

PONTIFÍCIA UNIVERSIDADE CATÓLICA
DO RIO DE JANEIRO



Rafael Araujo de Sousa

**Adaptatividade geométrica e numérica na geração
de malhas de elementos finitos em 2D e 3D**

Dissertação de Mestrado

Dissertação apresentada como requisito parcial para obtenção do título de Mestre pelo Programa de Pós-Graduação em Engenharia Civil da PUC-Rio. Área de Concentração: Estruturas.

Orientador: Luiz Fernando Campos Ramos Martha
Co-orientador: Antônio Carlos de Oliveira Miranda

Rio de Janeiro
Fevereiro de 2007.



Rafael Araujo de souza

**Adaptatividade geométrica e numérica na geração
de malhas de elementos finitos em 2D e 3D**

Dissertação apresentada como requisito parcial para obtenção do título de Mestre pelo Programa de Pós-Graduação em Engenharia Civil do Centro Técnico Científico da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

Prof. Luiz Fernando C. R. Martha

Presidente/Orientador
Departamento de Engenharia Civil - PUC-Rio

Prof. Carlos Alberto de Almeida

Departamento de Engenharia Mecânica - PUC-Rio

Prof. Marcelo de Andrade Dreux

Departamento de Engenharia Mecânica - PUC-Rio

Prof. Remo Magalhães de Souza

Departamento de Engenharia Civil - UFPA

Prof. William Wagner Matos Lira

Departamento de Engenharia Civil - UFAL

Prof. José Eugênio Leal

Coordenador Setorial
do Centro Técnico Científico - PUC-Rio

Rio de Janeiro, 14 de Fevereiro de 2007.

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

Rafael Araujo de Sousa

Graduou-se em Engenharia Civil, pela Universidade Federal do Pará, em fevereiro de 2005. Durante a graduação atuou na área de estruturas na análise de vibrações em cabos considerando a rigidez à flexão, a rigidez ao cisalhamento e a inércia à rotação.

Ficha Catalográfica

De Sousa, Rafael Araujo

Adaptatividade geométrica e numérica na geração de malhas de elementos finitos em 2D e 3D

Geração de malhas adaptativas de elementos finitos/ Rafael Araujo de Sousa; orientador: Luiz Fernando C. R. Martha ; co-orientador: Antônio Carlos de Oliveira Miranda. – 2007.

138 f.:il.; 29,7cm

Dissertação (Mestrado em Engenharia civil)– Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2007.

Inclui bibliografia.

1. Engenharia civil – Teses. 2. Elementos finitos. 3. Análise adaptativa. 4. Geração de malha. 5. Adaptatividade geométrica. I. Martha, Luiz Fernando Campos Ramos. II. Miranda, Antônio Carlos de Oliveira. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Civil. IV. Título.

CDD: 624

Aos meus pais, Stélio e Maria do Carmo;
Meu irmão, Allan Araujo de Sousa;
Minha avó, Paula Costa Ferreira Araujo.

Agradecimentos

Aos meus pais, Stélio Augusto Amorim de Sousa e Maria do Carmo Araujo de Sousa e a Deus, que me permitiram ingressar na universidade, um centro de conhecimento de valor imensurável, ao qual, infelizmente, poucos de nós em nosso país têm acesso.

A meu irmão Allan Araujo de Sousa; a minha avó, Paula Costa Ferreira Araujo e a todos meus tios e primos .

À Vanessa Ferreira, que esteve por muito tempo muito presente na minha vida e que até hoje tem influência sobre minhas decisões.

A todos os professores do departamento de engenharia civil da PUC-RIO, principalmente ao professor Luiz Fernando C. R. Martha pela orientação e pela oportunidade de trabalhar em um ambiente agradabilíssimo e em um tema tão interessante.

Ao Tecgraf pelo apoio financeiro e a todos meus amigos de lá: Joaquim Bento, William Lira, que sempre esteve disposto a me ajudar em todos instantes que o solicitei, Isabelle Telles *“paraense pai d’égua”*, Fábio Pereira *“companheiro do café da AFPUC”* e que também sempre se demonstrou pronto para me ajudar, ao Luiz Fernando Bitton, pelo incentivo em ingressar no Tecgraf e pela confiança depositada em mim, ao grande Antônio Miranda *“Caaaaramba”* pela grande ajuda e orientação no decorrer da dissertação e ao Marcos Arruda, que também me ajudou bastante e pela sua enorme paciência nas explicações e auxílios durante a implementação deste trabalho.

A todos meus colegas da pós graduação da PUC, especialmente a Amanda Jarek, Antônio Vicente *“Combatente”*, André Guedes *“Dedé Bambu”* e principalmente ao pessoal do Norte do Brasil (Pará e Amapá): Ygor Netto *“tio Chico”*, Renato Mendes *“Bolo”*, Plínio, Klessis, Jean *“Dado”* que me acompanha desde a graduação e ao grande Adenilson *“Mão toca”* parceiro do samba, dos bailes Funks, das disciplinas do mestrado e do esporte.

A todos os professores de engenharia civil da UFPa, em especial aos professores: José Perilo da Rosa Neto, José Raimundo Serra Pacha e principalmente ao Remo Magalhães de Souza, um grande professor, orientador e amigo, a quem tenho um grande sentimento de admiração, respeito e amizade. Ao Tiago Rodrigues da Silva, amigo do período da graduação.

A minha afilhada Arielle Bezerra Pinheiro, minha comadre Andréia Bezerra e ao meu grande amigo, compadre e xará Rafael Ferreira Pinheiro "*Cachacinha*" companheiro dos sambas e cervejas.

A todas outras amizades consolidadas durante minha vida: Fábio Luíz "*Chuck*" e toda a sua família, Bruno Flexa "*Tabajara*", Bruna Cristina "*preta Bru*", Francisco Rocha "*Bulldogru*", Igor Nunes "*Malaco*", Evandro Leomar "*Mano Brown*" e seu irmão Damião "*Jimmy*", etc.

A todos os poetas e compositores do samba, que sempre estiveram presentes durante toda esta estadia no Rio de Janeiro, tanto nos momentos de estudo quanto nos de lazer.

A CAPES, ao TECGRAF e a PUC-RIO por terem me dado a oportunidade de vir cursar a pós-graduação com as condições necessárias e suficientes para a sua conclusão.

Resumo

De sousa, Rafael Araujo.; Martha, Luiz Fernando C. R.

Adaptatividade geométrica e numérica na geração de malhas de elementos finitos em 2D e 3D. Rio de Janeiro, 2007. 138p. Dissertação de Mestrado - Departamento de Engenharia Civil, Pontifícia Universidade Católica do Rio de Janeiro.

Este trabalho apresenta uma metodologia para geração de malhas adaptativas de elementos finitos 2D e 3D usando modeladores geométricos com multi-regiões e superfícies paramétricas. A estratégia adaptativa adotada é fundamentada no refinamento independente das curvas, superfícies e sólidos. Inicialmente as curvas são refinadas, no seu espaço paramétrico, usando uma técnica de partição binária da curva (*binary-tree*). A discretização das curvas é usada como dado de entrada para o refinamento das superfícies. A discretização destas é realizada no seu espaço paramétrico e utiliza uma técnica de avanço de fronteira combinada com uma estrutura de dados do tipo *quadtree* para gerar uma malha não estruturada de superfície. Essas malhas de superfícies são usadas como dado de entrada para o refinamento dos domínios volumétricos. A discretização volumétrica combina uma estrutura de dados do tipo *octree* juntamente com a técnica de avanço de fronteira para gerar uma malha sólida não estruturada de elementos tetraédricos. As estruturas de dados auxiliares dos tipos *binary-tree*, *quadtree* e *octree* são utilizadas para armazenar os tamanhos característicos dos elementos gerados no refinamento das curvas, superfícies e regiões volumétricas. Estes tamanhos característicos são definidos pela estimativa de erro numérico associado à malha global do passo anterior do processo adaptativo. A estratégia adaptativa é implementada em dois modeladores: o *MTOOL* (2D) e o *MG* (3D), que são responsáveis pela criação de um modelo geométrico, podendo ter, multi-regiões, onde no caso 3D as curvas e superfícies são representadas por NURBS.

Palavras-chave

Elementos finitos; análise adaptativa; geração de malha; adaptatividade geométrica.

Abstract

De sousa, Rafael Araujo.; Martha, Luiz Fernando C. R.
Geometric and numerical adaptativity of 2D and 3D finite element meshes. Rio de Janeiro, 2007. 138p. Dissertação de Mestrado - Departamento de Engenharia Civil, Pontifícia Universidade Católica do Rio de Janeiro.

This work presents a methodology for adaptive generation of 2D and 3D finite-element meshes using geometric modeling with multi-regions and parametric surfaces. The adaptive strategy adopted in this methodology is based on independent refinements of curves, surfaces and solids. Initially, the model's curves are refined using a binary-partition algorithm in parametric space. The discretization of these curves is used as input for the refinement of adjacent surfaces. Surface discretization is also performed in parametric space and employs a *quadtree*-based refinement coupled to an advancing-front technique for the generation of an unstructured triangulation. These surface meshes are used as input for the refinement adjacent volumetric domains. Volume discretization combines an *octree* refinement with an advancing-front technique to generate an unstructural mesh of tetrahedral elements. In all stages of the adaptive strategy, the refinement of curves, surface meshes and solid meshes is based on estimated numerical errors associated to the mesh of the previous step in the adaptive process. In addition, curve and surface refinement takes into account metric distortions between parametric and Cartesian spaces and high curvatures of the model's geometric entities. The adaptive strategies are implemented in two different modelers: *MTOOL* (2D) and *MG* (3D), which are responsible for the creation of a geometric model with multi-regions, where for case 3D the curves and surfaces are represented by NURBS, and for the interactive and automatic finite-element mesh generation associated to surfaces and solid regions. Numerical examples of the simulation of engineering problems are presented in order to validate the methodology proposed in this work.

Keywords

Finite elements; adaptive analysis; mesh generation; geometric adaptivity.

Sumário

1. Introdução	18
1.1. Revisão bibliográfica	22
1.2. Objetivos	27
1.3. Organização do texto	28
2. Base Teórica	29
2.1. Estratégia adaptativa	29
2.2. Estimativa de erro	30
2.3. Métodos de suavização	35
2.3.1. Zienkiewicz e Zhu – Hinton e Campbell (Z2-HC)	36
2.3.2. Superconvergent patch recovery (SPR)	37
2.4. Geração de malhas	39
2.4.1. Geração de malhas não-estruturadas	41
2.4.1.1. Triangulação de Delaunay	42
2.4.1.2. Decomposição espacial recursiva	43
2.4.1.3. Avanço de fronteira	44
3. Processo adaptativo	46
3.1. Resumo do processo adaptativo	46
3.2. Construção da <i>octree</i> / <i>quadtree</i> global baseada no erro numérico	49
3.3. Refinamento da <i>octree</i> / <i>quadtree</i> global baseado na adaptatividade geométrica das curvas	50
3.4. Refinamento da <i>octree</i> global baseado na adaptatividade geométrica das superfícies	54
3.5. Refinamento da <i>octree</i> / <i>quadtree</i> global para forçar disparidade de tamanho mínima	57
3.6. Refinamento das curvas	58
3.7. Discretização das superfícies	61
3.8. Algoritmo de geração de malhas volumétricas	62
3.9. Localização do tamanho característico H para os novos elementos	64
3.10. Considerações finais	66

4. Implementações do processo adaptativo	67
4.1. <i>MTOOL</i> (2D)	69
4.1.1. Adaptatividade no <i>MTOOL</i>	72
4.2. <i>MG</i> (3D)	77
4.2.1. Adaptatividade no <i>MG</i>	78
5. Exemplos	84
5.1. Exemplos em 2D (<i>MTOOL</i>)	85
5.1.1. Peça mecânica com um furo circular	86
5.1.2. Placa quadrada com furo quadrado	92
5.1.3. Viga curta em balanço	96
5.1.4. Placa com furo quadrado com dois materiais	102
5.1.5. Placa complexa com três furos circulares e um quadrado	107
5.1.6. Barragem	111
5.2. Exemplos em 3D (<i>MG</i>)	114
5.2.1. Viga curta em balanço	115
5.2.2. Placa com furo circular	118
5.2.3. Sólido com uma cavidade cilíndrica	121
5.2.4. Eixo cilíndrico	124
5.2.5. Pórtico misto	126
6. Conclusão	129
6.1. Vantagens do método	131
6.2. Desvantagens do método	132
6.3. Sugestões para trabalhos futuros	132
7. Referências bibliográficas	133

