS, K. *Evolutionary Structural Optimisation as a Robust and Reliable Design Tool*. PhD Thesis of Philosophy. School of Aeronautical, Mechatronic and Mechanical Engineering The University of Sydney, Australia. 2006.
- QUERIN, O. M.; STEVEN, G. P.; XIE, Y. M. *Evolutionary structural optimisation (ESO) using a bidirectional algorithm*. Engineering Computations, Vol. 15 No. 8, pp. 1031-1048. 1998.
- QUERIN, O. M.; STEVEN, G. P.; XIE, Y. M. *Evolutionary Structural Optimisation Using An Additive Algorithm*. Finite Elements in Analysis and Design 34, 291–308. 2000.
- RAO, S. S. *Engineering Optimization: Theory and Practice*. 4th ed. John Wiley & Sons, Inc., Hoboken, New Jersey. 2009.

- RAUCH, E. F. *Plasticity of Metals During Cold Working*. Multiscale Phenomena in Plasticity. In: J. Lépinoux et al. (eds.), *Multiscale Phenomena in Plasticity*, Kluwer Academic Publishers, p. 303–318. 2000.
- RAYMOND, J. S. *A Generalization of Lax-Milgram's Theorem*. *Le Matematiche*, vol. LII Fasc. I, pp. 147–155. 1997.
- ROBINSON, P. M. *Root N-Consistent Semiparametric Regression*. *Econometrica*, vol. 56, no. 4, pp. 931-954. 1988.
- ROZVANY, G. I. N. *Aims, Scope, Methods, History And Unified Terminology Of Computer-Aided Topology Optimization In Structural Mechanics*. *Structural and Multidisciplinary Optimization*, 21(2), 90–108. 2001.
- ROZVANY, G. I. N.; BENDSØE, M. P.; KIRSCH, U. *Layout Optimization of Structures*, *Applied Mechanics Review*, v. 48, pp. 41-119. 1995.
- ROZVANY, G.; OLHOFF, N. *Topology optimization of structures and composite continua*. Kluwer Academic Publishers. Dordrecht, The Netherlands. 2000.
- ROZVANY, G. I. N.; OLHOFF, N.; CHENG, K.; TAYLOR, J. E. *On the Solid Plate Paradox in Structural Optimization*. *Journal of Structural Mechanics*, 10-1, pp. 1- 32. 1982.
- SAATY T. L. *The Analytic Hierarchy Process*. New York: McGraw-Hill. 1980.
- SCHNEIDER, D.; ERNEY, T. *Combination of Topology and Topography Optimization for Sheet Metal Structures*. Altair. Acessado em <ftp://ftp.altair.de/pub/edu/protected/free_papers/OptiCON2000.pdf>. 2000.
- SEGERLIND, L. *Applied Finite Element Analysis, 2nd Edition*. John Wiley & Sons, Inc., Canada. 1984.
- SEHLSTRÖM, A. *Multi-Objective Topology Optimization: Tracing of Pareto-Optimal Structures with Respect to Volume, Compliance and Fundamental Eigenvalue*. Master's thesis of Department of Applied Mechanics Division of Dynamics and Division of Material and Computational Mechanics. CHALMERS UNIVERSITY OF TECHNOLOGY. Gothenburg, Sweden. 2013.
- SETHIAN, J. A.; WIEGMANN, A. *Structural Boundary Design Via Level Set And Immersed Interface Methods*. *Journal of Computational Physics*, 163(2), 489–528. 2000.

- SIGMUND, O. *Design of material structures using topology optimization*. Ph.D. Thesis, Department of Solid Mechanics, Technical University of Denmark. 1994.
- SIGMUND, O. *On The Design Of Compliant Mechanisms Using Topology Optimization*. *Mech.* Springer. Structural and Multidisciplinary Optimization. 25, 495-526. 1997.
- SIGMUND, O.; PETERSSON, J. *Numerical Instabilities In Topology Optimization: A Survey On Procedures Dealing With Checkerboards, Mesh-Dependencies And Local Minima*. *Structural Optimization*, (2), 68–75. 1998.
- SILVA, E. C. N. *Técnicas de Otimização Aplicadas no Projeto de Peças Mecânicas*. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos, Escola Polit. da USP, SP, Brasil. Acessado em <<http://sites.poli.usp.br/d/pmr5215/otimizacao.pdf>>. 2001.
- SIMONETTI, H. L. *Otimização Topológica de Estruturas Elásticas 2D via MEC Baseada em uma Nova Abordagem Evolucionária*. Tese de Doutorado do Programa de Pós Graduação do Departamento de Engenharia Civil da UFOP, Ouro Preto. 2016.
- SONMEZ, F. *Structural Optimization Using Simulated Annealing*. *Simulated Annealing*. I-Tech Education and Publishing, Vienna, Austria. 2008.
- SYSWERDA, G. *Uniform Crossover In Genetic Algorithms*. Proceedings of 3rd International Conference on Genetic Algorithms, George Mason University, Morgan Kaufmann Publishers, Inc., pp. 2–9. 1989.
- THUILLIER, S.; MANACH, P. Y.; MENEZES, L. F. *Occurrence of strain path changes in a two-stage deep drawing process*. *Journal of Materials Processing Technology*, v. 210, p. 226-232. 2010.
- VANDERPLAATS, G. N. *Numerical Optimization Techniques for Engineering Design with Applications*. Boston: McGraw-Hill Book Company. 1984.
- VENCESLAU, H. M. *Subgradiente Diferenciável via Suavização Hiperbólica*. Tese de Doutorado do Programa de Pós-Graduação Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, UFRJ. Rio de Janeiro. 2013.
- WANG, Y. *A Study on Microstructures of Homogenization for Topology Optimization*. Thesis Doctor, Victoria University of Technology Melbourne, Australia. 2003.
- WANG, M. Y.; WANG, X., GUO, D. *A Level Set Method For Structural Topology*

- Optimization*. Computer Methods in Applied Mechanics and Engineering, 192(1–2), 227–246. 2003.
- WU, H. C. *Anisotropic Plasticity for Sheet Metals using the Concept of Combined Isotropic-Kinematic Hardening*. International Journal of Plasticity, v. 18, p. 1661-1682. 2002.
- XIE, Y.; STEVEN, G. *A Simple Evolutionary Procedure For Structural Optimization*. Computers & Structures, Vol. 49, No. 5, pp. 885-896. 1993.
- XIE, Y. M.; STEVEN, G. P. *Evolutionary Structural Optimization*. Springer-Verlag, Berlin. 1997.
- YAMADA, T.; MANABE, M.; IZUI, K.; NISHIWAKI, S. *A Topology Optimization Method for Geometrically Nonlinear Problems Incorporating Level Set Boundary Expressions and a Particle Method*. Journal of Advanced Mechanical Design, Systems, and Manufacturing, 7(4), 630–643. 2013.
- YAN, W.; GAO, Z. Shape optimization in the Navier-Stokes flow with thermal effects. *Numerical Methods for Partial Differential Equations*, 1–16. Wiley Periodicals, Inc. 2013.
- YULIN, M.; XIAOMING, W. *A level set method for structural topology optimization and its applications*. Advances in Engineering Software 35(7), 415– 441. 2004.
- ZADEH, L. A. *Optimality and non-scalar-valued performance criteria*. IEEE Trans Autom Control AC-8; 59–60. 1963.
- ZHAO, C.; STEVEN, G. P.; XIE, Y. M. *Evolutionary Natural Frequency Optimization Of Thin Plate Bending Vibration Problems*. Structural Optimization, 11, pp 244-251. 1996.
- ZHAO, Q.; CHEN, X.; MA, Z.; LIN, Y. *Robust Topology Optimization Based on Stochastic Collocation Methods under Loading Uncertainties*. Mathematical Problems in Engineering, vol 2015, 14 p. 2015.
- ZHOU, M.; PAGALDIPTI, N.; THOMAS, H. L.; SHYY, Y. K. *An integrated approach to topology, sizing, and shape optimization*. Structural and Multidisciplinary Optimization, Springer-Verlag, v. 26, p. 308–317. 2004.
- ZHOU, M.; ROZVANY, G. I. N. *The COC algorithm, part II: topological, geometrical and generalized shape optimization*. Computer Methods in Applied Mechanics and

Engineering, v.89, pp.309-336. 1991.

ZIONTS, S. *Multiple criteria mathematical programming: an updated overview and several approaches*. In: Mitra G, editor. *Mathematical models for decision support*, Berlin: Springer-Verlag; p. 135–167. 1988.

SITES ACESSADOS:

Salome-Meca: *software* livre. Versão 2014.1. Belo Horizonte, UFMG, 2014. Disponível em: <<http://www.code-aster.org/V2/spip.php?article303>>. Acesso em 10 fev. 2014.

Free Software Foundation, Inc. "What is GNU". Disponível em: <<http://www.gnu.org/>>. Acesso em 29 jan. 2014.

Silicon Graphics International Corp. "Open Inventor, Open Source". Disponível em: <<http://oss.sgi.com/projects/inventor/>>. Acesso em 17 de mar. 2014.

EPX Europlexus Software. "What is EPX". Disponível em: <<http://www-epx.cea.fr/index.php/what-is-epx>>. Acesso em 22 de mar. 2014.

Open Publication License. "Open Content". Disponível em: <<http://opencontent.org/openpub/>>. Acesso em 22 de mar. 2014.

Software Foundation, Inc. "GNU Lesser General Public License, version 2.1". Disponível em: <<https://www.gnu.org/licenses/old-licenses/lgpl-2.1.html>>. Acesso em 22 de mar. 2014.

Lazarus and Free Pascal Team. "About Lazarus Project". Disponível em: <<http://www.lazarus.freepascal.org/index.php?page=about>>. Acesso em 22 de mar. 2014.

Eficas and ASTK Internationalization. "An effort to bring multilingual support to the Code-Aster interfaces". Disponível em: <<http://codeaster-intl.sourceforge.net/index.htm>>. Acesso em 22 de mar. 2014.

The Python Software Foundation. "The Python Tutorial". Disponível em: <<http://docs.python.org/2/tutorial/index.html>>. Acesso em 23 de mar. 2014.

Structural Optimization using OptiStruct. Altair Engineering, Inc. Disponível em: <<http://www.altairhyperworks.com/>>. Acesso em 16 de ago. 2014.

APÊNDICE A

TABELAS DE VALORES DOS PARÂMETROS E DOS VETORES WSM RELATIVAS AO CAPÍTULO 6 – RESULTADOS NÃO LINEARES

Tabela A.1 – Valores dos parâmetros e do Vetor WSM (Exemplo 6.1, linear)

Iteração	Volume	Deslocamento (cm, %)	Tensão (MPa, %)	IP	WSM		
0	100,00	1,25E-04	29,17	234,69	64,83	100,00	77,93
1	95,83	1,25E-04	29,18	234,71	64,83	96,40	77,07
2	90,47	1,25E-04	29,24	234,76	64,85	95,72	76,61
3	85,31	1,26E-04	29,39	234,79	64,85	91,24	75,68
4	80,41	1,27E-04	29,60	234,60	64,80	90,05	75,13
5	75,82	1,29E-04	30,08	235,62	65,08	87,80	74,30
6	71,65	1,31E-04	30,43	235,50	65,05	84,51	73,24
7	67,49	1,33E-04	31,06	235,77	65,13	84,50	72,64
8	63,44	1,37E-04	31,86	237,30	65,55	78,39	71,09
9	59,41	1,40E-04	32,65	237,36	65,56	75,90	69,10
10	55,07	1,44E-04	33,67	236,19	65,24	74,59	66,96
11	50,70	1,49E-04	34,83	236,02	65,20	73,37	65,32
12	47,03	1,60E-04	37,26	239,84	66,25	72,51	64,75
13	43,95	1,69E-04	39,49	242,85	67,08	72,22	64,29
14	41,44	1,89E-04	44,09	250,18	69,10	72,45	64,72
15	39,16	2,00E-04	46,69	254,18	70,21	72,03	64,36
16	36,97	2,09E-04	48,76	257,93	71,25	72,15	63,77
17	34,78	2,28E-04	53,21	269,88	74,55	70,29	63,48
18	32,72	2,40E-04	56,02	276,82	76,46	69,97	62,95
19	30,51	2,59E-04	60,29	286,62	79,17	69,24	62,82
20	28,29	2,74E-04	63,75	284,40	78,56	69,41	62,00
21	26,92	2,79E-04	65,08	282,78	78,11	70,89	61,05
22	25,56	2,99E-04	69,71	298,45	82,44	68,54	61,11
23	24,12	3,13E-04	72,95	320,34	88,49	65,30	61,11
24	22,74	3,39E-04	79,10	338,24	93,43	62,41	61,74
25	21,65	3,63E-04	84,71	353,14	97,55	60,96	62,72
26	20,39	3,95E-04	91,96	357,75	98,82	61,16	64,16
27	19,31	4,29E-04	100,00	362,03	100,00	62,13	65,61

Tabela A.2 – Valores dos parâmetros e do Vetor WSM (Exemplo 6.1, não linear geométrico)

Iteração	Volume	Deslocamento (cm, %)		Tensão (MPa, %)		IP	WSM
0	100,00	0,6623	23,06	364,17	36,87	100,00	73,56
1	98,21	0,6624	23,07	364,20	36,87	99,18	72,53
2	96,69	0,6626	23,07	365,33	36,98	97,38	71,54
3	95,21	0,6630	23,09	366,03	37,06	95,46	70,55
4	93,52	0,6638	23,12	367,57	37,21	95,36	69,68
5	91,54	0,6653	23,17	369,77	37,43	93,21	68,43
6	89,10	0,6681	23,27	376,33	38,10	93,09	67,23
7	86,15	0,6732	23,44	384,78	38,95	89,01	65,35
8	82,71	0,6813	23,72	388,30	39,31	88,90	63,65
9	77,85	0,6938	24,16	400,72	40,57	88,05	61,25
10	71,75	0,7172	24,98	422,78	42,80	86,14	58,27
11	64,62	0,7727	26,91	442,85	44,83	85,87	55,22
12	58,81	0,8396	29,24	471,08	47,69	81,72	52,59
13	54,15	0,8983	31,28	493,15	49,92	80,03	50,72
14	50,04	0,9496	33,07	505,87	51,21	78,06	48,94
15	45,18	1,0298	35,86	512,39	51,87	77,11	47,09
16	40,37	1,1572	40,30	547,82	55,46	76,34	46,00
17	36,10	1,2752	44,41	593,21	60,05	75,06	45,16
18	32,40	1,4246	49,61	589,04	59,63	77,33	44,81
19	27,37	1,7192	59,87	649,97	65,80	79,02	45,68
20	21,46	2,2438	78,14	839,80	85,02	74,51	48,76
21	19,92	2,8717	100,00	987,81	100,00	73,51	55,06

Tabela A.3 – Valores dos parâmetros e do Vetor WSM (Exemplo 6.1, não linear físico e geométrico)

Iteração	Volume	Deslocamento (cm, %)		Tensão (MPa, %)		IP	WSM
0	100,00	0,9993	8,56	409,43	73,43	100,00	73,36
1	99,35	0,9994	8,56	409,43	73,43	96,15	72,58
2	98,52	0,9995	8,56	409,44	73,43	96,15	72,15
3	97,54	0,9998	8,56	409,43	73,43	94,34	71,42
4	96,53	1,0007	8,57	409,54	73,45	93,46	70,80
5	95,23	1,0021	8,58	409,88	73,51	92,59	70,03
6	93,18	1,0047	8,61	410,25	73,58	92,59	68,97
7	91,03	1,0112	8,66	411,23	73,75	91,74	67,77
8	89,07	1,0206	8,74	412,73	74,02	81,30	65,59
9	86,13	1,0306	8,83	416,24	74,65	78,13	63,77
10	81,74	1,0471	8,97	418,71	75,09	76,92	61,42
11	76,70	1,0751	9,21	421,61	75,61	75,19	58,70
12	70,88	1,1200	9,59	424,88	76,20	70,92	55,31
13	66,38	1,1861	10,16	430,60	77,23	70,42	53,16
14	61,72	1,2559	10,76	432,53	77,57	70,42	50,91
15	57,40	1,3293	11,39	434,48	77,92	70,42	48,85
16	53,44	1,4323	12,27	433,92	77,82	70,42	47,00
17	49,67	1,5380	13,17	436,59	78,30	69,93	45,26
18	45,71	1,6768	14,36	441,05	79,10	68,97	43,47
19	42,83	1,8752	16,06	446,76	80,12	68,49	42,45
20	40,39	2,1285	18,23	448,48	80,43	68,97	41,83
21	38,93	2,2624	19,38	448,35	80,41	68,97	41,37
22	37,86	2,3509	20,14	449,16	80,55	68,49	41,08
23	36,85	2,3670	20,28	451,73	81,02	68,97	40,57
24	35,37	2,6019	22,29	453,00	81,24	68,97	40,30
25	33,73	3,2841	28,13	459,75	82,45	68,97	41,09
26	32,05	4,4934	38,49	489,66	87,82	68,49	43,05
27	30,27	7,6951	65,92	532,34	95,47	68,49	49,80
28	29,68	11,6739	100,00	557,59	100,00	64,94	58,98