Lista de figuras

Figura 1 - Modelo geométrico bidimensional de uma peça mecânica	20
Figura 2 - Malha de elementos finitos criada sobre o modelo geométrico	21
Figura 3 - (a) malha original, (b) malha após o uso da estratégia do tipo h , (c) malha após o uso da estratégia do tipo p , (d) malha após o uso da estratégia do tipo r	30
Figura 4 - (a) mostra a contribuição de cada elemento, quadrilateral, para o valor de tensão, não suavizada, no nó adjacente aos mesmos, (b) mostra o valor da tensão suavizada no nó adjacente aos elementos quadrilaterais (Hinton&Campbell-[26])	37
Figura 5 - Notação da recuperação por patch: \odot nó que forma o patch; Δ pontos de amostragem; \bullet valores nodais determinados pelo procedimento de recuperação; \circ pontos nodais; r_i distância entre o ponto de amostragem i e o nó que forma o patch; Ω denota o domínio do problema e Ω_p indica o domínio do patch (Cavalcante-Neto J.B.-1998 [13])	38
Figura 6 - Tipos de malha (Batista V.H.F.-2005 [6])	40
Figura 7 - Malha de elementos finitos gerada por triangulação de Delaunay (Batista [6])	42
Figura 8 - Exemplo de geração de uma malha 2D de elementos finitos usando uma quadtree (Batista [6])	43
Figura 9 - Estrutura quadtree em forma de diagrama	44
Figura 10 - Exemplo de geração de uma malha 2D de elementos finitos usando avanço de fronteira (Batista V.H.F.-2005 [6])	45
Figura 11 - Fluxograma do novo processo adaptativo (2D e 3D)	47
Figura 12 - Curva a ser subdividida segundo os critérios geométricos	51
Figura 13 - (a) Ponto P_0 é inserido (b) Inseri-se uma nova subdivisão, fazendo-se $P_1 = P_m$ e P_m sendo a metade do novo trecho	52

Figura 14 - (a) Tubo com pressão interna, (b) Malha grosseira do modelo do tubo, (c) resultado do 1º passo do processo adaptativo usando apenas critérios baseados na estimativa de erro e (d) resultado do 1º passo do processo adaptativo usando os critérios de estimativa de erro e os critérios geométricos	53
Figura 15 - Processo de geração de malhas superficiais paramétricas em 3D (Lira [37])	54
Figura 16 - Superfície com as suas curvas de bordo considerando os parâmetros da adaptatividade geométrica das curvas	56
Figura 17 - Exemplo de uma malha em uma superfície 3D: (a) sem considerar informações sobre curvaturas; (b) quadtree auxiliar construída para realizar a adaptatividade geométrica da superfície da figura (c) e (c) malha da superfície após se considerar as suas curvaturas (Lira [37])	57
Figura 18 - (a) quadtree inicial sem refinamento que força a diferença mínima entre as células da octree / quadtree, (b) quadtree final com refinamento adicional que força a diferença mínima entre as células da octree / quadtree	58
Figura 19 - Refinamento de uma curva reta pela técnica de Decomposição Espacial Recursiva binária (binary-tree)	59
Figura 20 - (a) malha inicial com todas as curvas do tipo arco de círculo com dez subdivisões, (b) malha final com suas curvas com o número de subdivisão menor que dez	60
Figura 21 - Problema de refinamento em cantos quando o valor de h de cada elemento é associado ao seu centro geométrico	64
Figura 22 - Compara as malhas, referentes ao mesmo passo do processo adaptativo, com o valor de h de cada elemento associado ao seu centro geométrico (a) e com o valor de h associado aos vértices de cada elemento (b)	65
Figura 23 - Fluxograma mostrando o funcionamento do <i>MTOOL</i>	70
Figura 24 - Pseudo código da função <i>_MtoolAdaptiveQt</i>	73
Figura 25 - Pseudo-código da função que preenche o valor de h para elementos do tipo T3	75
Figura 26 - Pseudo-código da função que constrói a quadtree considerando ou não a adaptatividade baseada no erro numérico e na adaptatividade geométrica	76

Figura 27 - Pseudo código da função <i>_MtoolAdaptiveMesh</i>	76
Figura 28 - Pseudo código no <i>MG</i> que gerencia a gravação e leitura dos arquivos neutros	80
Figura 29 - Pseudo código no <i>MG</i> que mostra o método <i>mgRegenMeshes</i>	81
Figura 30 - Pseudo código no <i>MG</i> que mostra o método <i>mgAdapt_CreateMesh</i> , que é o método que gerencia todo o processo adaptativo	83
Figura 31 - Peça mecânica com um furo circular	86
Figura 32 - Comparação entre os métodos de suavização <i>Z2-HC</i> e <i>SPR</i> , utilizando-se elementos lineares (T3) usando apenas a adaptatividade numérica: malha inicial (a), malhas seguintes (b,c,d)	88
Figura 33 - Exemplo 1 com T3 usando a adaptatividade geométrica e numérica: malha inicial (a), malhas seguintes (b,c,d)	89
Figura 34 - Exemplo 1 com T3 e usando apenas a adaptatividade geométrica: malha inicial (a), malha seguinte (b)	90
Figura 35 - Exemplo 1 com T6 sem a adaptatividade geométrica mas com a adaptatividade numérica: malha inicial (a), malhas seguintes (b,c)	90
Figura 36 - Exemplo 1 com T6 usando a adaptatividade geométrica e numérica: malha inicial (a), malhas seguintes (b,c)	91
Figura 37 - Placa quadrada com furo quadrado	92
Figura 38 - Exemplo 2 com T3: malha inicial (a), malha seguinte (b,c,d)	93
Figura 39 - Exemplo 2 com T6: malha inicial (a), malhas seguintes (b,c)	94
Figura 40 - Viga curta em balanço	96
Figura 41 - Exemplo 3 com T3: malha inicial (a), malhas seguintes (b,c,d)	97
Figura 42 - Mostra a distribuição da razão de erro x_i entre os elementos da malha durante os passos do processo adaptativo	98
Figura 43 - Exemplo 3 com T6: malha inicial (a), malhas seguintes (b,c,d)	99
Figura 44 - Placa quadrada com furo quadrado com dois materiais	102
Figura 45 - Exemplo 4 com T3: malha inicial (a), malhas seguintes (b,c,d)	103
Figura 46 - Exemplo 4 com T6: malha inicial (a), malhas seguintes (b,c,d)	104
Figura 47 - Placa complexa com três furos circulares e um quadrado	107
Figura 48 - Exemplo 5 com T3: malha inicial (a), malhas seguintes (b,c,d)	108
Figura 49 - Exemplo 5 com T6: malha inicial (a), malhas seguintes (b,c)	109
Figura 50 - Barragem bidimensional	111

Figura 51 - Exemplo 6 com T3: malha inicial (a), malhas seguintes (b,c,d)	112
Figura 52 - Exemplo 6 com T6: malha inicial (a), malhas seguintes (b,c,d)	113
Figura 53 - Viga curta em balanço	115
Figura 54 - Exemplo 1 com TETRA10: malha inicial (a), malhas seguintes (b,c)	116
Figura 55 - Placa com furo circular	118
Figura 56 - Exemplo 2 com TETRA10: malha inicial (a), malhas seguintes (b,c)	119
Figura 57 - Sólido com cavidade cilíndrica	121
Figura 58 - Exemplo 3 com TETRA10: malha inicial (a), malhas seguintes (b,c)	122
Figura 59 - Eixo cilíndrico	124
Figura 60 - Exemplo 4 com TETRA10: malha inicial (a), malha seguinte (b)	125
Figura 61 - Pórtico misto 3D	126
Figura 62 - Exemplo 5 com TETRA10: malha inicial (a), malhas seguintes (b,c,d)	127

Lista de tabelas

Tabela 1 - Comparação dos resultados para a placa quadrada com furo quadrado	95
Tabela 2 - Comparação dos resultados para a viga curta em balanço	100
Tabela 3 - Comparação dos resultados para a placa quadrada com furo quadrado para dois materiais diferentes	105
Tabela 4 - Comparação dos resultados para a placa complexa com três furos circulares diferentes	110
Tabela 5 - Comparação dos resultados do exemplo bidimensional da Barragem, mostrando os valores obtidos por Zienkiewicz e o método proposto	114
Tabela 6 - Comparação dos resultados do exemplo da viga curta em balanço 3D	117
Tabela 7 - Comparação dos resultados do exemplo da placa 3D com furo circular	120
Tabela 8 - Comparação dos resultados do exemplo do sólido com uma cavidade cilíndrica	123
Tabela 9 - Comparação dos resultados do exemplo do eixo cilíndrico	125
Tabela 10 - Mostra os resultados obtidos para o exemplo do pórtico misto	128

Lista de símbolos

h	tamanho característico de cada elemento.
h_i	tamanho característico de cada elemento no passo anterior do processo.
\mathbf{x}_i	razão de erro de cada elemento.
D	matriz que define as tensões em função das deformações.
u	campo de deslocamentos.
\mathbf{e}	campo de deformações.
S	operador diferencial.
\mathbf{s}	campo de tensões.
q	vetor de carregamento.
t_p	vetor de forças prescritas.
N	funções de formas.
\hat{u}	valores nodais de deslocamento.
\tilde{u}	solução numérica dos deslocamentos.
K	matriz de rigidez.
E	módulo de elasticidade.
ν	coeficiente de Poisson.
Λ	funcional de energia.
f	vetor de forças nodais.
e	erro.
e_e	erro em deformações.
e_s	erro em tensões.
$\ e\ $	norma de energia do erro.
h	erro relativo.
h^*	máximo erro relativo.
\bar{h}	erro relativo corrente.
SPR	Superconvergent Recovery Patch.
$Z2-HC$	Zienkiewicz e Zhu – Hinton e Campbell.
MEF	Método dos Elementos Finitos.

MG	Mesh Generation.
FEMOOP	Finite Element Method – Object Oriented Programming.
MTOOL	Two-dimensional Mesh Tool.
ESAM	Extensible System Attributes Management.
CGC	Complete Geometric Complex.

1 INTRODUÇÃO

Dentre os diversos ramos de estudo da Engenharia Civil, encontra-se a análise estrutural, que consiste, descrevendo-se de forma resumida, em analisar as tensões e deformações presentes em um corpo qualquer, quando o mesmo está submetido a ações externas, tais como carregamentos ou deslocamentos impostos. Para tanto, é necessário conhecer as propriedades mecânicas dos materiais da estrutura, tais como seu modelo constitutivo, que define, por exemplo, se o seu comportamento é linear-elástico ou não-linear e plástico, ou se é homogêneo ou heterogêneo. Outro item a ser considerado são as características de atuação das ações externas, que podem ser estáticas ou dinâmicas. De uma forma resumida, o problema matemático da análise estrutural consiste em resolver equações diferenciais de equilíbrio (estático ou dinâmico) e compatibilidade e relações constitutivas em um determinado domínio, com condições de contorno em termos de deslocamentos e forças que podem variar ao longo do tempo.

Na maioria dos casos práticos, não se consegue resolver esse problema matemático analiticamente, quer seja pela complexidade geométrica da estrutura, pela complexidade das relações constitutivas, pela complexidade das ações externas, ou pela combinação de todos esses fatores.

Na prática, esse problema é resolvido utilizando-se métodos numéricos, que fornecem uma solução aproximada do problema. Os métodos numéricos mais conhecidos para resolver o problema da análise estrutural são o método das diferenças finitas, o método dos elementos finitos e o método dos elementos de contorno. O mais difundido é o método dos elementos finitos (MEF). Uma razão para isto é a confiabilidade da solução numérica obtida e a relativa simplicidade da implementação computacional do método, possibilitando a elaboração de programas robustos.

Prazeres em 2005 [51] define formalmente o MEF da seguinte forma: “é um método numérico aproximado para solução de problemas em meios contínuos que são descritos através de equações diferenciais, sob determinadas condições de contorno e condições iniciais. Como o método utiliza pontos do domínio, é classificado como um método de domínio. Porém, falar de elementos

finitos é falar de uma forma bastante genérica, já que existem vários métodos de elementos finitos^{*}, dentre os quais, o método da rigidez direta ou método dos deslocamentos é o mais difundido”.

A idéia principal do MEF é subdividir o domínio em pequenas regiões (elementos) com uma topologia pré-definida (triângulos, quadriláteros, tetraedros, etc), onde o comportamento do campo de interesse (tensões ou deslocamentos) pode ser aproximado por funções simples, tais como polinômios ou funções harmônicas. Os valores do campo em questão nos vértices (nós) destes elementos são determinados pela solução numérica, donde através da interpolação destes valores obtêm-se os valores do campo dentro do referido elemento. Este processo de subdividir o domínio é conhecido como discretização do modelo, de tal forma, que o modelo discretizado resultante é denominado de malha de elementos finitos.

Cavalcante-Neto em 1998 [13] destaca outro aspecto importante do MEF, que é a definição do grau de refinamento da malha, “pois pelo fato das funções de aproximação dos campos dentro dos elementos serem relativamente simples é necessário aumentar o grau de refinamento em regiões de gradientes elevados do campo de solução. O problema disto é que o campo da solução é geralmente desconhecido. Práticas normais para resolver este problema, envolvem em aumentar o número de pontos de discretização no domínio computacional e resolver o sistema de equações resultantes para examinar a mudança relativa na solução numérica. Em geral, este procedimento consome tempo, depende da experiência do analista e pode guiar a interpretações erradas se a solução não tiver entrado em uma faixa assintótica de convergência”.

Este trabalho trata da geração de malha de elementos finitos para modelos estruturais bi e tridimensionais de forma automática e adaptativa, isto é, com refinamento da malha de acordo com o gradiente do campo da solução.