Tabela A.4 – Valores dos parâmetros e do Vetor WSM (Exemplo 6.2, linear)

Iteração	Volume	Deslocamento (cm, %)		Tensão (MPa, %)		IP	WSM
0	100,00	0,9097	46,20	1632,37	76,22	100,00	83,52
1	96,57	0,9100	46,21	1632,32	76,22	97,35	81,43
2	89,29	0,9118	46,30	1631,61	76,19	94,71	77,34
3	80,91	0,9200	46,72	1621,12	75,70	93,11	72,83
4	70,53	0,9347	47,47	1618,09	75,56	78,67	65,91
5	60,52	0,9655	49,03	1627,88	76,01	77,26	60,97
6	52,58	1,0122	51,40	1641,68	76,66	76,49	57,42
7	47,88	1,0703	54,35	1667,37	77,86	75,16	55,69
8	44,23	1,1174	56,75	1679,09	78,40	73,64	54,29
9	40,54	1,1609	58,95	1694,76	79,14	72,26	52,86
10	37,36	1,2147	61,69	1699,88	79,38	71,34	51,83
11	34,70	1,2679	64,39	1704,62	79,60	70,87	51,12
12	32,56	1,3242	67,25	1721,83	80,40	69,24	50,64
13	30,62	1,3825	70,21	1743,73	81,42	69,12	50,49
14	28,94	1,4424	73,25	1783,52	83,28	67,53	50,41
15	27,35	1,5055	76,46	1828,23	85,37	63,16	50,12
16	26,20	1,5694	79,70	1847,26	86,26	60,18	50,11
17	24,85	1,6210	82,32	1859,86	86,85	58,52	49,96
18	23,60	1,6940	86,03	1892,10	88,35	58,34	50,41
19	22,41	1,7660	89,68	1927,78	90,02	56,85	50,74
20	21,11	1,8487	93,88	2007,13	93,72	53,33	51,12
21	19,95	1,9692	100,00	2141,56	100,00	50,13	52,37

Tabela A.5 – Valores dos parâmetros e do Vetor WSM (Exemplo 6.2, não linear geométrico)

Iteração	Volume	Deslocamento (cm, %)		Tensão (MPa, %)		IP	WSM
0	100,00	0,9142	44,36	1617,55	74,48	100,00	82,87
1	96,59	0,9145	44,37	1617,51	74,48	97,49	80,80
2	89,33	0,9163	44,46	1616,79	74,45	94,65	76,69
3	80,93	0,9246	44,87	1606,34	73,97	93,21	72,19
4	70,54	0,9394	45,59	1602,98	73,81	78,21	65,20
5	60,53	0,9704	47,09	1612,23	74,24	77,88	60,36
6	52,62	1,0173	49,36	1625,51	74,85	76,37	56,71
7	47,74	1,0743	52,13	1651,75	76,06	75,12	54,86
8	44,28	1,1248	54,58	1662,17	76,54	73,36	53,54
9	40,51	1,1663	56,60	1674,14	77,09	72,09	52,00
10	37,23	1,2203	59,21	1684,55	77,57	71,03	50,90
11	34,76	1,2788	62,05	1686,60	77,66	70,73	50,32
12	32,54	1,3277	64,42	1700,45	78,30	70,44	49,82
13	30,49	1,3907	67,48	1718,99	79,15	69,88	49,57
14	28,87	1,4534	70,53	1756,32	80,87	67,31	49,39
15	27,37	1,5154	73,53	1804,19	83,08	62,99	49,12
16	26,16	1,5761	76,48	1831,47	84,33	61,54	49,21
17	25,05	1,6334	79,26	1848,91	85,14	58,84	49,13
18	23,57	1,7010	82,54	1873,74	86,28	58,57	49,30
19	22,59	1,7828	86,51	1927,57	88,76	57,81	49,99
20	21,16	1,8425	89,41	1962,76	90,38	55,68	49,91
21	20,14	1,9871	96,42	2148,27	98,92	50,00	51,41
22	19,05	2,0608	100,00	2171,72	100,00	50,15	51,90

Tabela A.6 – Valores dos parâmetros e do Vetor WSM (Exemplo 6.2, não linear físico e geométrico)

Iteração	Volume	Deslocamento (cm, %)		Tensão (MPa, %)		IP	WSM
0	100,00	6,8601	10,65	565,55	84,01	100,00	74,96
1	98,88	6,8602	10,65	565,55	84,01	94,50	73,73
2	97,73	6,8604	10,65	565,55	84,01	93,30	72,99
3	96,24	6,8609	10,65	565,55	84,01	91,32	71,99
4	94,35	6,8623	10,65	565,56	84,01	88,74	70,70
5	91,78	6,8649	10,65	565,59	84,01	88,52	69,33
6	88,25	6,8704	10,66	565,65	84,02	88,30	67,46
7	83,28	6,8803	10,68	565,74	84,04	87,80	64,81
8	76,90	6,8978	10,70	565,91	84,06	87,58	61,46
9	69,85	6,9247	10,75	565,81	84,05	85,48	57,54
10	64,55	6,9606	10,80	565,55	84,01	84,84	54,71
11	60,17	6,9973	10,86	565,12	83,94	83,07	52,22
12	56,35	7,0346	10,92	564,83	83,90	82,14	50,12
13	52,99	7,0748	10,98	564,83	83,90	81,81	48,34
14	50,05	7,1228	11,05	564,28	83,82	81,75	46,81
15	47,73	7,1763	11,14	563,69	83,73	81,02	45,53
16	45,54	7,2253	11,21	563,54	83,71	80,77	44,38
17	43,45	7,2793	11,30	562,60	83,57	80,63	43,27
18	41,24	7,3354	11,38	562,38	83,54	78,57	41,90
19	39,86	7,4092	11,50	562,32	83,53	77,68	41,10
20	38,10	7,4726	11,60	561,46	83,40	77,15	40,13
21	36,71	7,5704	11,75	560,63	83,28	77,11	39,43
22	35,59	7,6633	11,89	559,49	83,11	76,79	38,83
23	34,58	7,7812	12,07	557,64	82,83	77,09	38,36
24	33,52	7,8958	12,25	555,32	82,49	76,55	37,75
25	32,60	8,0663	12,52	554,15	82,31	76,17	37,28
26	31,80	8,2450	12,79	549,15	81,57	76,64	36,91
27	31,00	8,4914	13,18	543,45	80,73	74,34	36,25
28	30,28	8,8496	13,73	541,58	80,45	74,40	35,99
29	29,67	9,4106	14,60	538,72	80,02	74,16	35,84
30	28,90	9,9753	15,48	539,22	80,10	73,34	35,57
31	28,29	11,0532	17,15	542,65	80,61	70,84	35,46
32	27,78	11,8084	18,32	543,61	80,75	70,40	35,46
33	27,07	12,5476	19,47	539,62	80,16	68,75	35,14
34	26,29	13,6999	21,26	540,82	80,33	67,92	35,12
35	25,77	15,8836	24,65	564,90	83,91	65,34	35,80
36	25,13	16,9399	26,29	575,97	85,56	63,10	35,80
37	24,44	18,3446	28,47	580,37	86,21	62,43	35,99
38	24,01	20,2965	31,50	593,57	88,17	60,77	36,57
39	23,33	21,2907	33,04	603,39	89,63	59,51	36,62
40	22,84	24,8809	38,61	637,85	94,75	55,96	37,92
41	22,48	28,0998	43,61	641,87	95,34	55,39	39,04
42	22,10	31,5982	49,03	652,20	96,88	53,59	40,21
43	21,80	36,7033	56,96	660,95	98,18	52,85	42,17
44	21,44	41,9853	65,15	660,47	98,11	52,73	44,11
45	21,32	50,4438	78,28	664,44	98,70	52,61	47,54
46	21,05	51,2604	79,55	664,97	98,77	52,34	47,71
47	21,04	56,9720	88,41	673,21	100,00	51,08	50,01
48	20,94	54,8702	85,15	669,77	99,49	50,90	49,03
49	20,84	56,6033	87,84	648,39	96,31	52,76	49,58
50	20,42	57,3003	88,92	651,42	96,76	51,98	49,60
51	20,27	63,8740	99,12	670,77	99,64	50,08	52,27
52	19,91	64,4416	100,00	664,36	98,68	49,24	52,12

Tabela A.7 – Valores dos parâmetros e do Vetor WSM (Exemplo 6.3, linear)

Iteração	Volume	Deslocamento (cm, %)		Tensão (MPa, %)		IP	WSM
0	100,00	0,5771	31,53	410,94	44,99	100,00	76,59
1	99,30	0,5772	31,53	410,96	44,99	81,82	74,12
2	98,33	0,5773	31,54	411,05	45,00	78,42	73,22
3	96,92	0,5778	31,57	411,36	45,04	78,38	72,49
4	94,66	0,5789	31,63	412,36	45,15	78,30	71,32
5	92,17	0,5818	31,78	415,03	45,44	77,91	70,05
6	89,19	0,5856	31,99	419,22	45,90	78,14	68,61
7	84,67	0,5911	32,30	424,42	46,47	75,86	66,12
8	79,04	0,6011	32,84	431,39	47,23	75,20	63,32
9	71,02	0,6171	33,71	445,58	48,78	74,29	59,40
10	61,15	0,6481	35,41	477,57	52,29	72,61	54,84
11	50,39	0,6958	38,01	517,50	56,66	71,37	50,18
12	41,89	0,7772	42,46	542,29	59,37	73,70	47,45
13	35,71	0,9155	50,01	586,07	64,16	70,25	46,27
14	31,44	1,0248	55,99	634,66	69,48	68,61	45,95
15	28,98	1,1668	63,74	707,17	77,42	63,93	46,94
16	27,49	1,2395	67,72	740,57	81,08	59,95	47,11
17	25,95	1,3363	73,00	766,91	83,96	56,44	47,56
18	24,61	1,4050	76,76	777,31	85,10	54,75	47,77
19	23,36	1,4774	80,71	795,75	87,12	55,58	48,45
20	21,88	1,5531	84,85	813,93	89,11	55,33	48,93
21	20,25	1,6653	90,98	862,46	94,42	51,84	49,81
22	19,11	1,8304	100,00	913,40	100,00	51,01	52,03

Tabela A.8 – Valores dos parâmetros e do Vetor WSM (Exemplo 6.3, não linear geométrico)

Iteração	Volume	Deslocamento (cm, %)		Tensão (MPa, %)		IP	WSM
0	100,00	0,5778	30,52	412,84	46,14	100,00	76,43
1	99,29	0,5779	30,52	412,86	46,14	81,79	73,96
2	98,30	0,5780	30,53	412,96	46,15	78,38	73,05
3	96,87	0,5786	30,56	413,27	46,19	78,34	72,31
4	94,56	0,5796	30,62	414,28	46,30	78,25	71,11
5	92,08	0,5826	30,77	417,09	46,61	77,85	69,84
6	89,07	0,5864	30,97	421,13	47,06	78,09	68,39
7	84,63	0,5920	31,27	426,67	47,68	75,77	65,94
8	78,98	0,6020	31,80	433,90	48,49	75,14	63,13
9	70,67	0,6195	32,72	450,01	50,29	74,58	59,14
10	60,94	0,6501	34,34	482,68	53,94	72,33	54,58
11	50,55	0,6964	36,78	519,45	58,05	70,98	50,04
12	41,76	0,7775	41,07	543,80	60,77	73,59	47,14
13	35,71	0,9218	48,69	590,34	65,97	70,53	46,14
14	31,42	1,0286	54,33	636,69	71,15	68,69	45,68
15	28,97	1,1702	61,81	709,46	79,29	63,57	46,57
16	27,31	1,2462	65,82	741,56	82,87	60,39	46,74
17	25,91	1,3503	71,32	771,54	86,22	56,43	47,32
18	24,60	1,4151	74,75	781,71	87,36	55,16	47,51
19	23,49	1,4813	78,24	798,95	89,29	55,47	48,07
20	21,82	1,5550	82,13	823,98	92,09	55,10	48,45
21	20,13	1,6799	88,73	868,45	97,05	51,82	49,41
22	19,06	1,8933	100,00	894,80	100,00	51,24	52,04

Tabela A.9 – Valores dos parâmetros e do Vetor WSM (Exemplo 6.3, não linear físico e geométrico)

Iteração	Volume	Deslocamento (cm, %)		Tensão (MPa, %)		IP	WSM
0	100,00	0,6511	32,61	291,91	71,69	100,00	79,51
1	99,35	0,6511	32,61	291,91	71,69	94,48	78,53
2	98,44	0,6512	32,62	291,93	71,69	92,93	77,88
3	97,37	0,6517	32,64	291,98	71,71	91,77	77,19
4	95,37	0,6525	32,68	292,08	71,73	87,08	75,61
5	93,07	0,6548	32,79	292,84	71,92	85,25	74,25
6	90,47	0,6581	32,96	293,80	72,15	83,95	72,81
7	86,96	0,6629	33,20	294,96	72,44	83,11	70,96
8	82,17	0,6704	33,58	296,72	72,87	82,84	68,57
9	76,38	0,6825	34,19	299,30	73,50	82,03	65,67
10	68,83	0,7023	35,18	302,84	74,37	81,61	62,02
11	60,84	0,7345	36,79	312,03	76,63	84,33	58,80
12	53,23	0,7777	38,95	322,03	79,09	83,80	55,57
13	46,65	0,8333	41,74	329,19	80,85	83,97	53,06
14	42,12	0,9102	45,59	343,04	84,24	80,26	51,61
15	39,17	0,9863	49,40	354,28	87,00	79,14	51,21
16	36,13	1,0451	52,35	358,82	88,12	78,79	50,46
17	34,22	1,1101	55,60	360,46	88,52	77,75	50,24
18	32,69	1,1707	58,63	364,43	89,50	76,69	50,21
19	30,53	1,2265	61,43	365,38	89,73	77,19	49,89
20	28,94	1,3028	65,25	369,28	90,69	77,01	50,14
21	27,65	1,3707	68,65	371,96	91,35	74,66	50,15
22	26,46	1,4246	71,35	367,44	90,24	74,34	50,09
23	25,00	1,4881	74,54	368,59	90,52	72,89	50,02
24	23,48	1,5772	79,00	370,12	90,90	72,93	50,44
25	22,38	1,6732	83,81	375,95	92,33	71,04	51,05
26	21,52	1,7550	87,90	379,48	93,20	69,97	51,64
27	20,68	1,8683	93,58	388,77	95,48	68,51	52,75
28	19,91	1,9965	100,00	407,19	100,00	66,07	54,20

APÊNDICE B

DESCRIÇÃO DOS PROGRAMAS UTILIZADOS E DO CÓDIGO IMPLEMENTADO

B.1 Lazarus®

Lazarus® é um *software* de código livre IDE (Integrated Development Environment), altamente compatível com o Borland Delphi®, para o desenvolvimento de aplicações rápidas. Possui uma variedade de componentes prontos para serem usados e um formulário gráfico (Form) designado para a fácil criação de complexas interfaces com o usuário, denominadas GUI (Graphical User Interfaces). Ainda, o programa permite o desenvolvimento de aplicativos do tipo *open source* ou comerciais incluindo visualizadores de imagens, aplicativos de banco de dados, editores gráficos, programas 3D, entre outros.

O Lazarus® pode ser definido como a associação de um Ambiente de Desenvolvimento Integrado (IDE), uma linguagem moderna de programação (o *Object Pascal*) e uma Biblioteca de Componentes Visuais (ou *Visual Component Library*, ou VCL), que constituem os objetos, de acordo com a Fig. B.1. A programação em Lazarus® é modulada, baseada em unidades (*Units*), de forma que estas podem ser importadas ou exportadas entre diferentes projetos, ou seja, várias unidades podem ser agrupadas em um único projeto, trabalhando em conjunto ou, da mesma forma, vários projetos podem ser reunidos cada qual contendo uma única unidade. Essas unidades representam toda a programação em modo texto, onde pode-se editar os códigos necessários para o funcionamento do programa. Porém, cada unidade pode ou não estar atrelada aos formulários (*Forms*), que permitem ao usuário verificar graficamente os comandos implementados nas unidades, auxiliando no bom funcionamento do programa e facilitando a detecção de possíveis erros de programação.

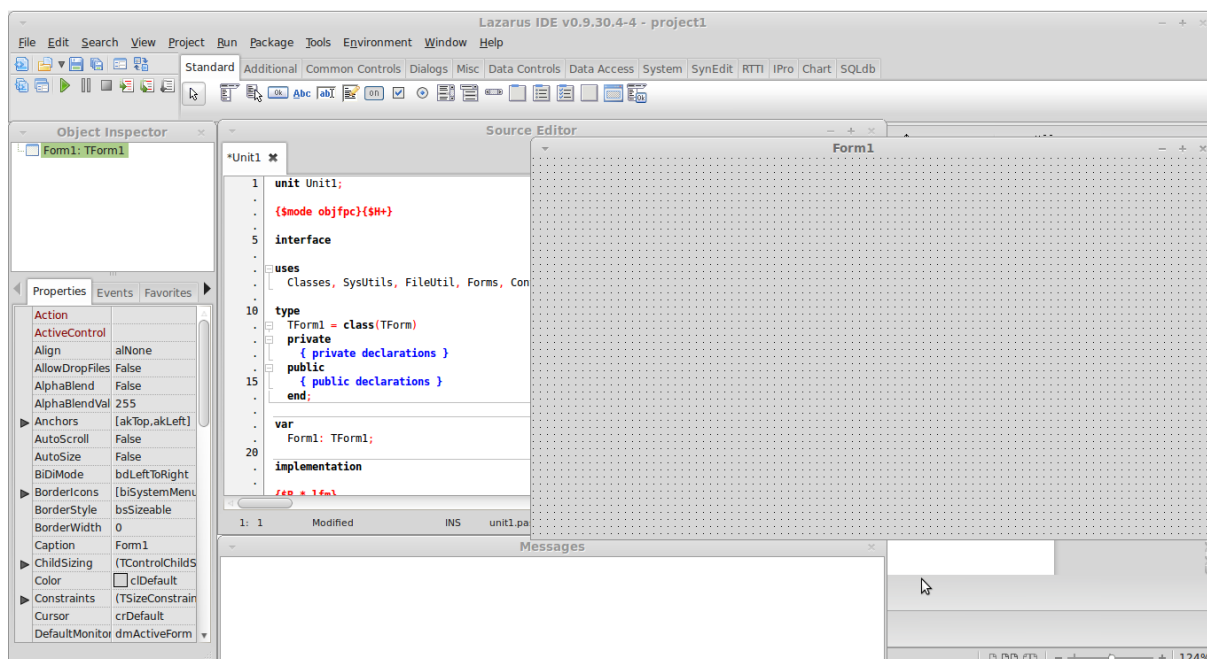


Figura B.1 – Lazarus[®], a IDE *open source* compatível com o programa Borland Delphi[®]

A estrutura do Lazarus[®] é orientada a objetos e eventos. Cada objeto possui propriedades que o caracterizam, além de eventos a ele associados. O instante em que determinado código implementado é ativado pelo programa depende do evento em que esse código está inserido. Assim, a elaboração de um aplicativo em Lazarus[®] pode ser dividida em duas partes: tempo de projeto e tempo de execução. O primeiro está intimamente relacionado ao desenvolvedor, e diz respeito às modificações que podem ser atribuídas aos componentes VCL ou ao código-fonte da linguagem. O segundo se relaciona ao usuário final, e diz respeito à execução das tarefas atribuídas aos componentes VCL, definidas em tempo de projeto, por dispositivos de entrada de dados (como mouse e teclado) ou por rotinas específicas associadas ao sistema operacional.

O Lazarus[®] possui uma interface amigável de desenvolvimento que favorece uma boa interação entre o usuário e o computador através das ferramentas que possui, diretamente aplicadas na implementação dos códigos do programa. É nela onde são utilizados os componentes que farão parte do programa desenvolvido. Em outras palavras, nesse ambiente os códigos nas chamadas *Units* são implementados, geralmente vinculados a formulários (*Forms*), os quais agregam os componentes que se deseja inserir no programa. Esses componentes podem ser botões, barra de menus, caixas de texto, entre outros. Cada

componente possui suas próprias características e propriedades que podem ser modificadas inicialmente ou ao longo da execução do programa, de forma a facilitar para o programador. Uma vez que esses componentes forem inseridos, funções, métodos ou eventos podem ser ativados em tempo de execução ou de projeto a fim de se obter os resultados desejados.

É importante ressaltar que o código para o compilador *Free Pascal* está licenciado sob a GNU¹ GPL (*GNU General Public License*), o que significa que o programa é de código aberto, livre ou qualquer denominação semelhante. Assim, é permitido modificar o código caso necessário, mas deve-se distribuir os programas modificados ou disponibilizá-los para os outros que queiram utilizá-lo. Além disso, a FCL (*Free Pascal Component Libraries*) é licenciada sob a LGPL (*GNU Lesser General Public License*, versão 2.1) modificada, o que significa que é permitido escrever programas próprios que utilizam essas bibliotecas, sendo ainda permitido vender esses programas produzidos sem a necessidade de fornecer ou disponibilizar o código criado. No entanto, assim como com o compilador, ao se fazer modificações na FCL essas alterações devem estar disponíveis para o público em geral e para o mundo todo.

Assim como vários *softwares* livres, várias pessoas colaboraram e ainda contribuem bastante para o projeto. Inicialmente, Cliff Baeseman, Shane Miller e Michael A. Hess permaneceram no projeto após o antigo Megido (versão anterior ao Lazarus[®]) acabar. A partir do início do desenvolvimento do Lazarus[®], os desenvolvedores Marc Weustink, Mattias Gaertner, Micha Nelissen, Vincent Snijders, Felipe Monteiro de Carvalho, Paul Ishenin, Zeljan Rikalo, Martin Friebe, Juha Manninen, Bart Broersma e Roozbeh Gholizadeh, como colaborador, integram a equipe do projeto ou já contribuíram bastante para o seu crescimento.

1 GNU (ou GNU/Linux) refere-se a um sistema operacional tipo Unix que é um *software* livre, ou seja, é *open source* (código aberto). O nome “GNU” é um acrônimo para “GNU is Not Unix!” ou “GNU Não é Unix!” Um sistema operacional do tipo Unix é um *software* que possui uma coleção de aplicações, bibliotecas e ferramentas de desenvolvimento, além de um programa que aloca recursos e se comunica com o hardware, denominado kernel. Como hoje em dia é comum o sistema GNU possuir um kernel chamado Linux (e esta combinação é chamada GNU/Linux), muitos o chamam (erroneamente) apenas por Linux por uma questão de comodidade.

B.2 Python[®]

Python[®] é uma linguagem de programação de código livre, poderosa e fácil de se aprender. Possui estruturas de dados eficientes e linguagem de alto nível (mais próxima a humana), além de uma abordagem simples, porém eficaz, para programação orientada a objetos. Sua sintaxe elegante e dinâmica, em conjunto com a sua natureza interpretada, faz com que o Python[®] seja ideal para a criação de *scripts* e desenvolvimento rápido de aplicações em muitas áreas na maioria das plataformas.

Pelo fato de ser uma linguagem de alto nível, os programas escritos em Python[®] necessitam ser processados antes de serem executados, uma vez que os computadores executam apenas programas escritos em linguagem de baixo nível, ou seja, linguagens mais próximas à máquina, o que demanda certo tempo de processamento e faz com que isso seja uma pequena desvantagem da linguagem. Entretanto, o fato de ser uma linguagem de alto nível possui grandes vantagens. Primeiro pela facilidade para se programar uma vez que demandam pouco tempo para escrever, tornando os algoritmos menores, mais fáceis para ler e mais próximo do correto. Segundo, pelo fato de serem portáveis, o que implica que podem ser executados em diferentes tipos de computadores com poucas ou até mesmo nenhuma modificação no código. Em contrapartida, programas feitos em linguagem de baixo nível podem ser executados apenas em um tipo de computador, necessitando serem reescritos para serem executados em outros dispositivos (DOWNEY, 2012).

De forma geral, linguagens de programação facilitam várias tarefas em um trabalho qualquer que envolva computadores. Automatizar uma tarefa proporciona um bom ganho de tempo para a realização da mesma. Por exemplo, um programa que realiza buscas e substituições ao longo de vários arquivos de texto, ou algum que renomeie e reorganize arquivos de fotos de uma forma específica. Talvez ainda um pequeno banco de dados personalizado, ou um aplicativo gráfico exclusivo, ou um simples jogo. Ou seja, são muitas vantagens que a programação pode oferecer.

Em comparação a outras linguagens de programação, como C[®], C++[®], Fortran[®], Borland Delphi[®] e algumas outras, os programas feitos em Python[®] não são compilados; ao invés

disso, eles são interpretados. Um programa, escrito em uma linguagem qualquer de programação, pode ser convertido, ou traduzido, em código de máquina por compilação ou interpretação. Se o método utilizado traduz todo o código do programa para apenas depois o mesmo ser executado, então se diz que o programa foi compilado e que o mecanismo utilizado para a tradução é um compilador (que nada mais é do que um programa de computador dedicado a este fim). A versão compilada do programa normalmente é armazenada, de forma que o programa pode ser executado um número indefinido de vezes sem que seja necessária nova compilação, o que compensa o tempo gasto na compilação. Isso acontece com linguagens como Pascal e C (linguagem de programação).

Se o texto do programa é traduzido na medida em que vai sendo executado, como em Javascript, Python®, Perl® ou PHP®, em um processo de tradução de trechos seguidos de sua execução imediata, então se diz que o programa foi interpretado e que o mecanismo utilizado para a tradução é um interpretador. Programas interpretados são geralmente mais lentos do que os compilados, mas são também geralmente mais flexíveis e podem trabalhar seguindo um fluxo de execução intermitente (método ideal para aplicações *web*). Frequentemente, linguagens interpretadas são chamadas também de *script*. A Fig. B.2 apresenta um *script* em Python® sendo mostrado através de um interpretador, o SPE (*Stani's Python Editor*).

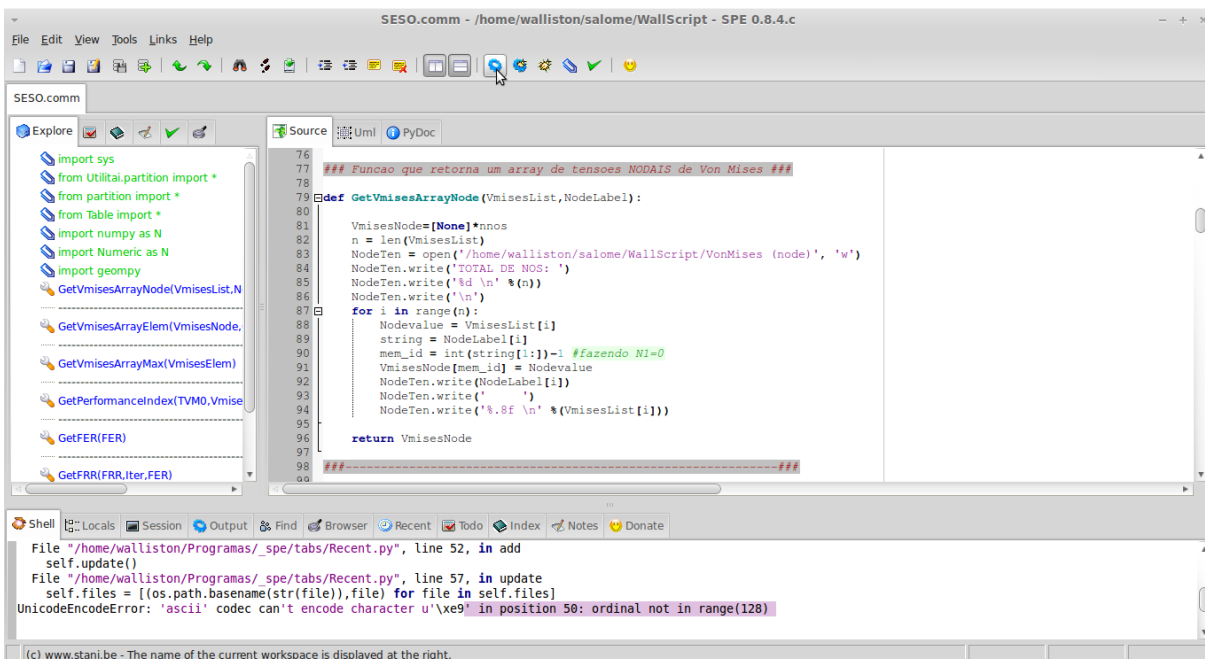


Figura B.2 – SPE® (*Stani's Python Editor*), um interpretador *open source* Python®

A vantagem de se utilizar *scripts* para implementação de códigos está no fato deles serem compatíveis com outros computadores ou, até mesmo, outras plataformas (sistemas operacionais). Assim, um *script* feito em um sistema Linux[®], ou baseado em Unix[®], será aceito em um sistema Windows ou Mac OS X[®], e vice-versa, sem a necessidade de adaptações (exceto devido às versões diferentes do interpretador, em alguns casos). Com relação ao Python[®], os interpretadores estão disponíveis para todos esses sistemas descritos (Windows[®], Mac OS X[®] e sistemas operacionais Unix[®], como Linux[®]). Para um *script* ser aceito pelo interpretador, de acordo com as regras dessa linguagem, é necessário que o mesmo possua a extensão “py”.

Python[®] é simples de usar e é uma linguagem de programação completa, oferecendo grande estrutura e suporte para programas extensos muito além dos *scripts* que a linguagem pode oferecer. Além disso, permite que os programas sejam escritos de forma compacta e legível, tornando o código em si menor (menos linhas de implementação) comparados ao mesmo código escrito em outras linguagens de programação. Isso acontece por várias razões:

- Por ser uma linguagem de alto nível, operações complexas podem ser expressas por uma simples instrução;
- Os agrupamentos são feitos por endentação ou recuo (espaçamento adequado dos blocos que correspondem ao código implementado) e não há declarações de início e término como na maioria das outras linguagens;
- Nenhuma declaração de variável ou argumento de forma prévia é necessária.

A linguagem de programação Python[®] é distribuída sob uma licença própria (compatível com a GNU/GPL), que impõe poucas restrições. É permitida a distribuição, comercial ou não, tanto da linguagem quanto de aplicações desenvolvidas nela, em formato binário ou código fonte, bastando cumprir a exigência de manter o aviso de *Copyright* da PSF (*Python software Foundation*).

B.3 FreeCAD[®]

FreeCAD[®] é um *software* de código livre que realiza modelagens paramétricas 3D, o que permite ao usuário modificar facilmente o desenho através da alteração dos parâmetros originais do modelo. É voltado diretamente para o público de engenharia mecânica, mas também se encaixa em uma ampla gama de usos em relação a área de engenharia e desenhos de forma geral, tais como civil, arquitetura, design, entre outros. Possui ferramentas similares a programas comerciais, como Catia[®], SolidWorks[®] ou Solid Edge[®].

O programa faz intenso uso de várias das grandes bibliotecas de desenvolvimento gráfico em código aberto que existem no campo da computação gráfica. Entre elas estão o OpenCascade[®], que é um poderoso kernel de CAD; o Coin3D[®], que é uma implementação do OpenInventor[®] que, por sua vez, é um pacote gráfico orientado a objeto que oferece uma adequada solução para problemas de programação que envolvem iterações gráficas; o Qt[®], que é um *framework*² mundialmente famoso e o Python[®], que é uma das melhores linguagens disponíveis para a criação de *scripts*. FreeCAD[®] pode também ser utilizado como uma biblioteca por outros programas.

A Fig. B.3 mostra a janela principal do programa, incluindo possibilidades para várias áreas que necessitam de modelagem gráfica.

2 *Framework* é uma coleção de classes que captura uma funcionalidade comum de várias aplicações, sendo que estas devem possuir algo razoavelmente grande em comum, como pertencer a um mesmo domínio de problema a ser resolvido.