Além do problema da geração da malha de elementos finitos, existe o problema da criação do modelo geométrico da estrutura, de sua visualização e da visualização dos resultados numéricos da análise estrutural. Estes problemas são tratados pela Computação Gráfica, que também trata de técnicas de modelagem de sólidos [27]. Diversas fases são importantes na modelagem de sólidos, indo desde a interface com o usuário até as estruturas de dados que serão utilizadas. Portanto, é importante usar programas especiais, chamados de modeladores, que reproduzem, em formato digital, as formas geométricas dos

^{*} para mais detalhes ver [51,68]

objetos da simulação [42]. Neste trabalho dois modeladores geométricos foram usados: o *MTOOL* (2D) [47] e o *MG* (3D) [48], ambos desenvolvidos no Tecgraf/PUC-Rio (Grupo de Tecnologia em Computação Gráfica). Mais detalhes sobre esses modeladores serão vistos no capítulo 4.

Processos adaptativos de análise utilizados em conjunção com ferramentas de modelagem de sólidos têm se tornado muito importantes para aumentar a confiabilidade do procedimento de análise, pois o tornam menos dependente da experiência do analista. Um processo adaptativo é um procedimento para avaliar o erro de discretização do modelo para uma dada solução e para melhorar esta discretização até se atingir uma tolerância de erro fornecida para a solução numérica. Portanto, é fundamental, dentro do contexto adaptativo, definir um método de estimativa do erro de discretização, além de técnicas de geração adaptativas de malhas em 2D e 3D.

De forma a caracterizar bem o problema que esta dissertação aborda, a Figura 1 mostra um modelo geométrico bidimensional de uma estrutura e a Figura 2 mostra uma malha de elementos finitos associada a este modelo.

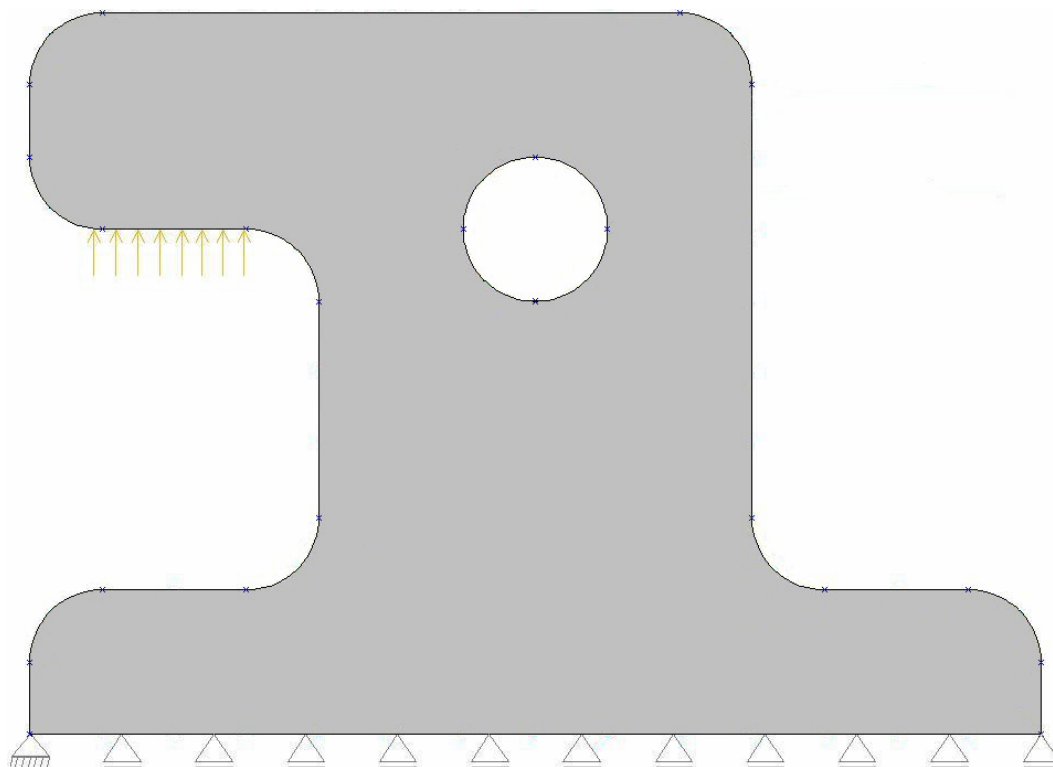


Figura 1 Modelo geométrico bidimensional de uma peça mecânica.

A Figura 1 mostra o modelo geométrico de uma peça mecânica, onde estão destacados segmentos de curvas de bordo e os vértices nas extremidades dos segmentos. As condições de contorno em termos de cargas distribuídas de bordo aplicadas e de restrições de apoio também estão indicadas.

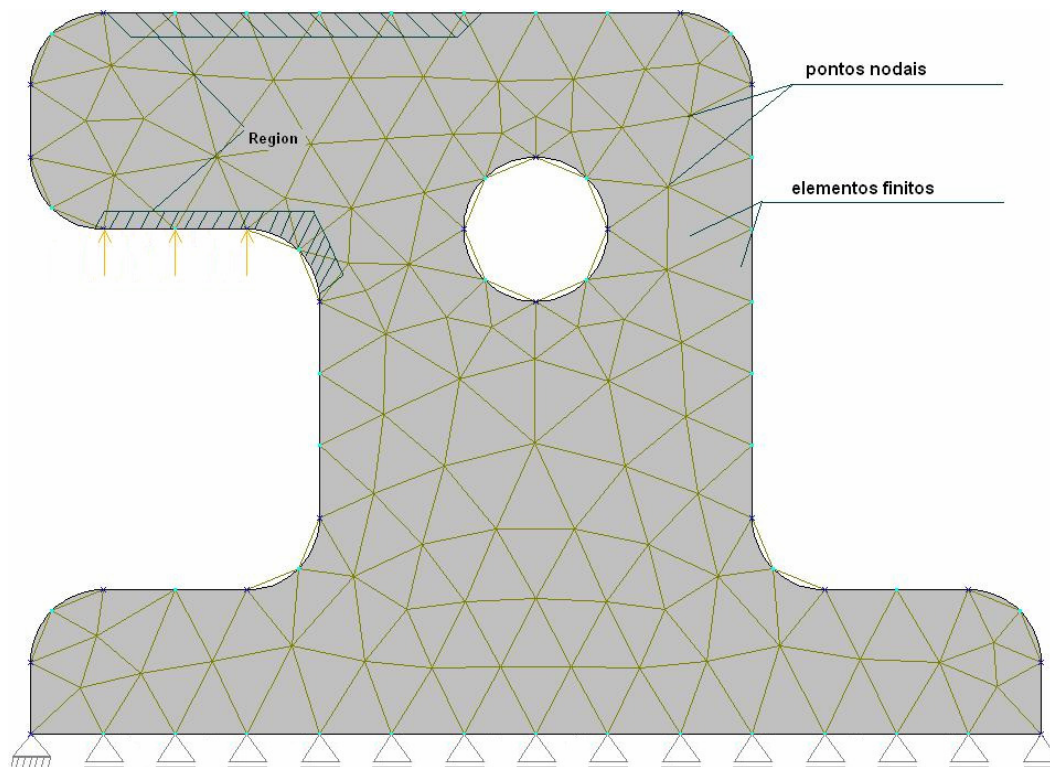


Figura 2 Malha de elementos finitos criada sobre o modelo geométrico.

A Figura 2 mostra a malha de elementos finitos correspondente ao modelo da peça mecânica da Figura 1. Estão destacadas as subdivisões de cada segmento de curva de bordo e a discretização do domínio em elementos finitos. Na malha de elementos finitos as cargas aplicadas e as restrições de apoio são transferidas de forma consistente para os pontos nodais (nós) da malha. As regiões hachuradas (*Region*), são regiões onde ocorrem concentrações de tensões. São estas regiões que devem ser mais refinadas, podendo para isto, diminuir o tamanho dos elementos (estratégia h), aumentar o grau da função de interpolação (estratégia p), realocar os nós existentes para que os mesmos fiquem mais próximos destas regiões (estratégia r), ou usando uma combinação destas técnicas, como por exemplo (estratégia $h-p$), (estratégia $h-r$). Neste trabalho usou-se apenas a estratégia h . Para fazer isso, seria necessário que o analista tivesse alguma idéia de onde ocorrem às concentrações de tensões do modelo para refiná-lo onde for necessário. É nesse contexto que se mostra a importância do processo adaptativo, pois o analista não precisa ter conhecimento das regiões onde ocorrem às concentrações de tensões. No processo adaptativo, o refinamento (ou “desrefinamento”) das regiões é feito de forma automática em conformidade com a estimativa de erro adotada.

1.1. Revisão Bibliográfica

Em 1910 Richardson [52], utilizando diferenças finitas, publicou o primeiro artigo mostrando um procedimento geral para a estimativa de erro.

Na década de 50 com o uso do método dos elementos finitos surgiu um dos primeiros sistemas com aplicação em engenharia, o ICES (Integrate Civil Engineering) desenvolvido no MIT (Massachusetts Institute of Technology). Na década seguinte o surgimento do programa SAP (Structural Analysis Program) influenciou decisivamente no desenvolvimento do método.

No início da década de 60, as malhas usadas nas simulações de problemas de engenharia usando o MEF eram geradas manualmente e no decorrer desta década através do uso de computadores começou-se a simplificar a tarefa de discretizar domínios mais complexos, resultando no surgimento de um campo de pesquisa chamado de geração de malhas.

Apesar dos progressos na área de geração de malhas estarem diretamente relacionados com avanços de outras áreas, como por exemplo: geometria computacional, estruturas de dados, análise numérica, etc., as evoluções feitas nestas outras áreas, que foram significativas, não foi de mesma proporção aos métodos de geração de malhas, o que se tornou um fator limitante nas simulações de problemas no decorrer da década de 80. Tal obstáculo serviu de força motriz para que pesquisadores gastassem uma gama de esforços, embebidos em desenvolver algoritmos capazes de tratar domínios arbitrários. Thompson & Weatherill em 1992 [58] expõem detalhes dos métodos de geração de malhas existentes até o referido ano.

Nos anos 70 Babuska and Rheinboldt [4,5] consideraram resíduos locais para estimar o erro da análise de tal forma que um refinamento poderia ser feito apenas em elementos onde tivessem ocorrido erros grandes.

Em 1971 Brauchli & Oden [10] usaram um procedimento baseado em média e na projeção L_2 para estimar o erro da análise, procedimento que tinha fácil implementação em qualquer código de elementos finitos. Este método foi usado para recuperar os gradientes e bons estimadores de erro foram obtidos. Já em 1987, Zienkiewicz e Zhu [62] introduziram um estimador de erro baseado na obtenção de valores melhorados dos gradientes (tensões) usando alguns processos de recuperação de tensão disponível. Em 1992, Zienkiewicz e Zhu [63-66] fundamentados no trabalho de Brauchli e Oden desenvolveram o *SPR* (*Superconvergent Patch Recovery*), que é um procedimento de suavização de gradientes sobre grupos (*patches*) locais de elementos, onde um procedimento

de mínimos quadrados é aplicado a um campo de gradiente polinomial de ordem mais elevada em relação ao gradiente considerado em pontos de amostragem superconvergentes, obtidos através do cálculo de elementos finitos. Posteriormente, Boroomand e Zienkiewicz [8] apresentaram um novo método superconvergente, que satisfaz as condições de equilíbrio em uma forma fraca mas não requer nenhum conhecimento de pontos superconvergentes (*REP* [13]).

Com base nestas duas formulações, Zienkiewicz, em 2004, [67] divide os estimadores de erro em dois tipos diferentes: os que são fundamentados em resíduos e os que utilizam processo de recuperação dos patches. Objetivando verificar a melhor das duas formulações, Babuska e um grupo de pesquisadores da universidade do Texas (Austin) fizeram exaustivos testes e produziram uma metodologia geral. Zienkiewicz [67] nomeia este método como *Babuska patch test*. Babuska propôs um valor para cada estimador de erro com um índice de robustez e depois muitos outros testes concluíram que o índice de robustez foi definitivamente ótimo para os métodos de recuperação mostrando a superioridade destes em relação à formulação baseada em resíduos. Neste mesmo artigo, Zienkiewicz comenta que apesar dos estimadores de erros existentes fornecerem excelentes resultados eles são uma fonte de pesquisa ilimitada.

A energia de deformação foi um dos primeiros acréscimos ao processo de recuperação, o qual foi proposto por Fraeijns de Veubeke [20-22]. Neste trabalho mostra-se que o procedimento padrão (deslocamento) resultaria em uma energia de deformação menor que a verdadeira energia contida na estrutura, ou seja, o cálculo da energia de deformação pelos processos de recuperação sempre subestimam o valor real da energia de deformação. Conseqüentemente muitos testes baseados na energia complementar foram usados como tentativa de melhorar a obtenção da energia de deformação, porém este processo resultou em muitas dificuldades. Zienkiewicz chegou até a sugerir alguns aperfeiçoamentos, mas a metodologia não forneceu uma boa forma prática para fornecer a energia de deformação de uma análise estrutural.

Cavalcante-Neto [12] propôs um sistema integrado para análise de elementos finitos em 2D. Sistema que foi dividido em um pré-processador, para a geração das malhas, um módulo de análise e um pós-processador, para a visualização dos resultados. Neste trabalho o estimador de erro utilizado foi o de recuperação e a suavização das tensões foi feita através do projetor de Hinton e Campbell [26] e utilizando como técnicas para a geração de malha a decomposição espacial recursiva *quadtree* numa parte do domínio, *binary-tree*

no contorno do modelo e entre o domínio e o contorno usa-se a contração de contorno segundo o critério de Delaunay.