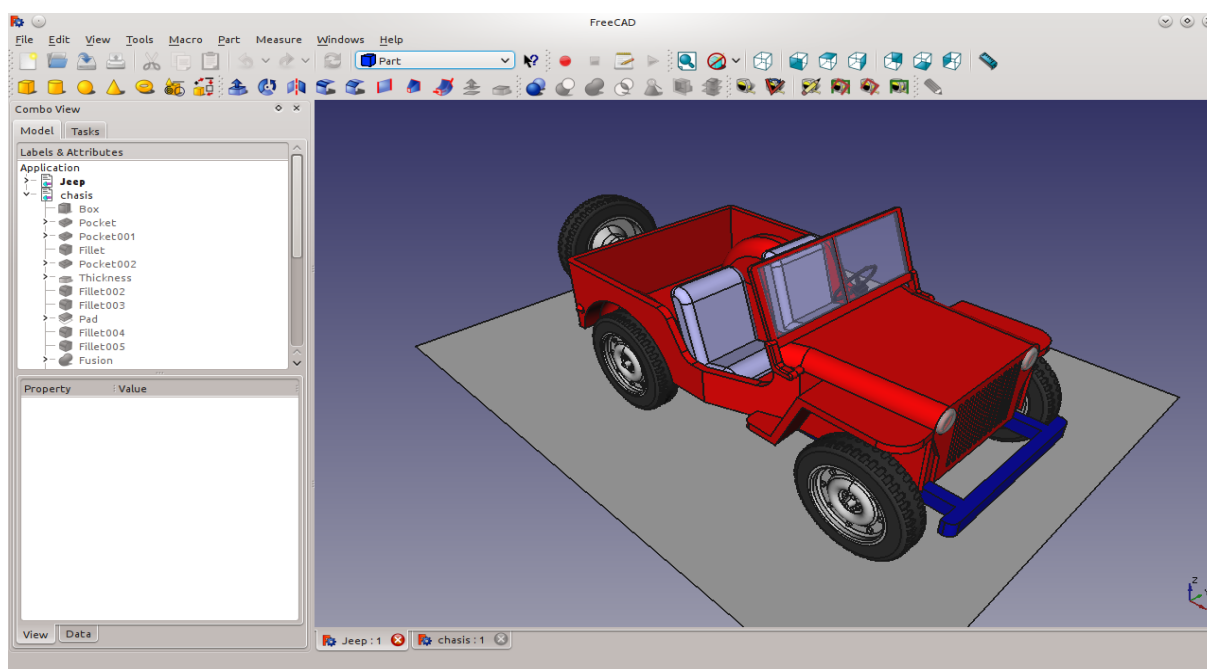


Figura B.3 – FreeCAD[®], uma ferramenta *open source* de modelagens paramétricas 3D³

Outra vantagem do programa está no fato de ser multiplataforma, ou seja, funciona perfeitamente em ambientes Windows, Linux/Unix e Mac OSX, com a mesma aparência e funcionalidade em todas estas plataformas.

FreeCAD[®] permite importar e exportar o projeto em vários formatos de desenho que podem ser aceitos em programas comerciais ou livres, garantindo assim significativa compatibilidade com outros programas para se fazer variados tipos de análise.

Assim como vários *softwares* livres, várias pessoas colaboraram e ainda contribuem bastante para o projeto. FreeCAD[®] é gerenciado por Jürgen Riegel, Werner Mayer e Yorik van Havre (este último vive na cidade de São Paulo atualmente) e possui como principais desenvolvedores Logari81, Luke A. Parry, Jose Luis Cercos Pita, Jan Rheinlaender, shoogen, tanderson69. Além disso, o programa está inserido sob duas licenças, a GNU *Lesser General Public Licence* (GNU LGPL2+ ou GNU LGPL versão 2.1) e a *Open Publication License*.

3 Extraído de <http://www.freecadweb.org/wiki/images/5/56/Freecad_jeep.png>. Acesso em 27 de fev. 2014.

B.4 Code-Aster®

Code-Aster® é um *software* de código livre de análise estrutural, via Método dos Elementos Finitos, desenvolvido pela EDF-R&D™ (*Électricité de France - Research and Development*) que oferece uma vasta opção de tipos de análise e modelagem que vão muito além das funções normais de um programa comum que segue essa linha, indo desde análises sísmicas à acústica de meios porosos, fadiga, fratura, dano, dinâmica, térmica, entre outras aplicações. O algoritmo foi desenvolvido em linguagem de programação Fortran® e está em constante atualização, contando atualmente com cerca de 1,2 milhão de linhas de código.

O programa é disponibilizado em três versões. A primeira versão é livre, a segunda é operacional e a terceira é de desenvolvimento. Todas as versões são originadas de um único código fonte compatível com sistemas baseados em Unix e Linux, apesar de existirem versões pouco mais antigas disponíveis para plataformas Windows graças à comunidade Aster (composto pelo fórum e grupo de discussão). A versão operacional é a versão usada pela equipe da EDF-R&D™ e seus fornecedores autorizados, e sua principal característica é a preocupação com a segurança, sendo, assim, alvo constante de manutenções corretivas e validações por um período de dois anos. A versão de desenvolvimento é enriquecida a cada semana com correções e inovações, totalizando cerca de 50 atualizações anuais. Após dois anos, esta versão se torna a versão operacional. O interessante aqui é que mais de 70 engenheiros já contribuíram para o desenvolvimento deste programa, e cerca de 20 engenheiros da EDF-R&D™ são designados para verificar a consistência e qualidade do *software*.

Ao se tratar de fenômenos mecânicos, o programa possibilita a realização de análise estática linear e não linear, dinâmica (com ou sem amortecimento, sendo linear ou não linear), fratura, dano, fadiga e interações solo-estrutura, fluido-estrutura e solo-fluido-estrutura. Fenômenos acústicos e térmicos também fazem parte das possibilidades desta ferramenta, além ainda, de outros tipos. O programa ainda permite a combinação de análises, sendo possível uma análise termo-hídrica, por exemplo. Sua biblioteca conta com cerca de 480 tipos de elementos finitos e 100 leis constitutivas, incluindo elementos lineares, planos e espaciais (3D). Além disso, o *solver* inclui vários procedimentos clássicos para os mais diversos cálculos numéricos, como

Newton-Raphson, Newmark, Runge-Kutta, Lanczos, entre outros.

É também um programa flexível e aberto, permitindo ao usuário a possibilidade de criar elementos finitos, de implementar novas leis constitutivas ou modificar alguma já existente, diretamente no programa, e uma inserção de comandos via Python[®] que pode ser desde novas leis constitutivas e novos elementos finitos (de forma indireta, neste caso) até intervenções diretas no modo de cálculo, como rotinas de otimização que foram realizadas neste trabalho. Essa intervenção pode ser feita diretamente em seu arquivo principal de entrada (formato “*.comm”) ou importada a partir de um *script* Python[®] (formato “*.py”), sendo uma pequena desvantagem a necessidade da utilização de macro-rotinas próprias para as análises.

Macro-rotinas (ou macrocomandos) são comandos nativos do Code-Aster[®], indispensáveis para a realização das análises, e que podem trabalhar em conjunto com códigos implementados em Python[®]. Logo, no mesmo arquivo “*.comm” são editadas as macro-rotinas com as variáveis, funções e demais códigos em Python[®]. Algumas macro-rotinas são indispensáveis no arquivo de entrada para qualquer análise, merecendo certo destaque e devem ser implementadas em um formato padrão, respeitando as regras impostas pelo programa. São escritas em um arquivo de texto simples e podem ser editadas pelo próprio usuário ou através de um programa específico para a geração e modificação de arquivos de comando, denominado EFICAS[®] (descrito a seguir). Pouco mais de 250 macro-rotinas diferentes compõem as opções possíveis disponíveis pelo programa para a realização de uma análise, sendo que cada macro-rotina possui particularidades e parâmetros específicos para serem definidos, compondo um subconjunto de novas opções em formato “tree view”, ou seja, formato hierárquico. Algumas opções são obrigatórias enquanto existem outras que são opcionais. A Fig. B.4 apresenta um exemplo de um arquivo “*.comm” de uma análise mecânica linear.

```

DEBUT(PAR_LOT='NON',);

Steel=DEFI_MATERIAU(ELAS=_F(E=2.1e+11,
                            NU=0.3,
                            RHO=7850,));

Mesh=LIRE_MAILLAGE(UNITE=20,
                  FORMAT='MED',
                  INFO=1,);

Mate=AFFE_MATERIAU(MAILLAGE=Mesh,
                  AFFE=_F(TOUT='OUI',
                          MATER=Steel,));

Mesh=MODI_MAILLAGE(reuse =Mesh,
                  MAILLAGE=Mesh,
                  ORIE_PEAU_3D=_F(GROUP_MA='all',));

Mode=AFFE_MODELE(MAILLAGE=Mesh,
                 AFFE=_F(TOUT='OUT',
                         PHENOMENE='MECANIQUE',
                         MODELISATION='3D',));

Char=AFFE_CHAR_MECA(MODELE=Mode,
                   DDL_IMPO=_F(GROUP_NO='rest',
                               DX=0,
                               DY=0,
                               DZ=0,),
                   FORCE_NODALE=_F(GROUP_NO='load',
                                   FZ=-100,));

Resu=MECA_STATIQUE(MODELE=Mode,
                  CHAM_MATER=Mate,
                  EXCIT=_F(CHARGE=Char,));

Resu=CALC_CHAMP(reuse =Resu,
                RESULTAT=Resu,
                CONTRAINTE=('SIGM_NOEU','SIGM_ELNO'),
                CRITERES=('SIEQ_ELNO','SIEQ_NOEU'),
                FORCE=('FORC_NODA','REAC_NODA',));

IMPR_RESU(FORMAT='MED',
          RESU=_F(MAILLAGE=Mesh,
                 RESULTAT=Resu,
                 NOM_CHAM='SIEQ_NOEU',
                 NOM_CMP='VMIS',));

FIN();
#CHECKSUM:4261d0e94999e885f0687b93326bd4e3 -:FIN CHECKSUM

```

Figura B.4 – Exemplo de arquivo “*.comm” do Code-Aster com as devidas macro-rotinas

As descrições das macro-rotinas são brevemente listados a seguir.

- ◆ DEBUT(): macro-rotina que define o início da sequência de comandos do Code-Aster®, sendo seu uso obrigatório.
- ◆ DEFI_MATERIAU(): macro-rotina que define as propriedades do material (módulo de elasticidade longitudinal, coeficiente de Poisson, densidade). Pode ser usado mais

de uma vez, o que garante que o Code-Aster® pode analisar uma estrutura com n materiais diferentes. Necessita de um nome para cada material definido.

- ◆ `LIRE_MALLAGE()`: macro-rotina que lê um tipo de formato de malha informado pelo usuário que será usada para a análise. Necessita de um nome.
- ◆ `AFFE_MATERIAU()`: macro-rotina utilizada para atribuir um material, definido previamente, aos elementos de uma malha. Aqui, várias definições podem ser feitas de acordo com os grupos de malha (ou de geometria) criados pelo usuário. Necessita de um nome, de uma macro-rotina de material e de uma macro-rotina de malha, ambas definidas previamente.
- ◆ `MODI_MALLAGE()`: macro-rotina que reorienta elementos de uma malha definida previamente. Entre suas várias possibilidades, é permitido, por exemplo, a geração de uma malha simétrica 3D a partir de uma malha definida em 2D, ou apenas uma realocação dos elementos ao longo da malha, mantendo-se na dimensão atual. Serve também para atualizar uma malha a partir de uma forma deformada calculada anteriormente. Necessita de um nome e de uma macro-rotina de malha definida previamente.
- ◆ `AFFE_MODELE()`: macro-rotina utilizada para atribuir um modelo e um determinado tipo de elemento a uma malha, criando, efetivamente, um modelo de elementos finitos. Necessita de um nome e de uma macro-rotina de malha definida previamente.
- ◆ `AFFE_CHAR_MECA()`: macro-rotina utilizada para atribuir condições de contorno, carregamentos e restrições a nós, elementos, faces e/ou volumes. Necessita de um nome e de uma macro-rotina de modelo definida previamente.
- ◆ `MECA_STATIQUE()`: macro-rotina que realiza a montagem do conjunto discreto de equações de elementos finitos para resolvê-lo de forma linear, estática e mecânica, gerando os resultados. É interessante ressaltar que vários modelos podem ser definidos no arquivo de entrada, mas aqui um deles será escolhido para ser calculado. Isso implica que vários modelos podem ser calculados de forma paralela. Para uma análise não-linear a macro-rotina `STAT_NON_LINE()` deve ser usada. Já para análises dinâmicas existe um conjunto de macro-rotinas específicas, como `DYNA_NON_LINE()`, `DYNA_TRAN_MODAL()`, `DYNA_LINE_HARM()`, entre outras, assim como existem `THER_LINEAIRE()`, `THER_NON_LINE()`, `THERM_NON_LINE_MO()` para análises térmicas, e igualmente para outros tipos de

análise. Necessita de um nome, de uma macro-rotina de modelo, de uma macro-rotina de material e de uma macro-rotina de condições impostas ao modelo (contorno, carregamento e restrições), todas definidas previamente.

- ◆ **CALC_CHAMP()**: macro-rotina usada para calcular os valores de deslocamentos, tensões, deformações, entre outros, pelos pontos de Gauss do elemento, de forma nodal (geral) ou pelos nós de cada elemento, considerados separadamente em relação aos nós dos elementos vizinhos. Por exemplo, SIGM_NOEU representa a tensão calculada em cada nó ao longo de todo o modelo, SIGM_ELNO representa a tensão calculada nos nós de cada elemento desconsiderando a contribuição dos elementos vizinhos e SIGM_ELGA representa a tensão calculada nos pontos de Gauss de cada elemento. Essa macro-rotina trabalha em conjunto com MECA_STATIQUE(), anexando os valores calculados ao arquivo de saída gerado por essa macro-rotina. Por isso, CALC_CHAMP() necessita de um nome e de uma macro-rotina de resultados definida previamente.
- ◆ **IMPR_RESU()**: macro-rotina usada para organizar e imprimir os resultados desejados em um ou mais arquivos de saída. O formato de saída pode ser escolhido aqui, assim como quais parâmetros devem ser impressos no arquivo. Caso nenhum parâmetro seja definido, todo o conteúdo da macro-rotina de resultados CALC_CHAMP() será impresso no arquivo. Necessita de uma macro-rotina de resultados e, caso o formato de saída seja MED, de uma macro-rotina de malha, ambos definidos previamente.
- ◆ **FIN()**: macro-rotina que define o fim da sequência de comandos do Code-Aster®. Seu uso é obrigatório, assim como o uso da macro-rotina DEBUT().

Com o arquivo de entrada devidamente editado, basta importá-lo para o Code-Aster® com o arquivo de malha anteriormente definido. A Fig. B.5 mostra a janela principal do programa, à frente, e uma janela de status, ao fundo. Através da janela principal, além da simples realização da análise feita pelo botão “run”, existe a possibilidade de informar ao programa a quantidade de memória total que poderá ser utilizada por ele e o tempo máximo permitido para a análise. Caso seja ultrapassado, a análise é interrompida. Nesta janela ainda, pode-se chamar o programa EFICAS® (descrito a seguir) para a criação do arquivo de entrada. Já a janela de status mostra o tempo gasto pela análise, atualizado periodicamente, assim como mensagens importantes referentes ao trabalho executado. Em outras palavras, mostra o

progresso da análise ao longo do cálculo.

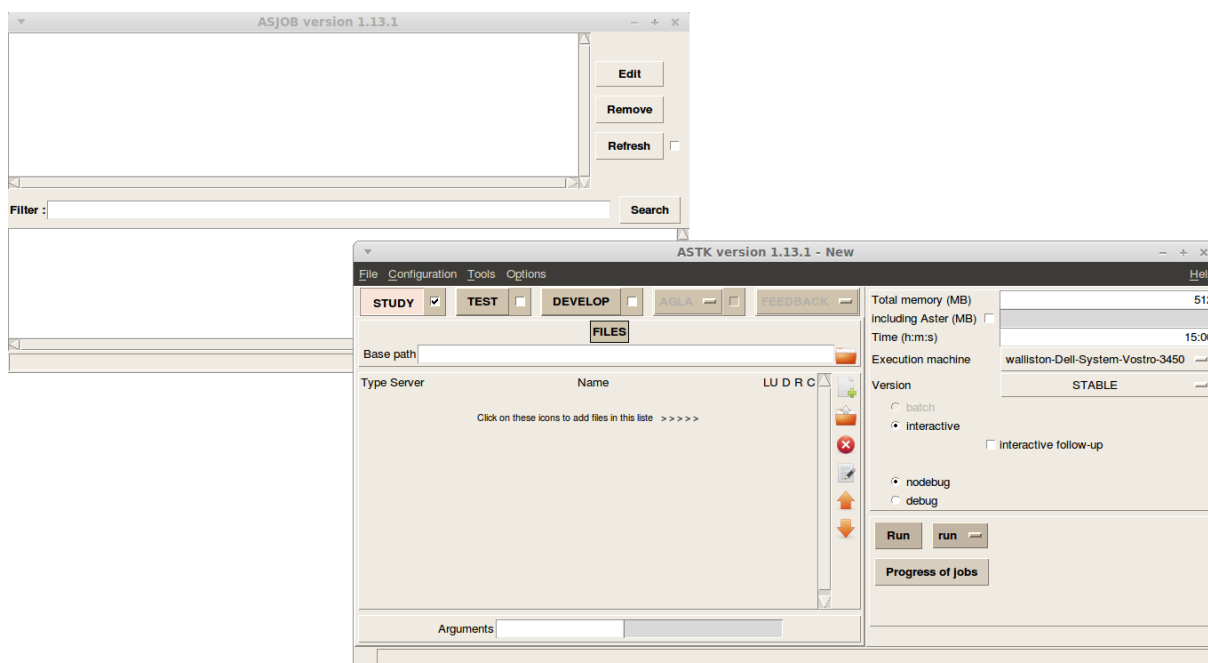


Figura B.5 – Code-Aster[®], *software open source* de análise estrutural

Além do seu arquivo principal de entrada (“*.comm”), o Code-Aster[®] necessita de um arquivo de malha também como entrada para a realização da análise (geralmente em formato “*.mmed”). Como saída, o programa basicamente fornece os seguintes arquivos:

- “*.mess”, que fornece todos os procedimentos utilizados pelo programa durante a análise, sendo assim a saída do solver, contendo mensagens, erros e avisos;
- “*.resu”, que fornece todos os resultados da análise em um formato de tabela;
- “*.rmed”, que fornece a malha final com os respectivos dados de saída, sendo este o arquivo utilizado pelo módulo ParaVis (descrito a seguir) para a visualização gráfica do resultado do modelo.
- “base”, sendo, na verdade, um diretório que contém informações importantes sobre a saída do *solver*, sendo utilizado para uma possível continuação da análise caso necessário.

Cada tipo de arquivo acima listado possui um identificador associado a ele, ou seja, um número. Como o Code-Aster[®] é oriundo do Fortran[®], estes números representam exatamente

os arquivos temporários “fort” que são gerados ao longo da execução de um programa qualquer, sendo que “fort.1”, por exemplo nesse caso, é o arquivo de identificador 1 definido automaticamente pelo Code-Aster®, ou mesmo pelo usuário. Uma vez que esses identificadores podem ser alterados pelo usuário eles devem obrigatoriamente serem diferentes entre si. Os identificadores padrão do Code-Aster® podem ser vistos na coluna LU da Fig. B.6. Ao seu lado existem as colunas D (que remetem à entrada), R (que remetem à saída) e C (que remetem aos arquivos comprimidos do tipo “base”, descrito acima). Todas estas opções são definidas na tela principal do programa.

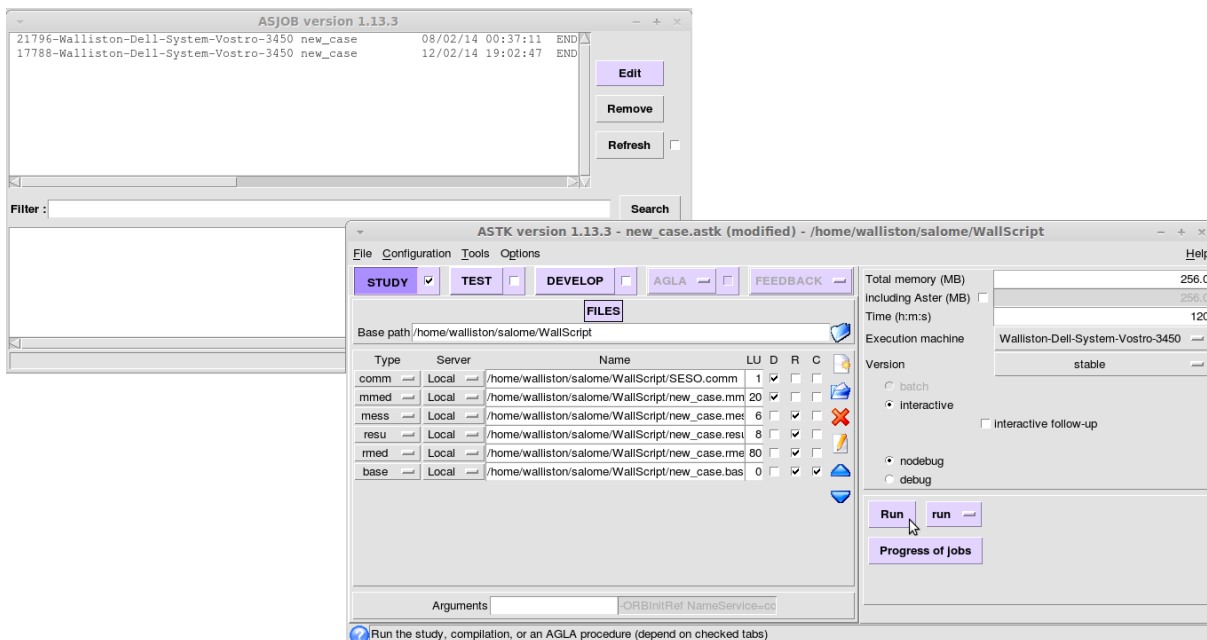


Figura B.6 – Definições dos tipos de arquivos de saída definidos pelo usuário no Code-Aster®

Além disso, o programa ainda cria um arquivo texto padrão no formato ASTK que descreve todas as definições feitas pelo usuário em relação ao modelo que deseja analisar. A grande vantagem desse arquivo é que ele é, na verdade, uma interface capaz de interpretar o modelo em qualquer computador que possua o programa Code-Aster®, incluindo via internet. Assim, os usuários podem facilmente distribuir seus arquivos de dados e resultados por diferentes máquinas garantindo, por intermédio desta interface, a transferência dos arquivos, incluindo os compactados, através da rede. Além disso, de acordo com os próprios desenvolvedores, eles disponibilizam a todos os usuários do *software* um servidor central, denominado Bull Novascale, que possui 120 processadores (1.6GHz e 960GB de memória RAM) para o auxílio

do cálculo dos modelos de forma remota, localizado no Research and Technology Calculation Centre (Centro de Pesquisa e Tecnologia de Cálculo, ou CCRT, em francês).

Finalmente, a licença do programa é regida pela GNU GPL (GNU General Public License).

B.5 EFICAS[®]

O editor de arquivos de comando (macro-rotinas) e verificador de sintaxe EFICAS[®] é um importante *software open source*, ou código aberto, com interface gráfica amigável que garante a criação e edição de arquivos de entrada para o Code Aster[®], com sintaxe correta de acordo com as escolhas do usuário. Essa sintaxe é ajustada através de um “catálogo de comandos” que o programa possui, sendo interpretados por palavras-chave e seguindo algumas regras e conceitos predefinidos.

O EFICAS[®] é um programa francês e seu nome significa *Editeur de Fichier de Commandes et Analyseur Sémantique*. Em outras palavras, o programa permite aos usuários editarem arquivos que serão interpretados como códigos, evitando assim erros de comando que não são permitidos de acordo com um determinado contexto, garantindo dessa forma a integridade do arquivo.

O programa EFICAS[®] pode usar vários comandos e lidar com várias versões em cada código. É composto por arquivos denominados “catálogos”, que contém todas as macro-rotinas para um código. Cada macro-rotina possui nome e parâmetros definidos pelos desenvolvedores. O arquivo de saída do EFICAS[®] possui a extensão “*.comm”, é um arquivo de texto simples e este é o único formato aceito para releitura pelo programa.

Tanto os arquivos de comando quanto os catálogos são arquivos Python[®], apesar do EFICAS[®] propriamente dito não entender isoladamente vários comandos nativos da linguagem de programação de origem de suas extensões. Isso significa que os arquivos “*.comm” podem ter trecho com códigos em Python[®] para serem usados pelo Code-Aster[®], apesar do próprio EFICAS[®] não interpretar os comandos e sendo necessário, portanto, comentá-los para o

reconhecimento do arquivo final. No entanto, as regras de sintaxe para a utilização dos códigos em Python[®] são as seguintes:

- ◆ Nomes de variáveis e identificadores iniciam-se com letras (A_Z ou a-z) ou sublinhado (“_”), sendo seguidos por letras, números ou sublinhados. O EFICAS[®] é “case-sensitive” (letras maiúsculas diferem de minúsculas) e possui uma limitação natural de 8 caracteres para cada nome. Uma *string* é definida por quotes simples ou duplos.
- ◆ Algumas palavras estão reservadas para o EFICAS[®], a saber: BLOC, EXCLUS, OPER, ASSD, EVAL, FACT, FORM, JDC, REGLE, VALIDATOR, ETAPE, ENTITE.
- ◆ Os arquivos de saída (“*.comm”) são arquivos Python[®], o que significa que a forma de atribuição de variáveis é “variável = valor”; um comentário é iniciado usando “#”; as tuplas (sequência ordenada e imutável de valores) são definidas por parênteses; as listas são definidas por colchetes; e as vírgulas “,” separam cada item das tuplas ou listas.

B.5.1 Passo a passo para um novo JDC (“*Jeu De Commande*”)

O EFICAS[®] é um programa simples, à primeira vista. Não há muitos comandos em sua janela principal. Quando o mesmo é iniciado pelo Salome-Meca[®], que é uma plataforma gráfica integrada ao núcleo do *software* Code-Aster[®], e é descrita a seguir, existe a possibilidade de escolher entre a versão específica para compatibilidade com o Code-Aster[®] ou a versão genérica que envolve uma concordância com outras plataformas, como o OpenTURNS[®]. Ao se iniciar a versão para Code-Aster[®] uma janela informa a quantidade e a versão dos catálogos que pode ser escolhido pelo usuário, como mostra a Fig. B.7.

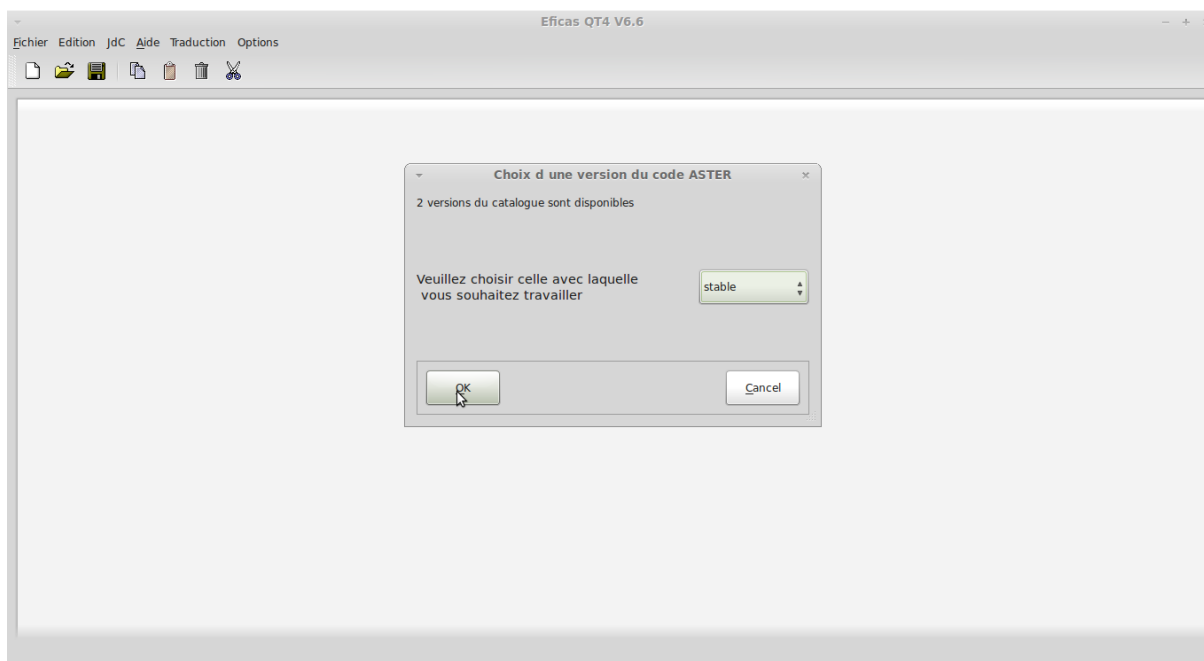


Figura B.7 – EFICAS[®], editor *open source* de arquivos de comando (macro-rotinas)

O JdC equivale a um novo projeto do programa e, ao se iniciar este projeto, a janela principal do EFICAS[®] fornece três vistas diferentes, conforme apresentado na Fig. B.8. Estas três vistas podem ser resumidas nos seguintes itens:

- O lado esquerdo contém as macro-rotinas já escolhidas pelo usuário, ou seja, os comandos que irão compor o arquivo final de saída. Uma interessante característica é que essa sessão possui sinalizadores que indicam se o arquivo final estará correto ou não, ou seja, figuras indicativas, ao lado de cada macro-rotina escolhida, ficam verdes quando está tudo certo, amarelas quando faltam dados dos parâmetros e vermelhas quando algo está errado;
- A parte central contém todas as informações e macro-rotinas que o usuário pode utilizar para gerar seu arquivo “*.comm”, já formatado de acordo com as regras seguidas pelo programa e em formato hierárquico de estrutura de dados. Existe a opção de procurar uma macro-rotina específica em ordem alfabética ou por grupo;
- O lado direito são as regras que devem ser seguidas em relação às macro-rotinas escolhidas. As regras não verificadas são alteradas para fonte em vermelho e as regras já cumpridas permanecem com a fonte preta. São elas:

- 1) AU_MOINS_UN: obriga o usuário a criar pelo menos um conceito de uma lista, predefinida pelo programa. Mais de um conceito pode ser criado;
- 2) UN_PARM: obriga o usuário a criar um e apenas um conceito de uma lista;
- 3) EXCLUS: significa que se uma palavra-chave é criada, uma ou mais palavras-chave podem não ser permitidas;
- 4) ENSEMBLE: a regra significa que se uma palavra-chave é selecionada, uma ou mais palavras-chave também devem ser, não importando a ordem;
- 5) PRESENT_PRESENT: a regra significa que se a primeira palavra-chave for selecionada, todas as outras também devem ser;
- 6) PRESENT_ABSENT: a regra significa que se a primeira palavra-chave for selecionada, nenhuma outra poderá ser.

Além disso, em um JDC podem ser encontrados os seguintes tipos de comandos:

- ✓ PROC, sendo este um comando que contém todas as suas palavras-chave;
- ✓ OPER, sendo este um comando que cria um objeto que deve possuir um tipo e um nome obrigatoriamente (como exemplos, pode ser um objeto do tipo malha, do tipo resultado, do tipo material...);
- ✓ MOT-CLEF FACTEUR (fator palavra-chave), sendo esta uma lista de palavras-chave que possuem de forma conjunta um significado para o código. Algumas são obrigatórias e podem depender das regras condicionais.
- ✓ MOT-CLEF SIMPLE (palavra-chave única), sendo esta uma única palavra que pode ou não ser obrigatória para o comando.

Todos estes itens estão associados a várias regras e condições, e são descritos no catálogo (Fig. B.8).