Lo em 1991 [40] apresentou um algoritmo para a discretização em triângulos de domínios planares arbitrários, gerando malhas de elementos finitos baseadas na técnica de triangulação de Delaunay e com o gerador de malhas desenvolvido pelo próprio Lo. A técnica utilizada na geração de malha e o gerador de malhas usados neste trabalho podem ser vistos com maiores detalhes em outros dois trabalhos do mesmo autor [38,39].

Beall & Shephard em 1996 [7] propuseram uma estrutura de dados baseada em uma hierarquia topológica, fornecendo desta forma uma estrutura de dados robusta para ser utilizada no processo adaptativo.

Sydenstricker *et al* em 2004 [57] apresentaram uma estimativa de erro *a-posteriori* de tensão para problemas com materiais heterogêneos com variações súbitas das propriedades elásticas.

Souza [56] fundamentado no trabalho de Meyer *et al* [46] propôs uma técnica de adaptatividade de geração de malha fundamentada apenas em critérios geométricos. Outro trabalho voltado para a qualidade da malha sem considerar a análise numérica de elementos finitos, mas que usa como estimador de erro a projeção L_2 é o de Grosso *et al* [25]. Outros trabalhos que fazem refinamento de malha considerando apenas a adaptatividade geométrica são os três artigos de Schreiner *et al* [53-55], propondo uma técnica de remalhamento de superfície e uma nova técnica de avanço de fronteira. O trabalho de Peixoto *et al* [49], onde é proposto um método de extração de malhas adaptativas e o trabalho de Kim *et al* [30] mostraram uma técnica de refinamento adaptativo em elementos finitos com enfoque para fluidos. Dentro do contexto de geração de malha, baseado apenas na adaptatividade geométrica, é importante ressaltar que o método clássico para extração de malhas a partir de volumes é o *Marching Cubes*^{*}.

Em 2001 Mackerle [41] divulgou uma exaustiva coleta de artigos publicados sobre estimativa de erro e adaptatividade em elementos finitos, desde 1990 até 2000. Neste trabalho Mackerle apenas comenta os tipos de estimadores de erro e os processos adaptativos já existentes, bem como as suas várias aplicações, fornecendo posteriormente uma vasta lista de artigos relacionados a cada tópico abordado.

^{*} <http://www.exaflop.org/docs/marchcubes>

O processo adaptativo em 2D já está bem estabelecido no meio científico, o que não ocorre para o caso 3D, fato também observado por A Merrouche *et al* [1], onde a parte relacionada à modelagem geométrica e geração de malha é muito mais complicada. Portanto, dos trabalhos pesquisados percebeu-se uma preocupação em desenvolver técnicas de geração de malhas mais eficientes e técnicas de otimização aplicadas ao processo de estimativa de erro. Abaixo, segue um pequeno resumo dos trabalhos que tratam exclusivamente sobre geração de malhas e adaptatividade 3D.

Frey *et al* em 1994 [23] apresentaram um algoritmo de geração de malha 3D adaptado para geometrias complexas.

Lee & Lo em 1997 [31] apresentaram um procedimento automático para o refinamento adaptativo em 3D. Neste trabalho os autores objetivaram eliminar alguns problemas observados no processo de suavização Superconvergent patch recovery (*SPR*), possibilitando obter campos de tensões suavizados mais “confiáveis” para problemas de elasticidade 3D.

Golias & Dutton em 1997 [24] apresentaram um método para a geração de tetraedros, baseado na triangulação de Delaunay. O método consiste em um procedimento de divisão de tetraedros e num método de otimização de malhas de tetraedros, fundamentado em uma organização topológica, usando a triangulação de Delaunay para gerar os tetraedros.

Lee & Lo em 1999 [32] usaram técnicas de otimização para obter melhorias no esforço computacional do processo de refinamento adaptativo de malhas de elementos finitos.

Cavalcante-Neto [13] estendeu seu trabalho anterior de 2D para 3D, migrando da linguagem de programação C para a C++, a qual é fundamentada na técnica de Programação Orientada a Objetos. Neste trabalho os métodos de suavização usados foram o *SPR* [63-66] e o *REP* [13].

Lee *et al* em 2000 [33] propuseram um algoritmo de refinamento de malha de elementos finitos 3D onde é possível converter malhas com elementos lineares em malhas com elementos quadráticos ou cúbicos. Foram testados os aumentos na ordem dos seguintes tipos de elementos finitos: tetraedro, pirâmide, hexaedro e prisma.

Lee em 2002 [34] propôs um novo algoritmo de geração de malha para gerar malhas em superfícies 3D, que é fundamentado em um novo modelo geométrico.

Aymone em 2003 [3] apresentou aspectos referentes às formulações Euleriana (descrição espacial) e Lagrangiana (descrição material) em algoritmos de geração de malhas em 3D.

Choi *et al* em 2004 [18] apresentaram um algoritmo para geração de malhas tetraédricas com refinamento local, fundamentado na técnica de avanço de fronteira, juntamente com uma técnica de otimização, fornecendo desta forma melhorias locais nas malhas (onde houver concentrações de tensões).

Lee & Xu em 2005 [35] apresentaram um novo algoritmo de geração de malha 3D para estruturas de paredes finas, este algoritmo é dividido em duas partes: na primeira fase gera-se uma malha de superfície no meio da estrutura de parede fina, posteriormente se usa esta malha para gerar a malha sólida 3D através de extrusão.

Hughes *et al* em 2005 [28] introduziram um novo tipo de análise, chamado por eles de análise *Isogeometric*, que é baseada em NURBS. A principal característica deste novo tipo de análise é a representação exata dos elementos por NURBS e o uso do conceito isoparamétrico para definir variáveis, como por exemplo: deslocamento, temperatura, etc.

Zuo *et al* em 2005 [68] apresentaram um método para eliminar dificuldades associadas ao remalhamento de malhas 3D em domínios de qualquer forma geométrica com ou sem descontinuidade geométrica. O novo algoritmo gera malhas de tetraedros em modelos com ou sem trincas.

Boussetta *et al* em 2006 [9] apresentaram um processo adaptativo para malhas de elementos finitos. Este processo não refaz a malha, mas sim combina um melhoramento local das arestas e nós vizinhos dos elementos que possuem um alto erro. Este processo também é válido para problemas com grande deformação. Os autores usaram o *SPR* [63-66] como método de suavização. Eles estabeleceram um limite máximo de elementos, alegando uma maior robustez do método proposto.

Acharjee & Zabarar em 2006 [2] apresentaram uma extensão do método de sensibilidade contínua (CSM) para 3D, esta extensão desenvolve equações de sensibilidade a partir de diferenciações das equações que governam o problema.

1.2. Objetivos

O enfoque deste trabalho é o de propor um novo processo adaptativo, fundamentado no refinamento independente das curvas, superfícies e sólidos. Além disto, o trabalho visa implementar as técnicas de geração de malha em 2D e 3D desenvolvidas inicialmente pelos trabalhos de Cavalcante-Neto [12,13], utilizando estimadores de erro baseados no processo de recuperação dos *patches* com uma nova metodologia na geração de malhas adaptativas de elementos finitos 2D e 3D com modeladores geométricos com multi-regiões e superfícies paramétricas, usando técnicas de decomposição espacial recursiva: *binary-tree* (1D), *quadtree* (2D) e *octree* (3D) juntamente com avanço de fronteira. Dentro do contexto de adaptatividade esse trabalho permite que o processo adaptativo ocorra respeitando a geometria do modelo de tal forma que que curvas e superfícies do modelo numérico sejam mais próximas das curvas e superfícies do modelo real.

Como objetivos específicos, citam-se:

- Geração adaptativa de malha em 2D, usando uma *quadtree* global.
- Geração adaptativa de malhas em 2D respeitando a geometria.
- Geração adaptativa de malha em 3D, usando uma *octree* global.
- Geração adaptativa de malhas em 3D respeitando a geometria.
- Conseguir refinar localmente regiões da malha que apresentem erros elevados ou que apresentem elementos de má qualidade de forma.
- Conseguir “desrefinar” localmente regiões da malha que não apresentem erros elevados ou também que não apresentem elementos de má qualidade de forma.

1.3. Organização do Texto

O capítulo 2 destina-se a apresentar a teoria utilizada em todo o processo adaptativo, desta forma ele está dividido em quatro seções: a primeira abordará as estratégias adaptativas, a segunda abordará a técnica de estimativa de erro utilizada, a terceira tratará das técnicas de suavização e a última mostrará as principais técnicas de geração de malhas.

No capítulo 3 se detalha o processo adaptativo utilizado neste trabalho, mostrando a técnica de geração de malha, a estratégia adaptativa, os critérios da adaptatividade geométrica, a construção da *octree* / *quadtree* global.

O capítulo 4 destina-se a apresentar as implementações feitas para a realização do trabalho, ou seja, mostra a parte interativa com o usuário do programa, assim como a parte da implementação referente à adaptatividade, usando o *MTOOL* [47] para 2D e o *MG* [48] para o caso 3D. Os detalhes referentes a todo processo de modelagem também são apresentados neste capítulo.

O capítulo 5 apresenta alguns exemplos padronizados na literatura técnica, objetivando desta forma validar a proposta apresentada pelo respectivo trabalho.

O capítulo 6 apresenta, por fim, as conclusões e sugestões para trabalhos futuros.

2 BASE TEÓRICA

Este capítulo tem por finalidade apresentar a formulação teórica do processo adaptativo adotado neste trabalho. Existem quatro aspectos fundamentais de um processo adaptativo para o problema da análise estrutural: a estratégia adaptativa, a técnica de estimativa de erro, os métodos de suavização utilizados e a técnica utilizada para a geração das malhas de elementos finitos.

2.1. Estratégia Adaptativa

No contexto de estratégias adaptativas, métodos de extensão têm sido preferencialmente escolhidos em relação a outros enfoques, conforme citado por Cavalcante-Neto [12,13]. Estes métodos incluem extensões h , p e r (Figura 3). Na extensão h a malha é refinada quando o indicador de erro excede uma tolerância pré-estabelecida de tal forma que se aumente o número de elementos da malha, diminuindo seus tamanhos, sem aumentar o grau dos seus polinômios. A extensão p é geralmente empregada em uma malha fixa. Neste caso se o erro de um determinado elemento exceder a tolerância pré estabelecida, então a ordem da função do elemento em questão é aumentada para reduzir o erro. A extensão r (de *relocação*) é empregada em um número fixo de nós e tenta mover os nós da malha para áreas que apresentem erros elevados, fazendo com que esta área fique mais refinada mas permanecendo com o mesmo número de elementos da malha original. Qualquer uma destas extensões pode ser também combinada em uma estratégia especial, como por exemplo, h - p , r - h , etc. A estratégia adaptativa utilizada neste trabalho foi a de extensão h .

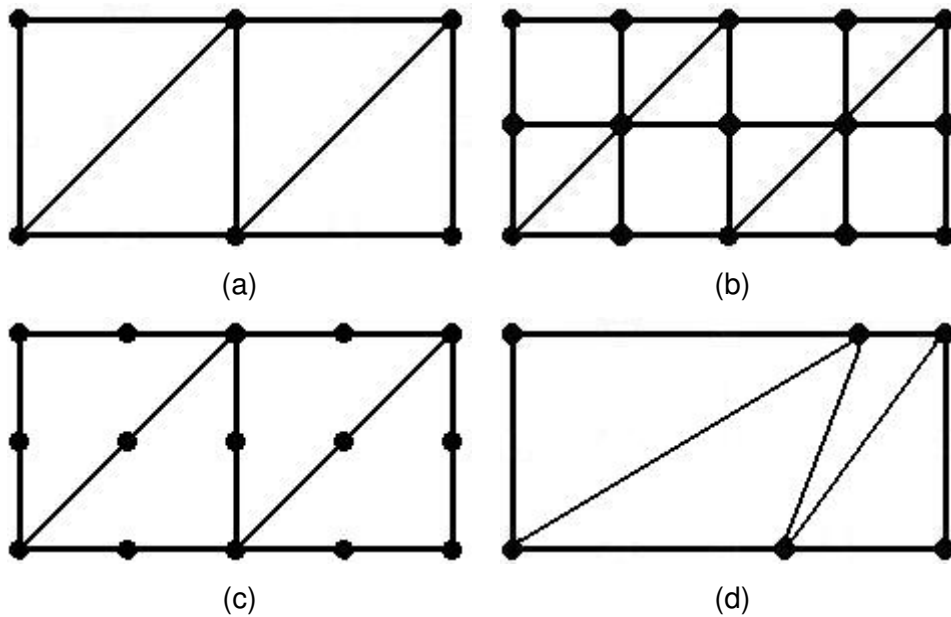


Figura 3 (a) malha original, (b) malha após o uso da estratégia do tipo h , (c) malha após o uso da estratégia do tipo p , (d) malha após o uso da estratégia do tipo r .

2.2. Estimativa de Erro

O conceito de adaptatividade e a própria essência do método dos elementos finitos estão intrinsicamente ligados à definição de erro, pois o método dos elementos finitos é um método numérico, ou seja, resolve problemas de forma aproximada e a idéia do uso da adaptatividade surgiu para melhorar as características das malhas que tivessem gerado resultados ruins na análise de elementos finitos. Além disso, é justamente a análise do erro que controla o critério de convergência do processo adaptativo.