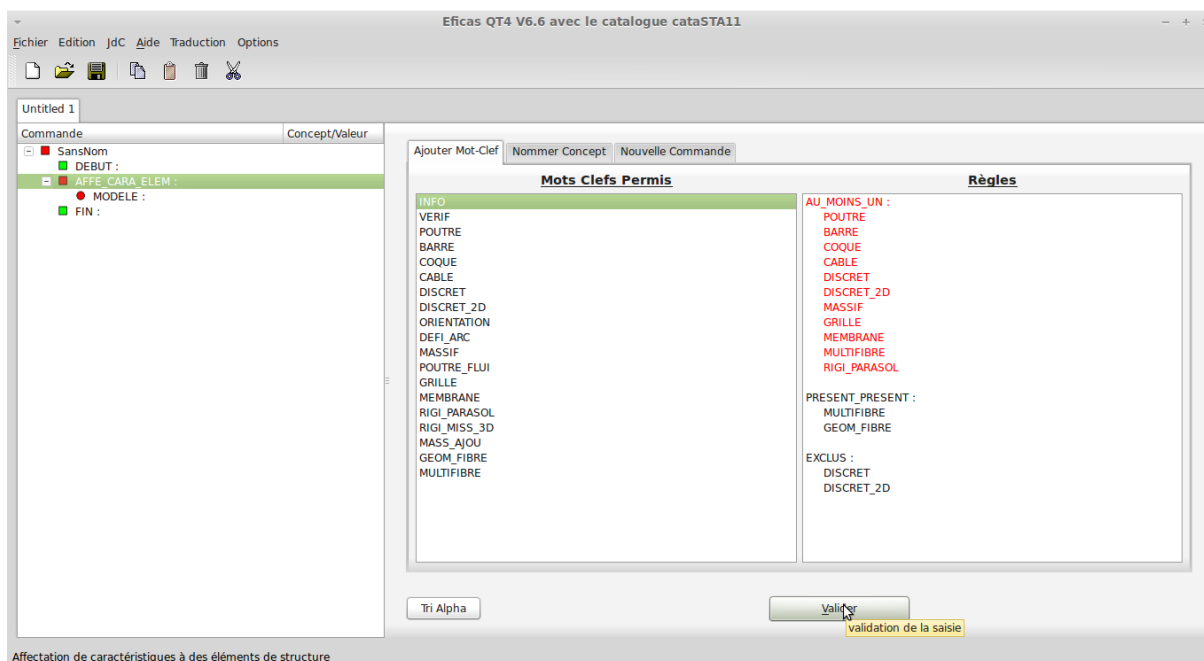


Figura B.8 – Ambiente JdC, com as respectivas vistas, do programa EFICAS®

Como já mencionado anteriormente, um simples catálogo de cores ajuda o usuário a saber se um item é válido ou não. Além do quadrado, círculos e losangos compõem as figuras que indicam erro ou não nas atividades do usuário, de acordo com as respectivas cores, assim descrito (Fig. B.9):

- Quadrados verdes significam objetos válidos (Opers, JdCs,...);
- Quadrados vermelhos significam objetos inválidos;
- Quadrados amarelos significam que os objetos OPER são válidos mas precisam de um nome;
- Círculos verdes significam palavras-chave corretas;
- Círculos vermelhos significam palavras-chave incorretas;
- Losangos verdes é uma condição que significa que todas as palavras-chave dependentes estão corretas;
- Losangos amarelos é uma condição que significa que uma ou mais palavras-chave dependentes são inválidas;

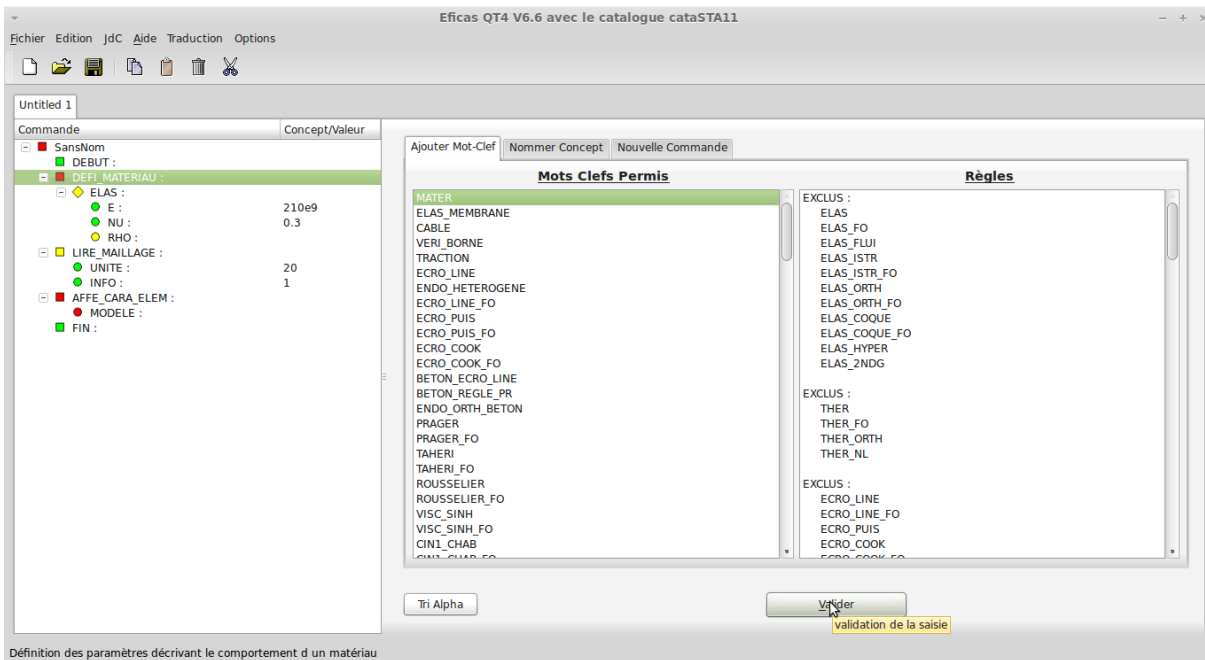


Figura B.9 – Representação simbólica das notificações do EFICAS®

As palavras-chave geralmente são parâmetros que o usuário pode definir. O programa EFICAS® aceita como parâmetro *string*, *real*, *integer* e *list*. É permitido ao usuário tanto digitar esses valores diretamente nos campos adequados quanto importá-los de arquivos-texto ASCII, selecionando todo o conteúdo do arquivo-texto ou apenas uma parte dele. O programa ainda permite reportar erros e trabalhar com vários arquivos JDC ao mesmo tempo, em diferentes abas e com catálogos distintos (ou em um mesmo catálogo) caso o usuário desejar, além de permitir também que se possa abrir o mesmo arquivo em abas diferentes.

Assim como vários *softwares* livres, várias pessoas colaboraram e ainda contribuem bastante para o projeto. O programa EFICAS® foi desenvolvido pela EDF-R&D™ (Électricité de France - Research and Development), a equipe do *software* Code-Aster®, a equipe do CAELinux™ e, além disso, contando com a colaboração de Jason Cauley que realizou a tradução do site do projeto do francês para o inglês para que aqueles que não sabem francês pudessem ter contato com a comunidade do Code-Aster para solucionar dúvidas, além de ajudar a divulgar o programa internacionalmente. Além disso, a licença do programa EFICAS® é regida pela GNU GPL (GNU General Public License).

B.6 Salome®

Salome® é um *software* de código livre com interface gráfica amigável que fornece uma plataforma genérica para pré e pós processamentos de simulações numéricas. É baseado na tecnologia CORBA (Common Object Request Broker Architecture), sendo este um padrão desenvolvido pelo Object Management Group (OMG) que permite a troca de informações, independente de plataformas de hardware, linguagens de programação e sistemas operacionais. Assim, componentes de aplicativos podem se comunicar uns com os outros, não importando onde eles estão localizados ou quem os projetou, sendo uma tecnologia que fornece a transparência de localização para ser capaz de executar estas aplicações.

A modelagem numérica de um problema físico, o desenvolvimento de soluções e sua integração no processo de design são muitas vezes controlados em ambientes muito específicos. Por isso, Salome® representa uma aplicação baseada em CAD genérico para simulação digital com “alta reatividade às evoluções do mercado e as expectativas dos usuários”. O programa é bem interessante, pois é baseado em uma arquitetura aberta e flexível construída a partir de componentes “reutilizáveis” disponíveis como *software* livre, ou seja, códigos abertos oriundos de outros programas.

O programa fornece ferramentas para a construção de aplicações complexas e integradas, contendo uma bela interface gráfica que permite a integração com os módulos de CAD, a geração de malha de modelos CAD, a definição de arquivos de entrada, a junção de códigos desenvolvidos e/ou importados de outros programas e a visualização dos resultados. Ainda, o programa é usado em pesquisa nuclear e estudos na área industrial pelo CEA™ e EDF™ no que diz respeito a reatores nucleares, mecânica estrutural, termo hidráulica, combustíveis nucleares, ciência dos materiais, geologia e simulação de gestão de resíduos, eletromagnetismo e radiação.

Como o Salome® é uma plataforma CAD/CAE (Computer Aided Design/Computer Aided Engineering), o usuário é capaz de realizar modelagens em 3D, visualizações de modelos importados de outras plataformas (de acordo com as extensões permitidas pelo programa) e pós processamento, tudo isso a partir de componentes “reutilizados”. Além disso, é permitido

ao usuário a geração de malhas no próprio programa e a exportação de arquivos para a resolução de modelos numéricos (*solver*) via elementos finitos a partir de preprocessadores específicos, de acordo com adaptações feitas através de determinados componentes. E, finalmente, todas as funcionalidades são acessíveis através do console Python integrado, localizado na parte inferior da janela principal, que permite a inserção de códigos da linguagem para a realização dos comandos do programa.

Em outras palavras, o Salome® é constituído de vários módulos (ou “subprogramas”) que realizam suas funções de forma independente, mas que estão interligados entre si (Fig. B.10). Assim, o usuário realiza sua análise etapa a etapa, o que, além de proporcionar o entendimento do programa mais fácil, diminui as possibilidades de erro na modelagem geométrica.

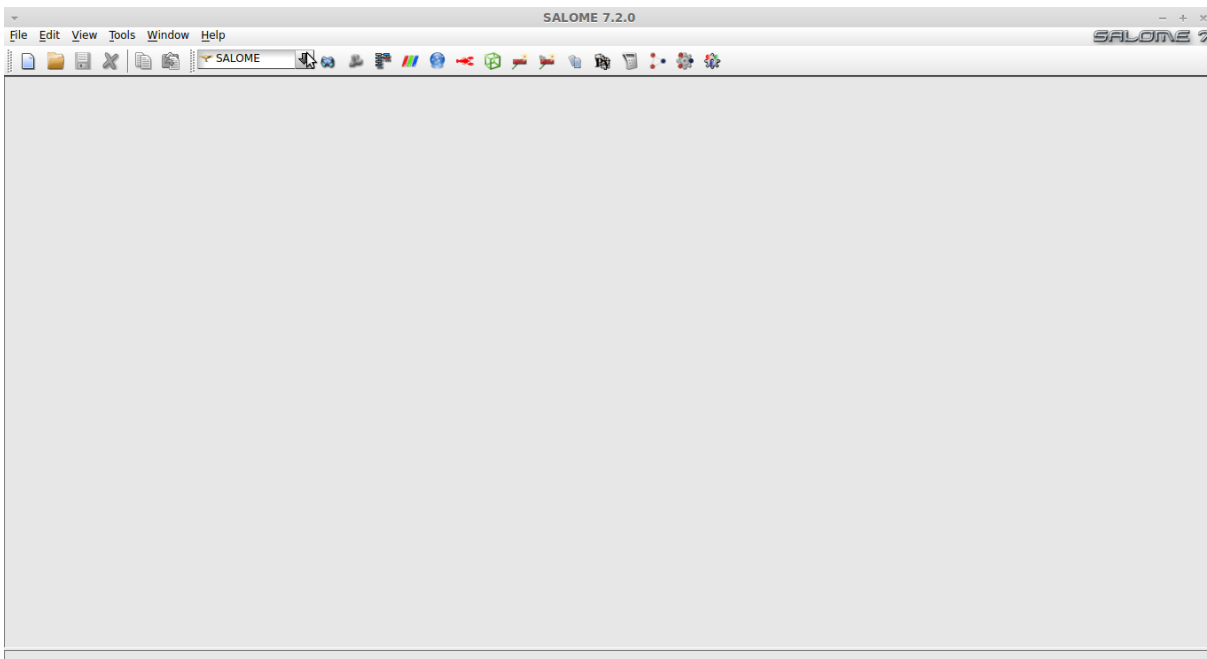


Figura B.10 – Salome®, plataforma genérica *open source* de pré e pós-processamento

O Salome® existe desde 2001 e foi desenvolvido pela CEA™ (Commissariat à l’Energie Atomique et aux Energies Alternatives) e EDF-R&D™ (Électricité de France - Research and Development). Está em constante atualização, contando com o apoio da Euriware™/Open Cascade® com 10 anos de intenso desenvolvimento. Além disso, a licença do programa é regida pela GNU GPL (GNU General Public License).

B.7 Salome-Meca®

Salome-Meca® é um *software open source*, ou código aberto, que reúne alguns componentes específicos e fundamentais para simulações numéricas via Método dos Elementos Finitos. Contém o programa Salome® (pré e pós processador) e o Code-Aster® (solver) em um mesmo pacote, necessários para a definição e desenho de um modelo inicial, o cálculo do mesmo e a apresentação dos resultados. A interface gráfica é praticamente idêntica à do Salome®, mas alguns componentes presentes no Salome-Meca® são exclusivos para este programa (Fig. B.11).

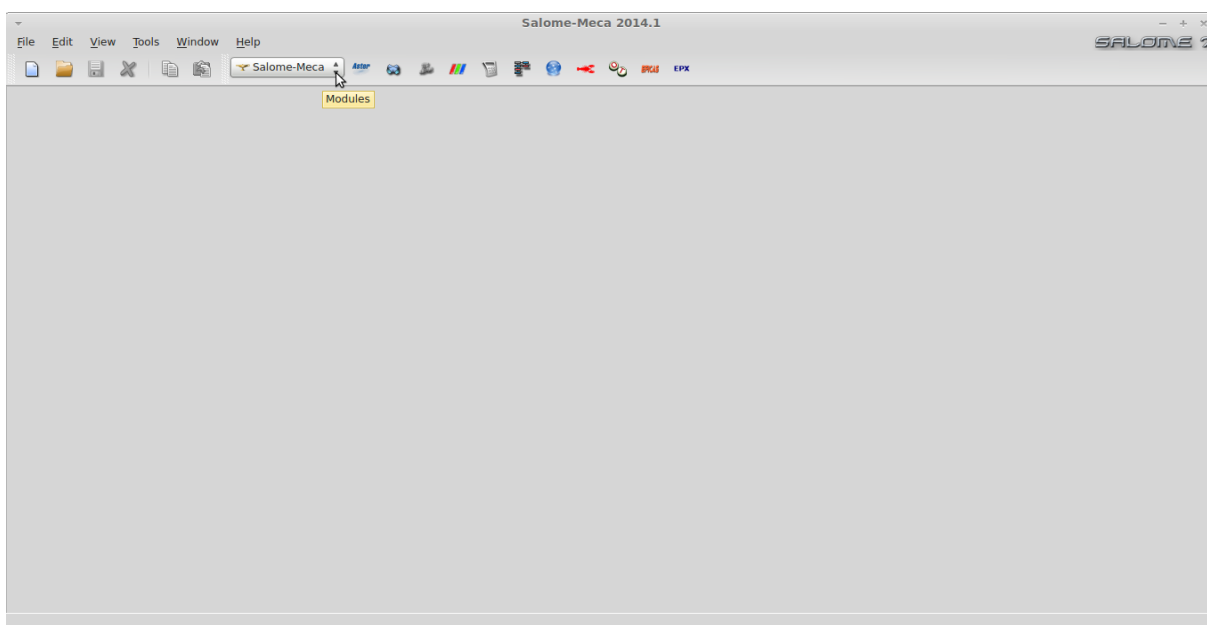


Figura B.11 – Salome-Meca®, programa *open source* completo de análise estrutural via MEF

A Fig. B.11 acima mostra a janela principal do programa, com os módulos que o compõem. Há uma caixa na parte superior da janela onde é permitido escolher carregar o módulo desejado. Alguns deles se encontram ao lado da caixa, para facilitar a seleção. São eles:

- *Aster*, que carrega o programa Code-Aster® em separado;
- *Geometry*, que carrega um módulo onde é possível importar e exportar desenhos geométricos em alguns formatos específicos, além de fazer e modificar desenhos geométricos planos e espaciais;

- *Mesh*, que carrega um módulo onde é possível importar, exportar, gerar e editar malhas em vários formatos;
- *ParaVis*, que carrega um módulo de visualização de resultados;
- *MED*, que carrega um módulo exclusivo de edição de malhas, seguindo um padrão para o armazenamento e a recuperação de dados computacionais associados às malhas numéricas e facilitando a interação entre os códigos e os *solvers*;
- *YACS*, que carrega um módulo que permite a construção, edição e execução de planos de cálculo que, por sua vez, define uma cadeia ou um acoplamento de códigos de computador;
- *JobManager*, que carrega um módulo que permite a criação, execução e acompanhamento de comandos de cálculo em diferentes tipos de computadores;
- *Homard*, que carrega um módulo destinado a adaptar as malhas de elementos finitos ou volumes finitos (2D/3D) através de técnicas de refinamento ou de seu oposto (“engrossamento”). Foi criado pela EDF-R&D™ (*Électricité de France - Research and Development*) e feito para ser utilizado de forma independente do código de computador ao qual está associado;
- *OpenTURNS*, que carrega o programa OpenTURNS® em separado, sendo este uma biblioteca científica utilizável como um módulo Python® dedicado para o tratamento de incertezas;
- *Eficas*, que carrega o programa EFICAS® em separado;
- *Europlexus*, que carrega um módulo do programa EuroPleXus® (EPX), sendo este um simulador de análise de fenômenos transientes envolvendo a interação fluido-estrutura. Foi desenvolvido pela CEA™ (*Commissariat à l’Energie Atomique et aux Energies Alternatives*) e pela Joint Research Centre da Comissão Européia.

Os módulos são carregados individualmente, apesar de estarem interligados. Na lateral esquerda da janela principal há a possibilidade de acompanhar o projeto iniciado ao longo de cada módulo já utilizado. Os módulos são separados em formato “*tree view*”, ou seja, os dados são dispostos em formato hierárquico mostrando todas as etapas concluídas do projeto até o momento. Porém, apenas os dados referentes ao módulo carregado ficam disponíveis para alterações e/ou visualizações.