Em geral existem dois tipos de estimativas de erro de discretização: *a-posteriori* e *a-priori* [12,57]. O que se procura, em estratégias *a-priori*, é tentar garantir de todas as formas que a malha gerada inicialmente já seja de boa qualidade, o que é válido e importante mas nem sempre é possível. Já a estratégia *a-posteriori* se preocupa em melhorar os resultados da análise de erro (qualidade da malha) a partir dos resultados de uma dada malha inicial. A estratégia utilizada neste trabalho é a *a-posteriori*, pois é a partir dos resultados da análise de elementos finitos da malha inicial que se obtém os parâmetros necessários para a geração da nova malha, resultando numa diminuição no valor do erro.

As análises de erros de discretizações têm se tornado cada vez mais importantes para aumentar a confiabilidade de modernos métodos numéricos aplicados a problemas de engenharia. Sempre que um método numérico, como o MEF, é usado para solucionar equações diferenciais relacionadas a um funcional definido para um problema discreto, erros são introduzidos pelo processo de discretização que reduz o modelo matemático contínuo para um modelo numérico discreto com um número finito de graus de liberdade.

Esta seção descreve de forma sucinta o procedimento normalmente adotado (Zienkiewicz [66]) para estimativa *a-posteriori* de erro de discretização em um modelo de elementos finitos.

Primeiramente será apresentada a formulação pelo MEF de um problema linear elástico com deslocamentos u definidos em um domínio Ω e com um contorno $\Gamma = \Gamma_u \cup \Gamma_t$, onde Γ_u é a parte do contorno em que os deslocamentos são prescritos e Γ_t é a parte do contorno em que os carregamentos são prescritos.

A equação abaixo apresenta a relação entre deslocamentos e deformações.

$$\varepsilon = Su \quad (1)$$

onde ε são as deformações, u são os deslocamentos e S é o operador diferencial que relaciona as deformações em função dos deslocamentos.

A equação abaixo apresenta a relação constitutiva.

$$\sigma = D\varepsilon \quad (2)$$

onde σ são as tensões e D é a matriz que define as tensões em função das deformações.

De posse das equações (1) e (2) pode-se colocar o problema segundo uma equação diferencial de equilíbrio a um carregamento q conforme a equação abaixo.

$$Lu - q \equiv S^T DSu - q = 0 \quad (3)$$

onde L é o operador diferencial auto-adjunto.

Fazendo-se uma transformação de coordenadas, pode-se obter uma relação entre as tensões σ em Ω com o carregamento t_p em Γ_t através da seguinte equação.

$$GDSu = t_p \quad (4)$$

onde G é um operador linear que relaciona o campo de tensões e as forças de superfície no contorno do domínio.

Numa aproximação por elementos finitos, tem-se

$$u \approx \hat{u} = N\tilde{u} \quad (5)$$

onde \tilde{u} é a solução numérica nodal para o campo de deslocamentos, N são as funções de forma, que relacionam deslocamentos nodais com os deslocamentos em qualquer ponto do interior do domínio de um elemento e \hat{u} é a solução, em termos de deslocamento, no interior do elemento finito. Obtém-se as equações de aproximação (por um processo padrão de Galerkin ou equivalentemente por minimização da energia potencial) para obter

$$K\tilde{u} - f = 0 \quad (6)$$

onde K é a matriz de rigidez e f é o vetor de forças nodais. Estes dois parâmetros são obtidos através das seguintes equações

$$K = \int_{\Omega} (SN)^T D(SN) d\Omega \quad (7)$$

$$f = \int_{\Omega} N^T q d\Omega + \int_{\Gamma} N^T t_p d\Gamma \quad (8)$$

Zienkiewicz [66] mostra em seu livro dois tipos distintos de estimadores de erro para problemas de elasticidade: os estimadores de erro baseados no processo de recuperação de *patches* e os fundamentados em resíduos. Mas antes de se elucidar as técnicas de recuperação utilizadas neste trabalho é necessário definir primeiramente o conceito de erro. Neste livro, Zienkiewicz, caracteriza o erro como sendo a diferença entre a solução exata e uma solução aproximada, conforme a equação abaixo.

$$e = u - \hat{u} \quad (9)$$

onde u é o resultado analítico e \hat{u} é o resultado aproximado. Neste contexto u é análogo a uma resposta (por exemplo, deslocamentos) em um procedimento típico de solução numérica.

De uma forma similar pode-se analisar o erro em análise estrutural através de comparações entre deformações ou tensões, tendo-se as seguintes equações como resultado.

$$e_\varepsilon = \varepsilon - \hat{\varepsilon} \quad (10)$$

$$e_\sigma = \sigma - \hat{\sigma} \quad (11)$$

onde ε é o valor exato das deformações, $\hat{\varepsilon}$ é o valor aproximado, e_ε é o valor do erro para deformações, σ é o valor exato das tensões, $\hat{\sigma}$ é o valor aproximado e e_σ é o erro referente às tensões.

Zienkiewicz [66] comenta que estas definições de erro podem muitas vezes levar a resultados ruins, pois para pontos abaixo de cargas pontuais os valores das deformações e tensões tenderiam ao infinito. Situações similares ocorrem em problemas que apresentam cantos, onde, como é bem conhecido, ocorrem singularidades nas tensões em uma análise elástica. Por estas razões é que foram introduzidos vários tipos de ‘normas’ para avaliar o erro da análise de elementos finitos. Estas normas podem ser expressas analiticamente de uma das três formas abaixo.

$$\|e\| = \left(\int_{\Omega} (Se)^T D(Se) d\Omega \right)^{1/2} = \left(\int_{\Omega} (e_\varepsilon)^T D(e_\varepsilon) d\Omega \right)^{1/2} = \left(\int_{\Omega} (e_\sigma)^T D^{-1}(e_\sigma) d\Omega \right)^{1/2} \quad (12)$$

Uma medida mais direta é chamada de norma L_2 , que pode ser associada com erros em qualquer quantificação. Para as tensões, a norma L_2 para o erro é

$$\|e\|_{L_2} = \left(\int_{\Omega} (e_\sigma)^T (e_\sigma) d\Omega \right)^{1/2} \quad (13)$$

Embora estas normas estejam definidas para todo o domínio, nota-se que o quadrado de cada norma pode ser obtido somando as contribuições dos elementos de tal forma que se tenha

$$\|e\|^2 = \sum_{i=1}^m \|e\|_i^2 \quad (14)$$

onde i representa um elemento e m o número total de elementos do modelo. Em uma malha considerada “ótima”, tenta-se fazer com que as contribuições para este quadrado da norma sejam iguais para todos os elementos. Entretanto, o valor absoluto da norma de energia ou da norma L_2 tem pouco significado físico. Desta forma, é preferível adotar um erro relativo

$$\eta = \frac{\|e\|}{\|u\|} \quad (15)$$

que pode ser determinado para todo o domínio Ω ou para subdomínios de elementos, onde $\|u\|$ é o valor positivo da raiz quadrada do dobro da energia de deformação, podendo ser expressa através da seguinte equação

$$\|u\| = \left(\int_{\Omega} (\sigma)^T D^{-1}(\sigma) d\Omega \right)^{1/2} \quad (16)$$

O erro relativo da norma de energia (η) é usado em estratégias adaptativas para avaliar o erro de discretização do problema em questão e para guiar para um refinamento que melhore a qualidade da malha e conseqüentemente os valores dos resultados obtidos. Em uma estratégia adaptativa de refinamento baseada em uma extensão h , o estimador de erro irá definir como o modelo discreto será refinado ou “desrefinado”. Um critério simples para se atingir o erro da solução dentro de um nível aceitável para todo o domínio pode ser estabelecido da seguinte forma

$$\bar{\eta} \leq \eta^* \quad (17)$$

onde η^* é o erro máximo permitido e $\bar{\eta}$ é o erro corrente da análise obtido por

$$\bar{\eta} = \frac{\|e\|}{\sqrt{\|\hat{u}\|^2 + \|e\|^2}} \quad (18)$$

onde $\|\hat{u}\|$ é a norma de energia obtida da solução por elementos finitos.

Um critério razoável utilizado para uma “malha ótima” é forçar que a norma de energia do erro ($\|e\|_i$) seja igualmente distribuída entre os elementos, desta forma para um dado elemento i , tem-se

$$\|e\|_i \leq \left(\eta^* \left[\frac{\|\hat{u}\|^2 + \|e\|^2}{m} \right]^{1/2} \right) = \bar{e}_m \quad (19)$$

onde m é o número total de elementos.

Pode-se então, desta forma, definir a razão de erro (ξ_i) como sendo

$$\xi_i = \frac{\|e\|_i}{\bar{e}_m} \quad (20)$$

É evidente que o refinamento será necessário se

$$\xi_i > 1 \quad (21)$$

Um procedimento mais eficiente consiste em designar completamente uma nova malha (estratégia adaptativa de extensão do tipo h) que satisfaça o seguinte requisito

$$\xi_i \leq 1 \quad (22)$$

No limite do refinamento da malha, assumindo uma certa taxa de convergência [66], o valor da razão de erro ξ_i é então usado para determinar o novo tamanho dos elementos que serão gerados na nova malha conforme a equação abaixo

$$h = \frac{h_i}{\xi_i^{1/p}} \quad (23)$$

onde h_i é o tamanho inicial do elemento, h é o tamanho final e p é a ordem polinomial da função de aproximação.

2.3. Métodos de Suavização

Esta seção descreve os procedimentos de recuperação de *patches* usados para se obter o campo $\bar{\sigma}$, já que o campo de tensões exato σ geralmente é desconhecido. Inicialmente, é descrita a técnica da suavização por médias nodais *Z2-HC* [62] e em seguida é mostrada a técnica *SPR* [63-66] fundamentada em sistemas de mínimos quadrados ponderados.

Vale ressaltar que neste trabalho a parte de implementação referente aos métodos de suavização encontram-se no programa chamado *FEMOOP* (Martha *et al* [43]), que está implementado seguindo uma programação orientada a objetos em C++.

O *FEMOOP* [43] é o módulo de análise utilizado tanto em 2D quanto em 3D. No caso 2D, o modelador geométrico utilizado para criar o modelo geométrico, criar e aplicar os atributos de análise às entidades geométricas, gerar as malhas de elementos finitos e responsável pela visualização dos resultados é o *MTOOL* [47]. Já para o caso 3D, o modelador geométrico é o *MG* [48] e visualizador é o *POS3D* [16].

2.3.1. Zienkiewicz e Zhu – Hinton e Campbell (Z2-HC)

Este método consiste em estimar valores nodais contínuos para nós internos da malha, podendo ser utilizada uma suavização global ou local^{*}. Uma breve descrição do método é apresentada abaixo.

Após se obter os resultados das tensões nos pontos de Gauss, deseja-se estimar uma representação global das tensões, de preferência na forma de valores médios nodais. Portanto os valores de tensões obtidos nos pontos de Gauss devem ser extrapolados para os nós e suavizados, o que pode ser feito de forma global ou local.

Na suavização global de tensões, os valores nodais de tensões são obtidos de forma a minimizar o erro global (de todo o modelo) na avaliação das tensões nos pontos de Gauss.

O procedimento usual, no entanto, é uma suavização local de tensões. Neste caso os valores das tensões nos pontos de Gauss são extrapolados para os nós de cada elemento. Após este passo os valores das tensões para um determinado nó são diferentes em cada um dos elementos que contém este nó. O passo seguinte, portanto, é a suavização dos valores nodais, o que normalmente é feito através de uma média dos valores obtidos de cada elemento adjacente ao nó. No entanto há casos em que deve existir uma descontinuidade na tensão, como por exemplo na interface entre dois materiais diferentes, onde deve haver um valor médio de tensão em um nó de interface para cada grupo de elementos que forem do mesmo material.

A extrapolação dos valores nos pontos de Gauss para os nós pode ser feita de duas formas: ou os valores são interpolados ou eles são obtidos através de um ajuste usando mínimos quadrados.

A Figura 4 mostra de uma forma simplificada a obtenção dos valores nodais através da contribuição de todos os elementos adjacentes ao nó, conforme a idéia do método.

^{*} para mais detalhes ver [26]

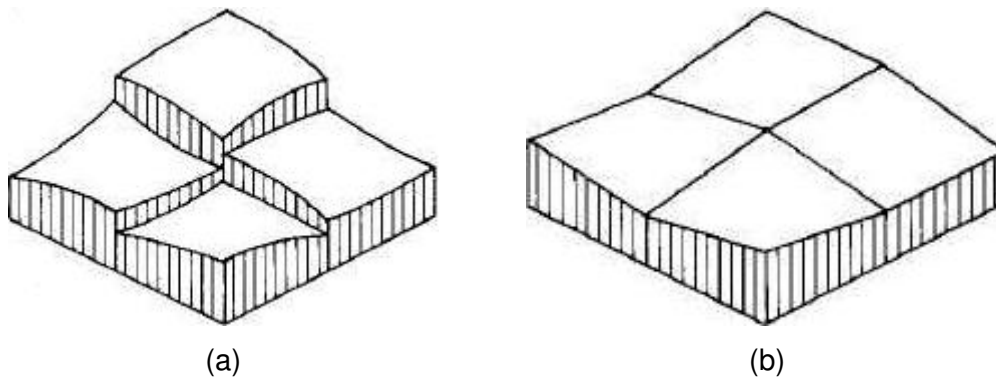


Figura 4 (a) mostra a contribuição de cada elemento, quadrilateral, para o valor de tensão, não suavizada, no nó adjacente aos mesmos, (b) mostra o valor da tensão suavizada no nó adjacente aos elementos quadrilaterais (Hinton & Campbell - [26]).

2.3.2. Superconvergent patch recovery (SPR)

Um campo genérico (por exemplo, tensão) pode ser aproximado pela seguinte expansão polinomial

$$\bar{\sigma} = Pa \quad (24)$$

onde P contém os termos polinomiais apropriados e a é um conjunto de parâmetros desconhecidos. Pode-se observar que esta expansão é usada para cada componente do tensor de tensões. Por exemplo, para problemas bidimensionais usando o elemento finito isoparamétrico quadrático de 8 nós (ver Figura 5.b) a seguinte aproximação é utilizada

$$P = [1, x, y, x^2, xy, y^2] \quad (25)$$

$$a = [a_0, a_1, a_2, a_3, a_4, a_5]^T \quad (26)$$

Os coeficientes desconhecidos a_i podem ser determinados por ajuste através do método dos mínimos quadrados ponderados da expansão polinomial (24) com relação aos valores de σ obtidos da solução de elementos finitos nos pontos de amostragem, isto é, $\hat{\sigma}$. Pequenos grupos (*patches*) de elementos são usados para realizar ajustes locais de mínimos quadrados e um peso (w_i) é considerado aqui para enfatizar a influência dos pontos de amostragem que estão mais próximos do nó que forma o *patch* (ver Figura 5.b). Desta forma, o peso (w_i) pode ser obtido de seguinte maneira

$$\varpi_i = \frac{1}{\rho_i^p}, \quad (27)$$

onde ρ_i é a distância Euclidiana entre o ponto de amostragem i e o nó que forma o *patch* e p é um inteiro. O uso de pesos maiores que zero pode ser efetivo para resolver problemas com gradientes bem elevados e também melhora o problema do mau condicionamento, que acontece em alguns problemas que usam o *SPR* [63-66].

Objetivando ilustrar a recuperação superconvergente discreta [66], considere um *patch* de elementos contendo m pontos de amostragem, como ilustrado na Figura 5, onde existe um ponto de amostragem i qualquer com coordenadas Cartesianas (x_i, y_i) em relação aos eixos globais. Assim sendo, o problema de mínimos quadrados ponderados é reduzido a minimização do seguinte funcional

$$\Lambda = \sum_{i=1}^m \varpi_i^2 [\hat{\sigma}(x_i, y_i) - \bar{\sigma}(x_i, y_i)]^2 \quad (28)$$

Substituindo a equação (24) na equação (28) obtém-se

$$\Lambda = \sum_{i=1}^m \varpi_i^2 [\hat{\sigma}(x_i, y_i) - P(x_i, y_i)]^2 \quad (29)$$

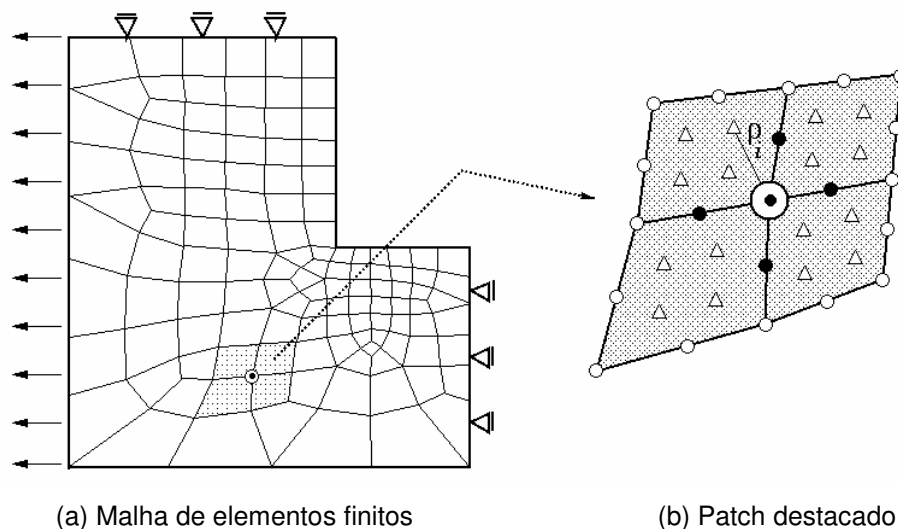


Figura 5 Notação da recuperação por patch: \odot nó que forma o patch; Δ pontos de amostragem; \bullet valores nodais determinados pelo procedimento de recuperação; \circ pontos nodais; ρ_i distância entre o ponto de amostragem i e o nó que forma o patch; Ω denota o domínio do problema e Ω_p indica o domínio do patch (Cavalcante Neto J.B.-1998 [13]).

A Figura 5 mostra uma malha gerada em um exemplo clássico da teoria da elasticidade, destacando a formação do patch.

O problema de minimização é resolvido fazendo-se $\partial\Lambda/\partial a = 0$, levando ao seguinte conjunto de equações lineares

$$Aa = b \quad (30)$$

onde

$$A = \sum_{i=1}^m \varpi_i^2 P^T(x_i, y_i) P(x_i, y_i) \quad (31)$$

$$b = \sum_{i=1}^m \varpi_i^2 P^T(x_i, y_i) \hat{\sigma}(x_i, y_i) \quad (32)$$

Resolvendo-se o sistema de equações em (30) para a , pode-se substituir seu valor na equação (24) e obter o valor das tensões.

2.4. Geração de Malhas

Esta seção apresenta as idéias gerais das principais técnicas de geração de malhas de elementos finitos. A técnica de geração utilizada neste trabalho foi, fundamentalmente, a mesma técnica usada no trabalho de Cavalcante-Neto [13], que combina duas das técnicas apresentadas, tal como descrito no próximo capítulo.

Usualmente, as malhas são classificadas em três grupos principais, os quais são: estruturadas, não-estruturadas e as híbridas. Entretanto, não há um consenso quanto à definição de cada tipo. Uma das sugestões é fazer a diferenciação baseada na topologia da vizinhança dos elementos. Desta forma, as malhas estruturadas são caracterizadas por seus nós internos terem um número constante de elementos adjacentes (ver Figura 6.a). Já as malhas não estruturadas não apresentam número de elementos adjacentes aos nós internos de forma constante (ver Figura 6.b). As malhas híbridas, como o próprio nome já diz, apresentam em algumas regiões do seu domínio características das malhas estruturadas e em outras partes, características das malhas não-estruturadas (ver Figura 6.c).

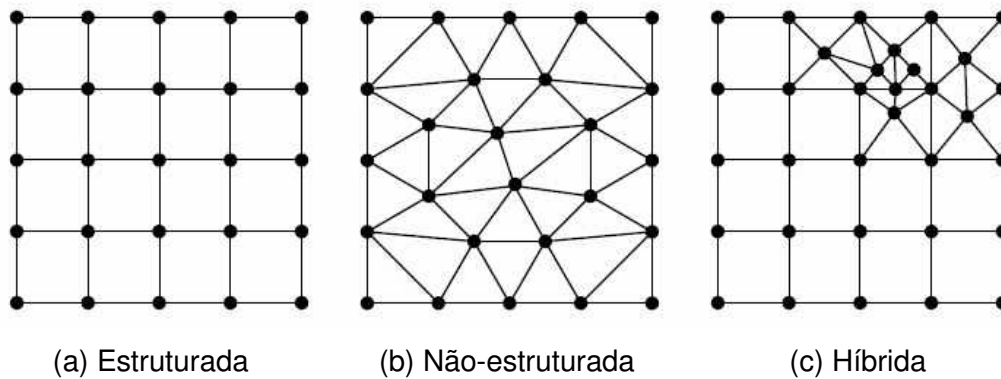


Figura 6 Tipos de malha (Batista V.H.F.-2005 [6]).

Malhas estruturadas discretizam o domínio em elementos possuindo conectividade implícita, ou seja, precisa-se apenas das coordenadas dos nós dos elementos para determinar todas as relações de conectividade existente, facilitando o processo de geração. Já nas malhas não-estruturadas as relações de conectividade entre os elementos têm que ser obtidas explicitamente através de uma tabela que especifique claramente a conectividade de cada elemento. Nas malhas híbridas a conectividade dependerá da combinação final entre as malhas estruturadas e as não-estruturadas.

As malhas não-estruturadas são preferíveis quando se tratar de discretização de domínios que apresentem geometrias arbitrárias (complexas), pois este tipo de malha se ajusta melhor ao contorno, permitem também a aplicação de refinamento local e adaptativo, o que é impossível em malhas estruturadas e muito difícil nas híbridas. As malhas híbridas também são usadas em domínios com geometria arbitrária, mas o processo de geração de malhas é mais complicado que o processo que usa malhas não-estruturadas.

Os elementos mais freqüentes em malhas estruturadas são quadriláteros em 2D e hexaedros em 3D. Nas malhas não-estruturadas são os triângulos e tetraedros, apesar de também ser possível utilizar elementos do tipo quadriláteros, prismas, pirâmides e hexaedros. Já nas malhas híbridas, podem ser usados elementos de formas variadas.

O uso preferencial de triângulos e tetraedros nas malhas não-estruturadas é justificado pelo fato de que estes elementos são capazes de se adaptar a contornos de domínios complexos e permitem suave transição de tamanho entre os elementos. Porém, a quantidade de elementos necessários para discretizar um domínio qualquer usando triângulos e tetraedros é maior quando comparado ao uso de quadriláteros e hexaedros. Apesar disso, a utilização de triângulos

para problemas 2D continuou sendo usada pelo fato de existirem inúmeros trabalhos de pesquisa eficientes tratando o processo de geração de malhas utilizando elementos triangulares [38]; para o caso 3D só em meados da década de 80 e início da, de 90 é que se começou a pesquisar sobre algoritmos de geração de malhas sólidas. Devido a este fato, hexaedros eram usados preferencialmente em detrimento aos tetraedros. Como resultado destas pesquisas já se têm um número razoável de algoritmos de geração de malhas 3D, porém poucos se destacam por possibilitar precisão, robustez, flexibilidade quanto às formas do domínio e elevado nível de automatização, além é óbvio, de resultar em malhas de “boa qualidade”. Carey [11] faz um resumo do avanço da pesquisa no estudo de técnicas de geração de malhas.

É de grande importância citar que esta seção do trabalho, foi baseada no trabalho de Batista [6], que apresentou uma enriquecedora revisão bibliográfica sobre as técnicas de geração de malhas, além de propor uma nova técnica para a geração de malhas não-estruturadas tetraédricas fundamentada em avanço de fronteira. Muitas referências citadas no trabalho de Batista [6] também foram pesquisadas como requisito básico para o entendimento do assunto abordado nesta seção, bem como para a elaboração deste trabalho.

2.4.1. Geração de malhas não-estruturadas

A capacidade de adequar-se a domínios arbitrários e a relativa facilidade de automatização impulsionaram a aplicação das malhas não-estruturadas na mecânica dos sólidos computacional, na dinâmica dos fluidos computacional e na geometria computacional. O grande interesse nesta área levou com que fossem desenvolvidos métodos que tratassem diferentes tipos de domínios (planos, superfícies tridimensionais e sólidos).

Os principais métodos de geração de malhas não-estruturadas são baseados na triangulação de Delaunay, Avanço de Fronteira e Decomposição Espacial Recursiva. Os métodos baseados na triangulação de Delaunay possuem um sólido embasamento matemático, mas apresentam dificuldades quando existem restrições tais como o contorno do domínio. Os métodos baseados no Avanço de Fronteira respeitam o contorno e têm um controle total da inserção dos nós internos. Já os métodos baseados na Decomposição Espacial Recursiva possibilitam acoplamento aos modeladores geométricos, aumentando o grau de automatização dos métodos.

2.4.1.1. Triangulação de Delaunay

A triangulação de Delaunay decompõe poliedros convexos gerados por conjuntos de pontos de modo único, satisfazendo ao critério da “esfera vazia”, que assegura que o círculo circunscrito a qualquer triângulo pertencente a triangulação não contenha pontos em seu interior (ver Figura 7).

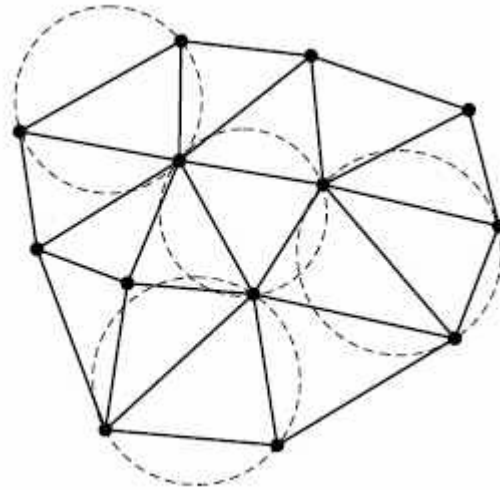


Figura 7 Malha de elementos finitos gerada por triangulação de Delaunay (Batista [6]).

Objetivando forçar a presença de arestas em triangulações de Delaunay, Chew [17] desenvolveu um algoritmo que constrói a chamada triangulação de Delaunay com restrição. A partir deste instante, diferentes algoritmos surgiram, servindo de base para novos algoritmos de geração de malhas.

Cavendish *et al* em 1985 [15] foram os primeiros a aplicarem a triangulação de Delaunay para a geração de malhas tridimensionais.