B.7.1 Realizando uma simulação numérica de um modelo

Primeiramente, importa-se um projeto já iniciado do Salome[®], ou se importa um arquivo de desenho criado a partir de outro *software* (como FreeCAD[®], AutoCAD[®], entre outros), ou se inicia um novo desenho em *Geometry*, ou simplesmente cria-se um desenho via linguagem de programação Python. Serão definidos junto com o modelo (importado ou criado) alguns grupos que conterão informações importantes, como sobre condições iniciais e de contorno, ou ainda, caso o usuário opte por definir vários materiais ou discretizar o modelo com malhas diferentes, cada grupo conterá as devidas informações. Ao menos um grupo precisa ser definido nesta etapa, sendo que eles poderão ser definidos de forma pontual, linear, plana ou espacial, ou seja, ou uma face da estrutura ou um segmento de um dos lados ou algum vértice ou mesmo toda a estrutura são exemplos de informações que estarão contidas em um ou mais grupos definidos. A qualquer momento é permitida a inclusão ou a remoção de grupos. É importante lembrar que qualquer intervenção pode ser feita via linguagem de programação, ou seja, um *script* pode executar uma ação externa que irá sobrepor as configurações atuais (Fig. B.12).

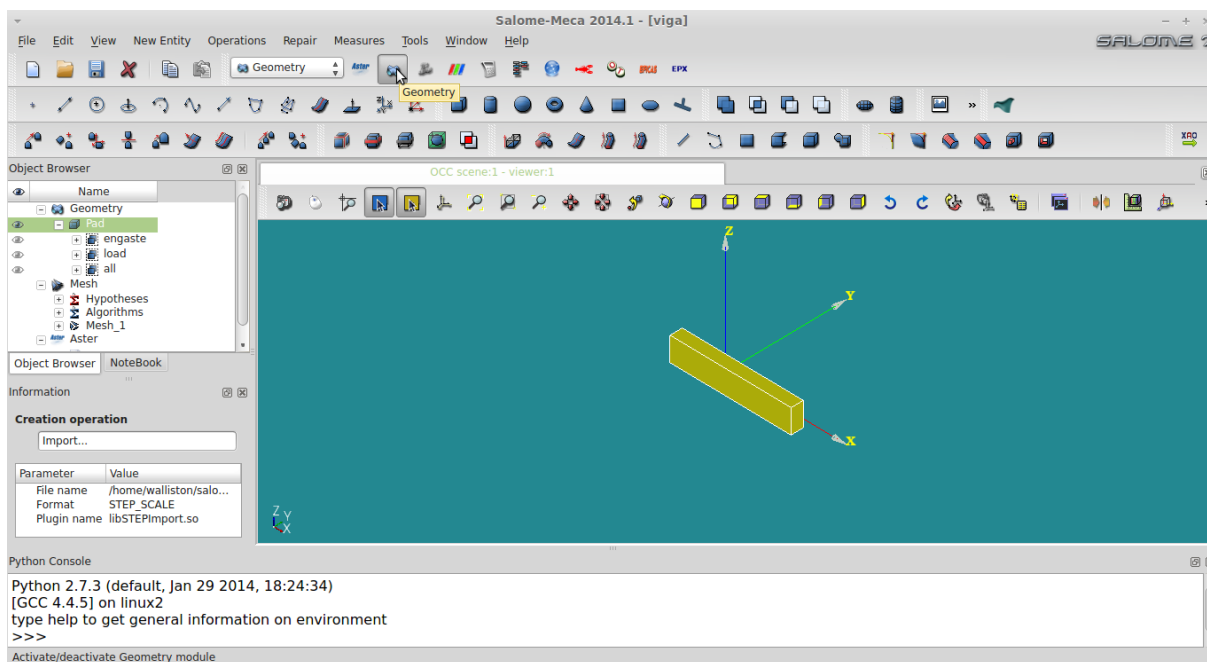


Figura B.12 – Módulo de geometria do programa Salome-Meca[®] (Geometry)

Em seguida, a malha deve ser criada. Aqui serão definidos os elementos finitos em todas as

dimensões, ou seja, elementos lineares, planos e espaciais. O programa ainda permite a definição de hipóteses, em relação às malhas, que representam as condições de contorno que serão levadas em conta no cálculo de malhas ou “sub-malhas”, isto é, grupos ou partes da malha discretizada em todo o domínio. Essas hipóteses permitem que se gerencie o nível de detalhe das malhas ou “sub-malhas” resultantes. Ao aplicar diferentes hipóteses com diferentes parâmetros, é permitido programar a quantidade ou tamanho dos elementos que compõem a malha, possibilitando gerar uma malha ou “sub-malha” mais grosseira ou mais refinada. Por exemplo, para o domínio unidimensional e levando em conta as bordas ou arestas da malha, tem-se como hipóteses o número de segmentos, o comprimento local e o tamanho máximo. Para o bidimensional e tridimensional, levando em conta as faces e volumes da malha respectivamente, tem-se como hipóteses o número máximo de elementos por área e por volume, respectivamente. Há também a possibilidade da criação de grupos de nós e/ou elementos, inclusive associados ou não aos grupos definidos na geometria (fig. B.13).

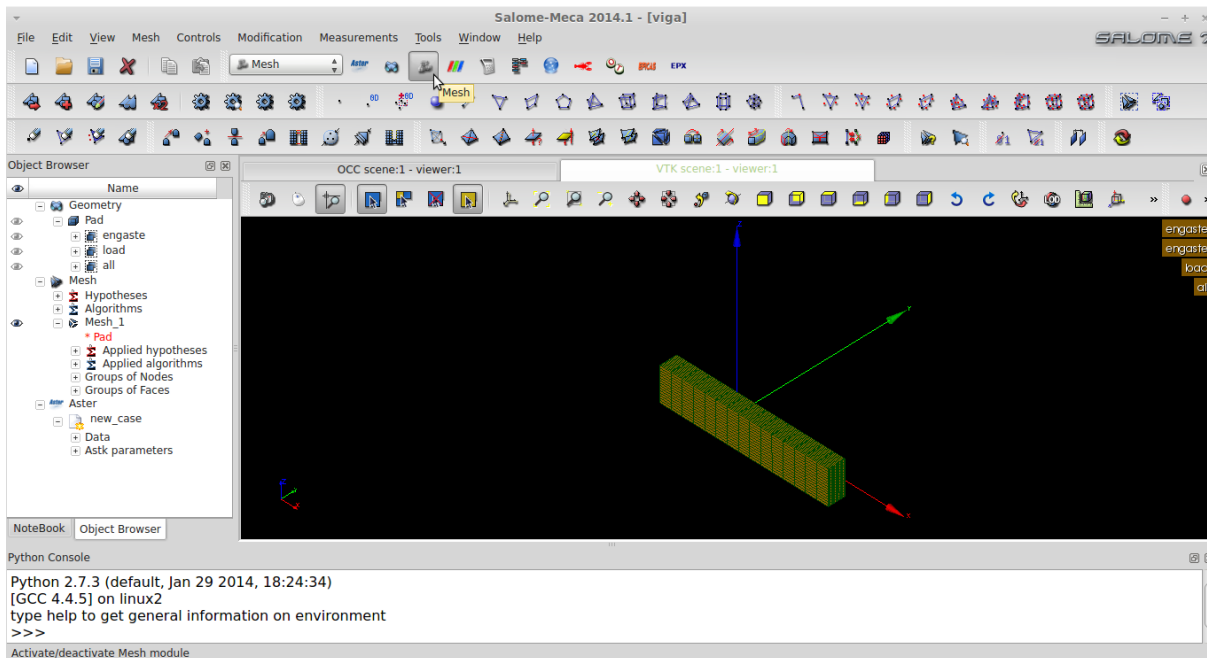


Figura B.13 – Módulo de geração de malha do programa Salome-Meca® (Mesh)

Logo após o modelo deve ser exportado para o Code Aster®, caso se tenha um arquivo prévio de entrada para o mesmo (“*.comm”). Caso contrário, basta usar uma ferramenta do programa chamada “wizard”, sendo associada a um gerador simplificado do arquivo de entrada e que

permite escolher entre análise linear elástica, modal, linear térmica e fratura. Convém ressaltar que, por ser simplificado, a análise poderá ser muito restrita, impossibilitando várias verificações. Por isso, é interessante utilizar o programa EFICAS[®] para a criação do arquivo.

Após a exportação do modelo com o devido arquivo de entrada, o diretório, onde serão salvos os arquivos de saída, deve ser definido, assim como o diretório do arquivo da malha (“*.mmed”). É permitido ao usuário definir a versão do Code-Aster[®], além de um tempo limite para a análise, em segundos, e a quantidade de memória total que será utilizada para esta finalidade. Todo o processo pode ser acompanhado em uma janela secundária, que lista o tempo total atual utilizado, assim como possíveis erros que possam ocorrer (Fig. B.14).

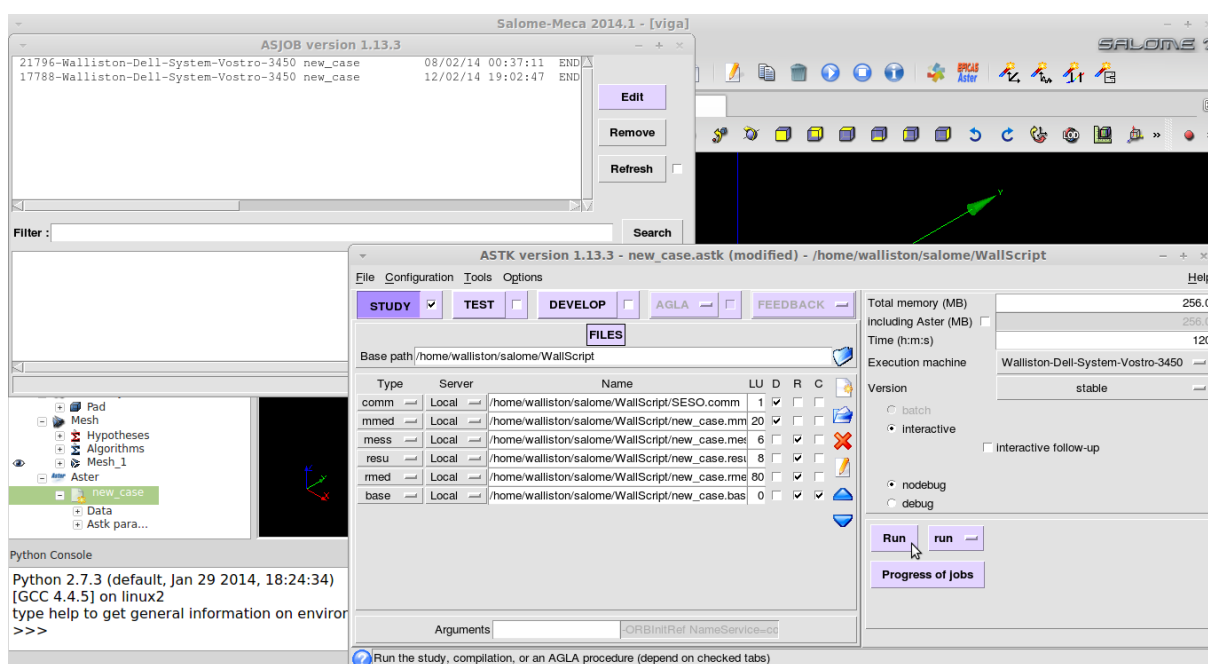


Figura B.14 – Módulo de cálculo estrutural do programa Salome-Meca[®] (Aster)

Finalmente, os resultados podem ser vistos no *ParaVis* importando o arquivo de saída da malha (“*.rmed”). Existe ainda a possibilidade de visualizar vários resultados por vez, além da escolha de respostas, como tensões principais, deslocamentos, deformações, entre outros (Fig. B.15).

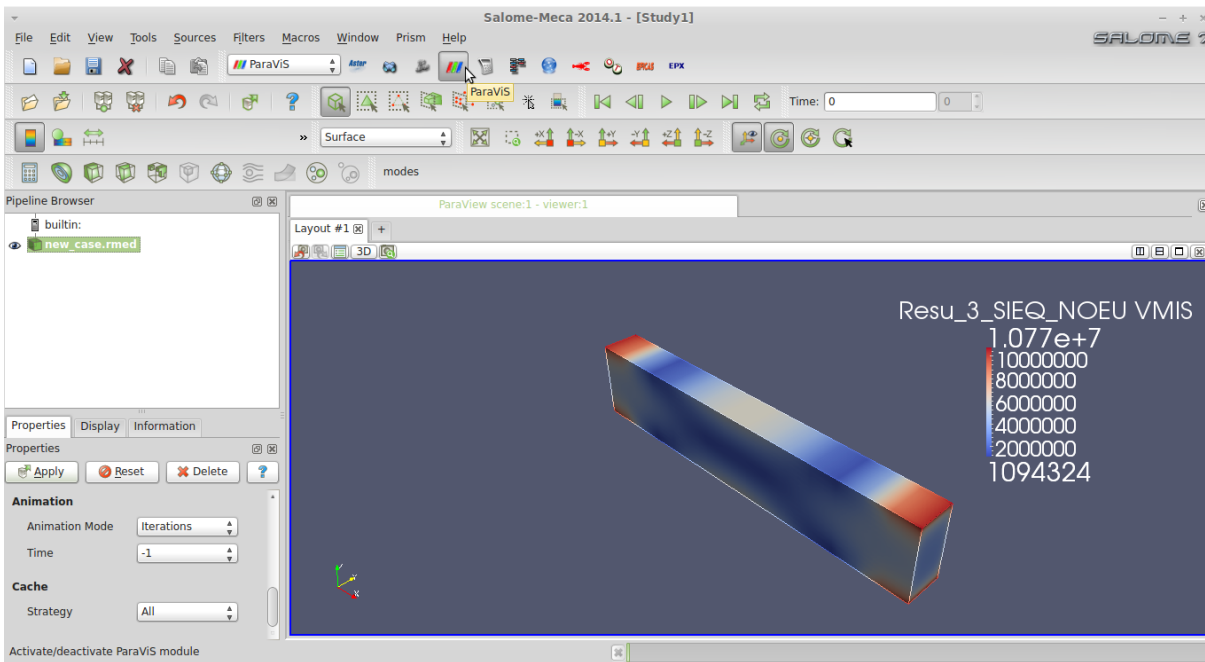


Figura B.15 – Módulo de visualização do programa Salome-Meca® (ParaVis)

Assim como o Salome®, o Salome-Meca® foi desenvolvido pela EDF-R&D™ (*Électricité de France - Research and Development*). É um projeto que existe desde 2007 e a licença do programa é regida pela GNU LGPL (GNU Lesser General Public License versão 2.1).

B.8 O Programa UniOM (Unidades para a Otimização de Modelos)

O programa *UniOM v1.1* foi desenvolvido em Lazarus® para o auxílio na análise dos resultados. Basicamente, o programa serve para unir as rotinas de otimização escritas em Python® às macro-rotinas do arquivo “*.comm” de entrada do Code-Aster®, evitando edições manuais. Aqui também são definidos os valores dos parâmetros de otimização que serão utilizados na análise. Assim, em outras palavras, o programa nada mais é do que um gerador de um arquivo “*.comm” completo, incluindo todas as definições e atribuições necessárias para a otimização. É importante ressaltar que um arquivo prévio com as macro-rotinas necessárias para a análise em questão já deve ter sido criado previamente. A Fig. B.16 mostra a tela inicial e principal do programa.

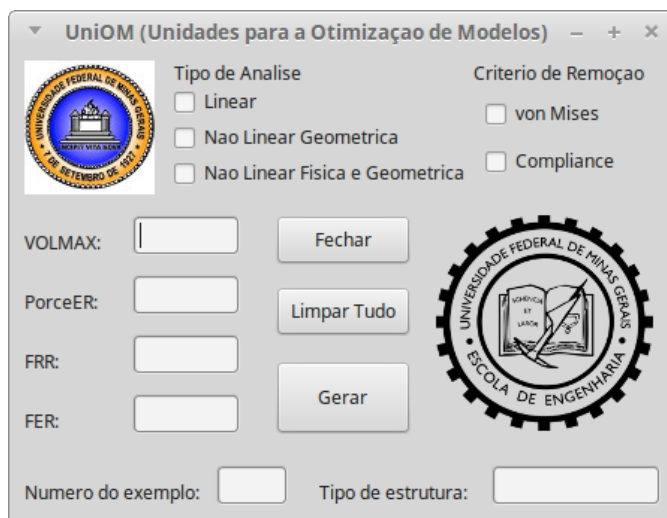


Figura B.16 – Programa UniOM (Unidades para a Otimização de Modelos)

Os parâmetros de otimização são definidos nessa tela e, ao pressionar o botão “Gerar”, o arquivo-texto é criado. É necessário informar um número para o exemplo e o tipo de estrutura (chapa, placa, entre outros). Além do tipo de análise estática (linear, não linear geométrica, não linear física e geométrica), o critério de remoção também pode ser escolhido entre Tensão de von Mises e o *Compliance*, embora o último não tenha sido estudado nessa Tese.