As principais características de geradores de malhas fundamentados na triangulação de Delaunay são: a produção de mais de um elemento por nó inserido, problemas de arredondamento no teste da “esfera vazia” e o surgimento de *slivers*, que são tetraedros cujos os quatro vértices estão praticamente no mesmo plano.

2.4.1.2. Decomposição Espacial Recursiva

O emprego desta técnica permite a representação de domínios com formas complexas de modo compacto e hierárquico. A idéia básica envolvida é similar a um dos paradigmas fundamentais na construção de algoritmos (*dividir para conquistar*). *Quadtrees* e *octrees* são exemplos de estruturas baseadas neste conceito e que permitem a modelagem geométrica de domínios bi e tridimensionais. Yerry & Shephard [60,61] apresentaram um algoritmo para geração de malhas 2D, onde as malhas são geradas a partir de *quadtrees*. O processo pode ser resumido da seguinte forma (vide Figura 8). Primeiramente define-se um quadrado englobando todo o domínio, que será a primeira célula da *quadtree*. Em seguida este quadrado é dividido em quatro partes iguais. Cada parte (nova célula) pode ser classificada da seguinte forma: cheia quando o quadrado estiver inteiramente dentro do domínio, vazia quando nenhuma parte do quadrado estiver dentro do domínio ou parcial quando pelo menos uma parte da célula estiver dentro do domínio. Cada célula nova é subdividida em outras quatro células, que também serão classificadas de acordo com o mesmo critério descrito anteriormente. Este procedimento continua até se atingir o nível de refinamento desejado, ou no caso adaptativo, até se atingir o erro requerido. Após se atingir este critério de parada da subdivisão, as células do tipo parcial são seccionadas para satisfazer o contorno e a *quadtree* é modificada para forçar um único nível de diferença entre células adjacentes, o que suaviza a transição entre os tamanhos dos elementos. Finalmente converte-se os quadrados em triângulos e ajusta-se os pontos do contorno. Estes ajustes são responsáveis pela distribuição irregular dos pontos ao longo do contorno, o que é uma das características do método.

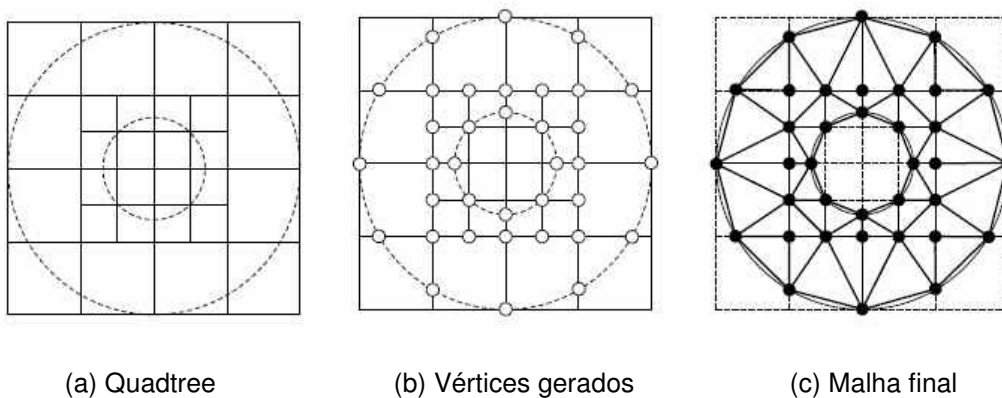


Figura 8 Exemplo de geração de uma malha 2D de elementos finitos usando uma *quadtree* (Batista [6]).

A Figura 9 mostra a *quadtree* em forma de diagrama de árvore referente à *quadtree* mostrada na Figura 8.a, com as células que estão parcialmente dentro do domínio, células que estão totalmente dentro e células que estão fora do domínio.

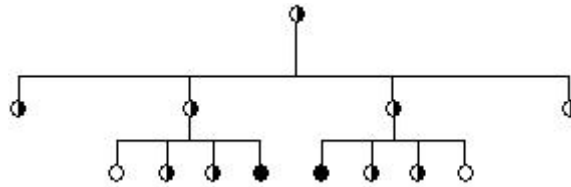


Figura 9 Estrutura *quadtree* em forma de diagrama.

2.4.1.3. Avanço de Fronteira

Esta técnica foi concebida no trabalho de Lo em 1985 [38] surgindo com o intuito de gerar malhas em domínios bidimensionais arbitrários. Inicialmente o contorno do domínio é representado pela união disjunta de arestas delimitando um polígono fechado simples. Os nós pertencentes ao contorno externo são ordenados em sentido anti-horário. Já os nós internos são numerados em sentido contrário. Devido a esta ordenação fica definido que a área a ser triangulada fica sempre à esquerda das arestas do domínio. Antes de se iniciar a discretização são calculadas as coordenadas dos nós internos do domínio obedecendo o espaçamento nodal estabelecido. A etapa de triangulação inicia com a definição de um fronte de geração exatamente igual ao contorno do domínio. Então selecionam-se dois nós do domínio que sejam capazes de gerar um elemento com cada aresta do fronte. O nó que resultar em um elemento com “melhor qualidade”, segundo o critério aplicado, será o nó escolhido. Desta forma o fronte vai avançando até que não haja mais aresta que não pertença a nenhum elemento. A Figura 10 ilustra a descrição do método feita acima.

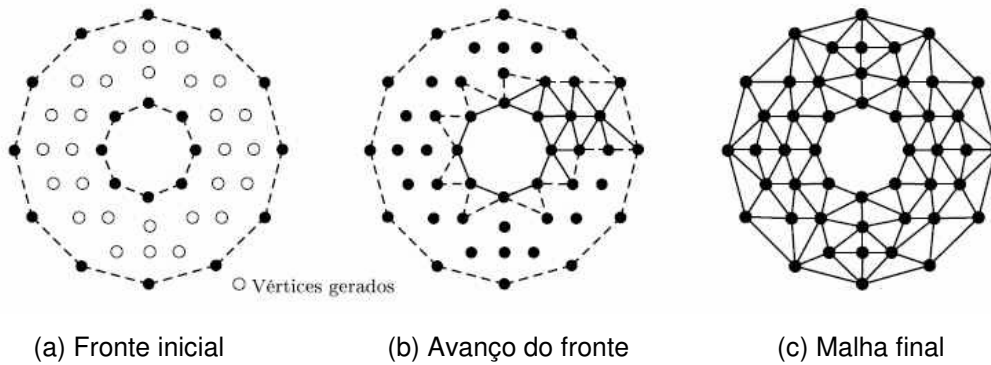


Figura 10 Exemplo de geração de uma malha 2D de elementos finitos usando avanço de fronteira (Batista V.H.F.-2005 [6]).

Apesar da concepção do algoritmo ser atribuída a Lo foi o algoritmo idealizado por Peraire *et al* em 1987 [50] que popularizou o método, pois a inovação proposta por Peraire *et al* permite maior controle na inserção de nós e o acoplamento a processos adaptativos. Nesse trabalho, Peraire *et al* montam uma lista com nós candidatos para formar o elemento, com um ponto ideal e com os nós vizinhos à aresta selecionada. O nó escolhido será aquele que formar um elemento, cujo o interior não contenha outro nó da lista dos candidatos e cuja as arestas não interceptem nenhuma aresta do fronte.

3 PROCESSO ADAPTATIVO

Este capítulo descreve o processo adotado para a geração adaptativa de malhas de elementos finitos 2D e 3D utilizada neste trabalho. Conforme mencionado anteriormente, este processo é fundamentado no algoritmo para geração de malhas volumétricas desenvolvido por Cavalcante-Neto [13] e no algoritmo de geração de malhas de superfície desenvolvido por Miranda *et al* [45]. Portanto, este capítulo objetiva também descrever sucintamente as técnicas de geração de malhas volumétricas e de superfície. A versão 2D do algoritmo é um caso particular dessas técnicas [44].

3.1. Resumo do processo adaptativo

A estratégia adaptativa utilizada neste trabalho é a do tipo h com estimativa de erro *a-posteriori*. Os métodos de suavização de respostas (estimativa de resposta “melhorada”), tal como descrito no capítulo anterior, são o *Z2-HC* [62], baseado na média nodal, e o *SPR* [63-66], baseado em recuperação de *patches*. A técnica de geração de malhas é baseada em *Decomposição Espacial Recursiva* (seção 2.4.1.2) e *Avanço de Fronteira* (seção 2.4.1.3), usando os algoritmos desenvolvidos por Cavalcante-Neto e Miranda *et al*, mas com uma nova abordagem da técnica de *Decomposição Espacial Recursiva*, pois as estruturas de dados do tipo *quadtree* (2D) e *octree* (3D) não têm mais apenas a função de gerar pontos no interior do domínio. Nas versões anteriores, essas estruturas de dados eram utilizadas para cada região do domínio (sub-domínio) de forma independente (uma estrutura por região). A partir deste trabalho, estas estruturas são globais, isto é, existe apenas uma estrutura *quadtree* ou *octree* para gerenciar o processo de geração adaptativa de malhas de um modelo (mesmo que tenha várias regiões). Além disso, essas estruturas agora são responsáveis pelo refinamento das curvas e superfícies, além de controlar o tamanho dos elementos que são gerados no domínio. A Figura 11 mostra o fluxograma do processo adaptativo apresentado neste trabalho.

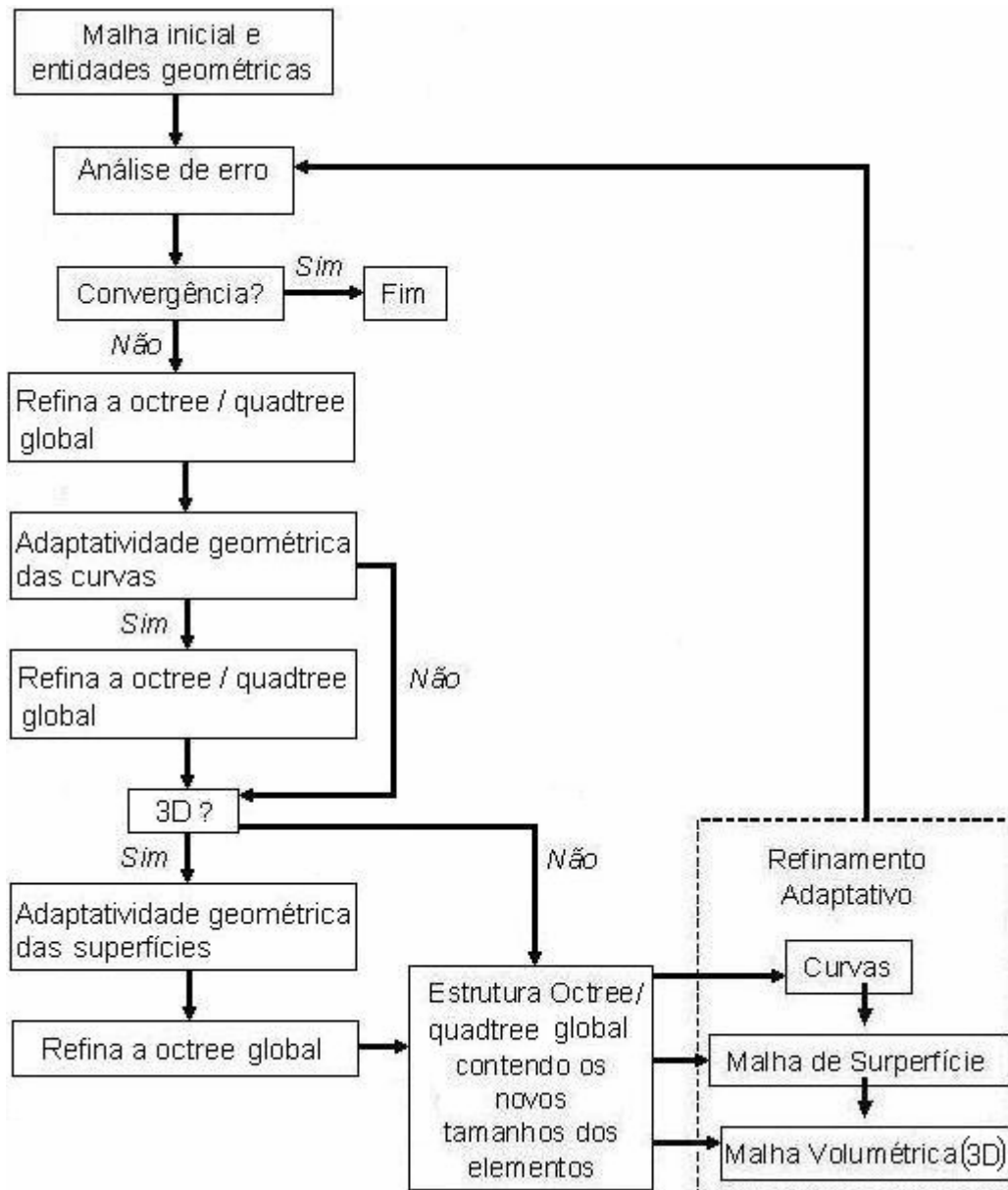


Figura 11 Fluxograma do novo processo adaptativo (2D e 3D).