B.9 Estrutura de Dados e Pseudocódigo

Para proceder com a otimização por meio de uma intervenção no Salome-Meca[®], um trecho de código foi inserido no arquivo principal de entrada do Code-Aster[®], em linguagem Python[®]. Esse código age em paralelo com a resolução do modelo proposto, feita pelo programa, correspondendo a um laço que abrange uma análise estrutural completa a cada iteração. De fato, o que acontece é que, a cada iteração, as respostas anteriores servem de base para as atualizações seguintes, tanto em questão de comparação, dados parâmetros anteriores, quanto em questão de modificação, de acordo com o modelo anterior. A comparação permite verificar se o critério de remoção foi atendido, ao passo que a modificação garante a execução do critério satisfeito através da alteração da topologia parcial que é oriunda da iteração anterior. Em outras palavras, os elementos são classificados e removidos a cada iteração até a obtenção da topologia ótima.

Primeiramente, os dados do modelo são coletados e armazenados. Assim, a geometria do modelo é definida com os parâmetros do material, que são informados pelo usuário. O problema é discretizado e, em seguida, são definidos todos os grupos pertencentes a ela, sendo estes associados à geometria. Logo após, o arquivo de entrada para a análise é desenvolvido, contendo também as condições de contorno e carregamento. Além disso, os parâmetros de otimização também são definidos aqui e são escolhidos de acordo com o tipo de problema para que a topologia final seja obtida (Fig. B.17).

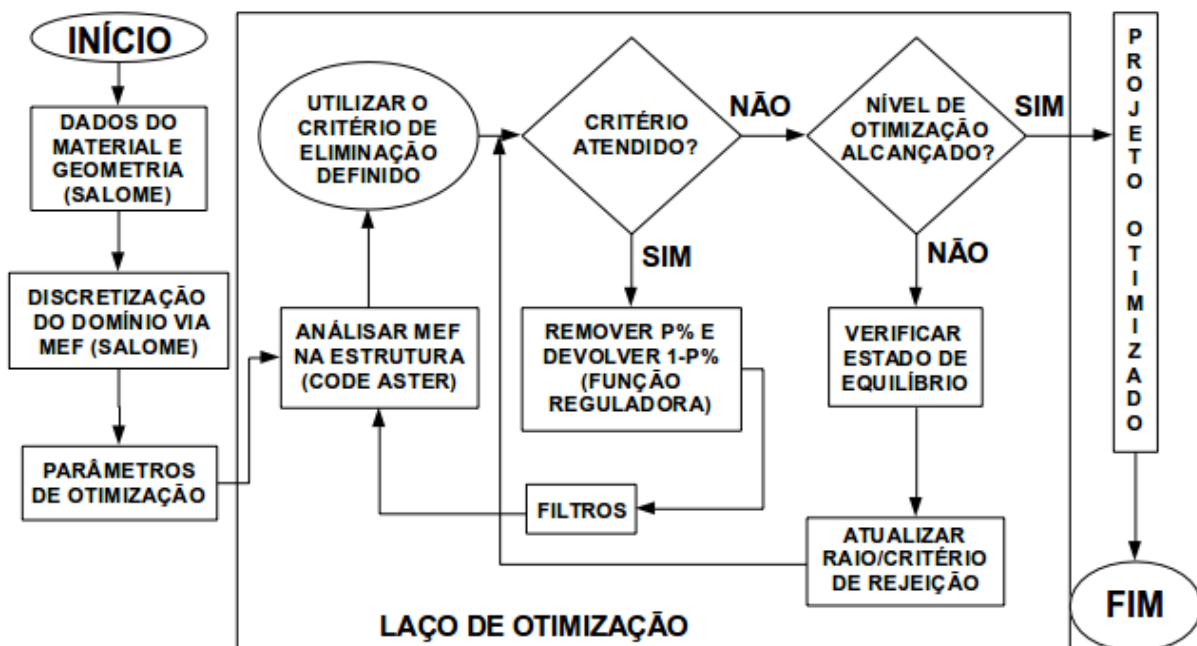


Figura B.17: Fluxograma de otimização do SESO

Após definir os parâmetros de entrada, a malha de elementos é gerada. Assim, os parâmetros de otimização precisam ser estabelecidos para se ter o início do processo iterativo. A malha é verificada e inicia-se o critério de remoção através da Tensão Máxima de von Mises. A porcentagem de elementos removidos por iteração depende dos parâmetros de calibração definidos antes do início da avaliação do modelo. Se o critério for atendido, ou seja, a inequação que verifica e compara a Tensão Máxima de von Mises com a Tensão de von Mises do elemento for satisfeita, ocorre a remoção e a devolução de elementos, de acordo com as funções reguladoras e filtros. Esse processo é feito n vezes até que o critério não seja mais atendido. Se ainda assim o nível de otimização não foi alcançado, o estado de equilíbrio da estrutura é verificado, de acordo com o índice de performance, e o fator de raio de rejeição é

atualizado, para novamente o processo de remoção de elementos ser iniciado até que o critério de remoção não seja atendido e o nível de otimização seja alcançado. Dessa forma, obtêm-se a estrutura final, já otimizada, com os respectivos resultados.

O código feito em Python[®] conta atualmente com mais de 500 linhas e ainda está em processo de desenvolvimento e adaptação. É dividido em três partes, sendo a primeira parte referente às definições das funções Python[®] de otimização, a segunda parte referente às macro-rotinas do Code-Aster[®] e a terceira parte a chamada das funções em conjunto com os resultados da análise, englobados pelo laço de otimização.

```

#####
#
#####
# SESO (versão importada para Python)
# Readaptado de Fortran por Walliston Fernandes
# Usavel em pacotes EDF/Code Aster, Salome-Meca
# Programa de Pos Graduacao em Engenharia de Estruturas da UFG (Doutorado)
# Iniciado em 29/08/2013
# Ultima vez editado em 03/10/2014

#####
#
#####
# INICIO DE OBSERVACOES IMPORTANTES

# Arrays alocaveis devem ser declarados como "None do_array = [None]*variavel";
# Arrays possuem propriedades especificas, como sort(), reverse(), entre outras;

# Versao 1.7 inclui: taxa de otimizacao, criterio de parada de "tensao ultima/escoamento" e FRR maximo.
# Versao 1.8 inclui: Versao 1.7 + otimizacao multiobjetivo (volume, tensao, deslocamento e IP)
# Versao 1.9 inclui: Versao 1.8 + AHP (Analytic Hierarchy Process)

# FIM DE OBSERVACOES IMPORTANTES

#####
#
#####
# INICIO DE PACOTES NECESSARIOS
# (Mudar o(s) caminho(s) de sys.path.append, conforme necessario)

import sys
sys.path.append('/home/walliston/salome/SALOME-MECA-2013.1-LGPL/tools/
Code_aster_standalone_20131_public/STA10.8/bibpyt/Utilitai')
sys.path.append('/home/walliston/salome/SALOME-MECA-2013.1-LGPL/tools/
Code_aster_standalone_20131_public/public/eficas/Aster/Cata/cataSTA9')
#sys.path.append('/usr/lib/python2.7/dist-packages') # Nao eh necessario, a principio
from Utilitai.partition import *
from partition import *
from Table import *
import numpy
#import Numeric as N
import geonpy

# FIM DE PACOTES NECESSARIOS

#####
#
#####
# INICIO DE PARAMETROS DE OTIMIZACAO SESO E MULTIOBJETIVO

FRR = 1.85 # Fator de Rasio de Rejeicao inicial
FER = 0.95 # Fator Evolucionario de Rejeicao inicial
Rvol = 95 # % de elementos a serem retirados por iteracao (0-100)
VolMax = 20 # % maxima de volume final permitida a estrutura (0-100)
SIGult = 1500 # tensao maxima admissivel (criterio de parada)
FRRmax = 10 # valor maximo de FRR admissivel (criterio de parada)
CritRem = 0 # Criterio de remocao: 0 para TMM, 1 para compliance (1 apenas para analises nao lineares)

#Pesos para MSM (Weighted Sum Method)
w1 = 0.1595
w2 = 0.3994
w3 = 0.0812
w4 = 0.3599

# FIM DE PARAMETROS DE OTIMIZACAO SESO

```

Figura B.18 – Trecho inicial do código implementado em Python[®]

A Fig. B.18 mostra o trecho inicial do código, que corresponde às importações dos pacotes necessários e as declarações dos parâmetros de otimização, necessários para a análise. A seguir, a declaração das funções de otimização é feita, conforme ilustrado na Fig. B.19.

```
# INICIO DAS FUNCOES PYTHON DE OTIMIZACAO

### Funcao que retorna o valor atual do Volume ###
def GetVolume(VolList,VolDList,ig):
    def GetVolume(VolList,ig):
        VolAtual = 0
        # VD = 0
        n = len(VolList)
        # p = len(VolDList)
        Volume = open('/home/walliston/salome/WallScript/Volume%d.wsf' %(ig), 'w')
        for i in range(n):
            VolAtual = VolAtual + VolList[i]
        # for i in range(p):
        #     VD = VD + VolDList[i]

        # soma = VolAtual + VD

        Volume.write('%0.8f \n' %(VolAtual))
        # Volume.write('%0.8f \n' %(VD))
        # Volume.write('%0.8f \n' %(soma))

        return VolAtual
        # return VolAtual,VD

### ----- ###

### Funcao que retorna um array de tensoes NODAIS de Von Mises ###
def GetVmisesArrayNode(VmisesList,NodeLabel,ig):
    VmisesNode=[None]*nnos
    n = len(VmisesList)
    NodeTen = open('/home/walliston/salome/WallScript/VonMises (node)d.wsf' %(ig), 'w')
    NodeTen.write('TOTAL DE NOS: ')
    NodeTen.write('%d \n' %(n))
    NodeTen.write('\n')
    for i in range(n):
        Nodevalue = VmisesList[i]
        string = NodeLabel[i]
        mem_id = int(string[:])-1 #fazendo N1=0
        VmisesNode[mem_id] = Nodevalue
        NodeTen.write(NodeLabel[i])
        NodeTen.write(' ')
        NodeTen.write('%0.8f \n' %(VmisesList[i]))

    return VmisesNode

###-----###

### Funcao que retorna um array de tensoes POR ELEMENTO de Von Mises ###
def GetVmisesArrayElem(VmisesNode,Connex,nelem,ig):
    VmisesElem = [None]*nelem
    ElemTen = open('/home/walliston/salome/WallScript/VonMises (elem)d.wsf' %(ig), 'w')
    ElemTen.write('TOTAL DE ELEMENTOS: ')
    ElemTen.write('%d \n' %(nelem))
    ElemTen.write('\n')
    for i in range(nelem):
        if deleted[i]:
            continue # ignora elementos deletados
        numnod = len(Connex[i]) # numero de nos dos elementos
```

Figura B.19 – Primeira parte do código implementado em Python®

As funções declaradas retornam arrays (conjunto de valores) para serem lidos posteriormente. Alguns desses arrays são impressos em arquivos temporários para uma visualização do usuário, garantindo um bom acompanhamento do cálculo ao longo das iterações de

otimização. As macro-rotinas são declaradas em seguida, como pode ser visto na Fig. B.20.

```
# INICIO DAS MACRO-ROTINAS ASTER

# Definição do tamanho das arrays

Mate = [None]*50
Mode = [None]*50
Mesh = [None]*50
Resu = [None]*50
#Fix = [None]*50
Char = [None]*50
#Rampe = [None]*50
#TList = [None]*50
TabVM = [None]*50
TabC = [None]*50
TabD = [None]*50
Volu = [None]*50
Xi = [None]*50 #volume reduzido (para ponto otimo)
Yi = [None]*50 #tensao max/tensao med (para ponto otimo)
Zi = [None]*50 #tensao max10/tensao med (para ponto otimo)

WSMsol = [None]*50 #vetor solucao de weighted sum method
WSM = [None]*50 #matriz de weighted sum method
for i in range(50):
    WSM[i] = [None]*4

#Declaração das Macro-Rotinas Aster

DEBUT(PAR_LOT='NON',);

Steel=DEFI_MATERIAU(ELAS=_F(E=210e3,
                           NU=0.3,
                           RHO=7850.0,),,);

Trash=DEFI_MATERIAU(ELAS=_F(E=1000,
                           NU=0.3,
                           RHO=1000,),,);

Mesh[0]=LIRE_MAILLAGE(UNITE=20,
                     FORMAT='MED',
                     INFO=1,);
```

Figura B.20 – Segunda parte do código implementado em Python®

Aqui são mostradas as primeiras macro-rotinas da análise. Os arrays definidos em Python®, ao serem variáveis, podem ser usados pelo Code-Aster® para armazenar todos os valores em todos os passos de otimização da análise. Uma importante e fundamental característica do Code-Aster® aqui é que o arquivo de saída “*.rmed” guarda todas as informações das malhas de cada iteração em um só arquivo, facilitando sua manipulação e visualização ao fim da análise. Assim, o arquivo é atualizado a cada iteração, contendo todas as malhas até a presente iteração.

Além disso, outro ponto fundamental nessa parte do código é a possibilidade de importação da malha gerada pelo Code-Aster® em um objeto Python®, permitindo a manipulação da malha pelas funções definidas na primeira parte para, em seguida, retorná-la ao Code-Aster® com as devidas alterações. E, ao fim desse trecho mostrado, ainda há a extração das propriedades do modelo em uma variável Python® a fim de monitorar o volume inicial assim como em cada iteração, além de poder usá-lo como critério de parada de remoção de elementos.

Logo após a análise do Code-Aster®, as chamadas das funções de otimização acontecem progressivamente, como na Fig. B.21.

```

# INICIO DAS CHAMADAS DE ROTINAS DE OTIMIZACAO SESO
    Tab0 = Tab[i].EXTR_TABLE() # extrai uma tabela qualquer do Code Aster (nesse caso, VMIS de
    CREA_TABLE)

    pytab = Tab0.values() # formata os valores de uma tabela para ser exportada para python

    VmisesNode = GetVmisesArrayNode(pytab['VMIS'],pytab['NOEUD'],ig) # chamada da função que retorna
    um array de tensões de Von Mises por no

    VmisesElem = GetVmisesArrayElem(VmisesNode,Connex,nelem,ig) # chamada da função que retorna um
    array de tensões de Von Mises por elemento

    VmisesMax = GetVmisesArrayMax(VmisesElem,ig) # chamada da função que retorna um array de tensões
    de Von Mises por elemento em ordem decrescente

    TVM0 = VmisesMax[0][1]*Initvol

    Vol = Initvol

    Iter = Iter+1

    FER = GetFER(FER)

    FRR = GetFRR(FRR,Iter,FER)

    IP = GetPerformanceIndex(TVM0,VmisesMax,Vol,Iter,FER,FRR,ig)

    TVMmax = GetTVMmax(VmisesMax,nelem)

    retelem,removed,NewModel = GetNewTopology(ElemGroup,TVMmax,nelem,FRR,VmisesElem,ig)

    Mesh[i+1]=CREA_MAILLAGE(MAILLAGE=Mesh[i],
                           DETR_GROUP_MA= F(GROUP_MA='all',),);

    Mesh[i+1]=DEFI_GROUP(reuse =Mesh[i+1],
                        MAILLAGE=Mesh[i+1],
                        CREA_GROUP_MA= F(NOM='all',
                                          MAILLE=NewModel,
                                          TYPE_MAILLE='3D',),);

    Volu[i]=POST_ELEM(MASS_INER= F(MAILLE=NewModel,),
                     MODELE=Mode[i],
                     CHAM_MATER=Mate[i],);

#   VoluD[i]=POST_ELEM(MASS_INER= F(MAILLE=removed,),
#                     MODELE=Mode[i],
#                     CHAM_MATER=Mate[i],); #####

    DEFI_FICHER(ACTION='ASSOCIER',
                UNITE=55,
                FICHER='/home/walliston/salome/WallScript/MassData',
                TYPE='ASCII',
                ACCES='NEW',);

    IMPR_TABLE(TABLE=Volu[i],
              FORMAT='TABLEAU',
              UNITE=55,);

#   IMPR_TABLE(TABLE=VoluD[i],
#             FORMAT='TABLEAU',
#             UNITE=56,); #####

    Vol0 = Volu[i].EXTR_TABLE() # extrai uma tabela qualquer do Code Aster (nesse caso, VMIS de
    CREA_TABLE)

```

Figura B.21 – Terceira parte do código implementado em Python®

Finalmente, os resultados do Code-Aster® são definidos e exportados em tabelas para serem feitas as devidas verificações, de acordo com a terceira e última parte do código. Uma a uma as rotinas de otimização são chamadas para que os elementos sejam avaliados e devidamente removidos. E, ao fim das rotinas, o ciclo é repetido até que o critério de remoção não seja mais atendido e/ou o critério de parada seja alcançado, conforme o fluxograma apresentado na Fig. B.17, encerrando a análise e gerando a topologia ótima final.