A descrição do processo pode ser resumida da seguinte forma. Primeiramente, por se usar uma estimativa de erro *a-posteriori*, parte-se de um modelo sólido ou plano, onde já exista uma malha de elementos triangulares / tetraédricos associada. Posteriormente, seleciona-se o método de suavização que se deseja utilizar. Em seguida, o programa de análise (*FEMOOP* [43]) lê o arquivo gerado pelo modelador e gera um arquivo que contém informações das razões de erro associados a cada elemento (ξ_i) (ver equação 20) de acordo com o método de suavização adotado. E são estes valores que são usados para refinar a *octree / quadtree* global.

Considera-se que existe um modelador geométrico associado ao ambiente de geração adaptativa de malhas de elementos finitos. O modelo geométrico plano tem uma descrição topológica dos vértices, curvas e regiões planas, assim como uma descrição geométrica associada, que consiste das coordenadas dos vértices e representação matemática das curvas. Em 3D, a descrição topológica envolve vértices, curvas, superfícies e regiões; e a descrição geométrica também considera a representação matemática das superfícies. Tanto no caso plano quanto no tridimensional o modelo pode conter várias regiões. Neste ambiente, os atributos da simulação, tais como propriedades de materiais, cargas e restrições de suporte, estão associados às entidades geométricas. Dessa forma, as entidades da malha de elementos finitos (nós e elementos) recebem automaticamente os atributos das entidades geométricas que estão associadas. Com isso, pode-se regerar as malhas sem que os atributos sejam perdidos. Neste trabalho, o modelador utilizado é o *MTOOL* [47] para o caso 2D e o *MG* [48] para o caso 3D.

A razão de erro de cada elemento e as curvaturas das curvas e superfícies vão definir os tamanhos dos novos elementos que serão gerados no próximo passo do processo adaptativo. A distribuição espacial do tamanho dos novos elementos é armazenada em uma estrutura de dados global do tipo *octree* (3D) / *quadtree* (2D).

A idéia de se utilizar essas técnicas de decomposição espacial recursiva para armazenar a distribuição espacial dos tamanhos dos elementos em um processo de geração de malhas de elementos finitos não é nova. Cavalcante-Neto [13] e Miranda [44,45] lançaram esta idéia. A novidade apresentada é o fato de a *octree* / *quadtree* ser global, isto é, uma única estrutura de dados é criada para todo o modelo, mesmo que ele tenha várias regiões. Nos trabalhos citados uma *octree* / *quadtree* era gerada para cada região do modelo. Além disso, a *octree* / *quadtree* global contém as informações sobre os tamanhos dos elementos necessários para reduzir o erro de discretização (adaptatividade numérica) e sobre os tamanhos dos elementos necessários para modelar adequadamente as curvaturas de curvas e superfícies (adaptatividade geométrica).

A próxima seção descreve a construção da *octree* / *quadtree* global em função da razão de erro dos elementos finitos da malha anterior. A seção 3.3 descreve o refinamento da *octree* / *quadtree* global de acordo com as curvaturas das curvas. No caso 3D, a *octree* é refinada subseqüentemente em função da geometria das superfícies tal como descrito na seção 3.4.

Uma vez obtida a *octree* / *quadtree* global levando em conta o erro numérico e as curvaturas de curvas e superfícies, o passo seguinte é a geração da malha de elementos finitos propriamente dita. A geração de malhas é feita por regiões e parte da discretização das curvas (seção 3.6) para gerar as malhas de superfície no caso 3D (seção 3.7) e das malhas de domínio no caso 2D. No caso 3D, a geração da malha sólida (seção 3.8) de uma região depende da discretização das suas curvas e superfícies de bordo.

A necessidade de ordenar a discretização começando pelas curvas, passando pelas superfícies e finalizando com as regiões se deve a dois motivos. Primeiro, por se adotar uma técnica de Avanço de Fronteira, que parte do contorno discretizado de cada região. Segundo, a discretização prévia das fronteiras das regiões força a compatibilidade entre as malhas das regiões.

Após se realizar um passo do processo adaptativo, é mostrado o resultado do erro corrente, identificando se é necessário ou não mais um passo.

3.2. Construção da *octree* / *quadtree* global baseada no erro numérico

O primeiro cubo / quadrado da *octree* / *quadtree* global é inicializado com a menor dimensão que é capaz de envolver completamente o modelo (vide seção 2.4.1.2).

Após se inicializar a *octree* / *quadtree* global, calcula-se, de acordo com o (a) volume (área) de cada elemento e com a razão de erro (ξ_i) associada a ele, o tamanho característico h que será associado ao seu centro geométrico. Este tamanho característico será usado no refinamento da *octree* / *quadtree* e é calculado da seguinte forma. Primeiramente, calcula-se o volume / área real do elemento (V_{REAL}, A_{REAL}). Em seguida calcula-se o comprimento L da aresta (lado) do tetraedro (triângulo) regular (equilátero) que tem o (a) mesmo (mesma) volume (área) do elemento. A aresta L é adotada como o tamanho característico h_i do elemento na malha inicial. Esse cálculo é feito com base nos tetraedros / triângulos regulares / equiláteros pois estas são as formas ideais dos elementos que serão gerados na nova malha.

O valor de L para o caso plano é obtido da seguinte equação, através da relação entre a área do triângulo equilátero e o seu lado.

$$L = \sqrt{\frac{4A_{REAL}}{\sqrt{3}}} \quad (33)$$

O valor de L para o caso tridimensional é obtido da seguinte equação, que é obtida pela relação entre o volume do tetraedro regular e sua aresta.

$$L = \sqrt[3]{\frac{12V_{REAL}}{\sqrt{2}}} \quad (34)$$

O tamanho característico h que será associado ao centro geométrico de cada elemento será o valor de L obtido pelas equações (33) ou (34) dividido pelo valor ξ_i do elemento, de acordo com a equação (23), considerando $h_i = L$.

O refinamento da *octree* / *quadtree* é realizado localizando a célula da *octree* / *quadtree* onde se encontra o centro geométrico de cada elemento e testando se o tamanho desta célula é maior que o tamanho h . Em caso afirmativo a célula da *octree* / *quadtree* será dividida em 8 cubos / 4 quadrados iguais. O refinamento termina quando os centros geométricos de todos elementos estiverem localizados em células, cujos os tamanhos são menores ou iguais aos tamanhos característicos dos elementos correspondentes.

3.3. Refinamento da *octree* / *quadtree* global baseado na adaptatividade geométrica das curvas

Outra ferramenta muito importante idealizada e implementada durante o desenvolvimento deste trabalho foi a adaptatividade geométrica das curvas de bordo. Em muitas ocasiões, quando se usa o processo adaptativo apenas baseado no erro numérico da malha, a nova malha gerada (no passo seguinte do processo) não respeita a geometria real das curvas do modelo. Isto acontece, sobretudo, quando estas curvas possuem trechos com curvaturas acentuadas e não há nesses trechos concentrações de tensões que requeiram maior refinamento dos mesmos. Nesses casos, utiliza-se o processo adaptativo baseado nos estimadores de erro juntamente com a adaptatividade geométrica, que busca refinar mais os trechos com maiores curvaturas para que as geometrias das curvas sejam respeitadas.

Devido a isso, após ser inicializada e refinada a *octree* / *quadtree* global através do erro numérico, o processo adaptativo tem a opção de considerar as curvaturas das curvas do modelo no refinamento da *octree* / *quadtree* global.

O processo adaptativo considerando também a adaptatividade geométrica de curvas depende de alguns parâmetros que são fornecidos como dados de entrada.

- 1) Tolerância angular (ang_{TOL});
- 2) Distância máxima, medida ao longo da curva, entre duas subdivisões consecutivas pertencentes à curva (d_{MAX});
- 3) Distância mínima, medida ao longo da curva, entre duas subdivisões consecutivas pertencentes à curva (d_{MIN}).

O refinamento da *octree* / *quadtree* global considerando esses parâmetros para cada curva é feito de forma recursiva. A curva é subdividida em trechos sobre os quais se efetuam testes que, se forem satisfeitos, irão resultar no acréscimo de uma nova subdivisão à curva. Se estes testes falharem, o trecho é subdividido em dois novos trechos de comprimentos iguais, e os testes são novamente realizados para cada um destes trechos, recursivamente.

Inicialmente, utiliza-se a curva inteira como primeiro trecho a ser testado. Em seguida, se necessário, a curva vai sendo subdividida até que os critérios adotados sejam todos satisfeitos para cada trecho testado.

Usando-se a Figura 12 como referência, pode-se descrever, em ordem, os testes efetuados para cada trecho:

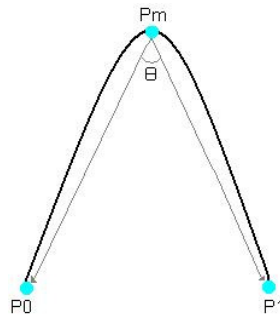


Figura 12 Curva a ser subdividida segundo os critérios geométricos.

- 1) Compara-se entre o comprimento total do trecho, medido ao longo da curva (Figura 12), e a distância mínima permitida entre duas subdivisões consecutivas. Se o comprimento total do trecho for menor ou igual à distância mínima permitida, o ponto P0 é acrescentado como uma nova subdivisão da curva (Figura 13.a). Se o comprimento total do trecho for maior que a distância mínima permitida, passa-se ao teste seguinte.

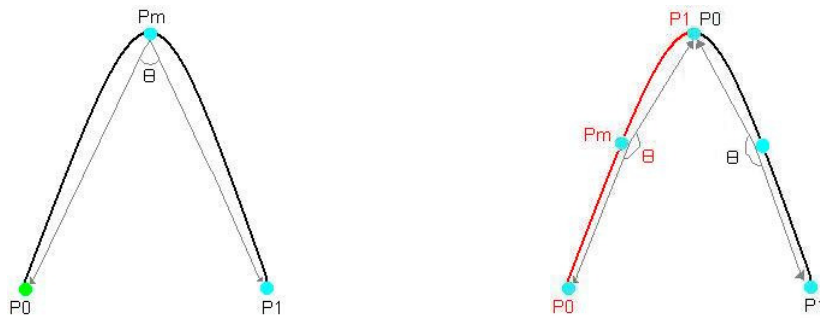


Figura 13 (a) Ponto P_0 é inserido (b) Inseri-se uma nova subdivisão, fazendo-se $P_1 = P_m$ e P_m sendo a metade do novo trecho.

2) Compara-se entre o comprimento total do trecho, medido ao longo da curva, e a distância máxima permitida entre duas subdivisões consecutivas. Se o comprimento total do trecho for maior que a distância máxima permitida, o trecho deve ser subdividido em dois novos trechos de igual comprimento sobre os quais os mesmos testes descritos aqui serão aplicados (Figura 13.b). Isto é obtido fazendo-se $P_1 = P_m$ para o primeiro trecho (em vermelho) e $P_0 = P_m$ para o segundo trecho (em preto), e recalculando-se os pontos intermediários para cada trecho (pontos localizados na metade do comprimento de cada trecho). Se o comprimento total do trecho for menor que a distância máxima permitida, passa-se ao teste seguinte.

3) Calcula-se o ângulo entre os vetores cujos vértices iniciais encontram-se no ponto localizado na metade do comprimento do trecho em questão (ponto P_m) e cujos vértices finais encontram-se sobre os pontos inicial e final do trecho (P_0 e P_1). Calcula-se, então, o módulo da diferença entre o valor deste ângulo e 180° . Se esta diferença for menor que a tolerância angular pré-estabelecida, o ponto P_0 é acrescentado como uma nova subdivisão da curva (Figura 13.a). Se esta diferença for maior que a tolerância angular pré-estabelecida, o trecho é subdividido em dois novos trechos de comprimento iguais (Figura 13.b) sobre os quais os mesmos testes descritos aqui serão aplicados.

Os critérios da adaptatividade geométrica podem ser estabelecidos pelo usuário do programa, de tal forma, que os trechos das curvas que apresentem maiores curvaturas serão mais refinados. Este fato também se torna muito importante para adaptatividade baseada em erro, pois, em geral, são justamente estes trechos que normalmente apresentam grande gradiente de tensões.

Neste trabalho, a adaptatividade geométrica é utilizada no refinamento da *octree* / *quadtree* global da seguinte forma:

- 1) Calcula-se o número de subdivisões da curva;

2) De cada trecho obtém-se as coordenadas Cartesianas do seu centro geométrico e o tamanho do trecho;

3) Estes valores são repassados para o refinamento da *octree* / *quadtree* global. Isto é feito de forma análoga ao que foi descrito na seção anterior para refinar a *octree* / *quadtree* de acordo com o tamanho característico requerido pela razão de erro numérico.

A Figura 14.a, 14.b, 14.c e 14.d tentam ilustrar a importância do uso da adaptatividade geométrica. A Figura 14.a mostra um tubo submetido a pressão interna (no ambiente de modelagem), a Figura 14.b mostra o mesmo tubo representado por uma malha de elementos finitos bem grosseira, a Figura 14.c mostra o resultado após o primeiro passo da adaptatividade usando apenas critérios baseados na estimativa de erro, percebe-se nesta figura, que houve uma melhora na qualidade da malha em comparação com a malha da Figura 14.b, já a Figura 14.d mostra o resultado após o primeiro passo da adaptatividade usando os critérios baseados na estimativa de erro e a adaptatividade geométrica, percebe-se que nesta figura há uma melhora na qualidade da malha em comparação com as malhas das Figura 14.b e 14.c.

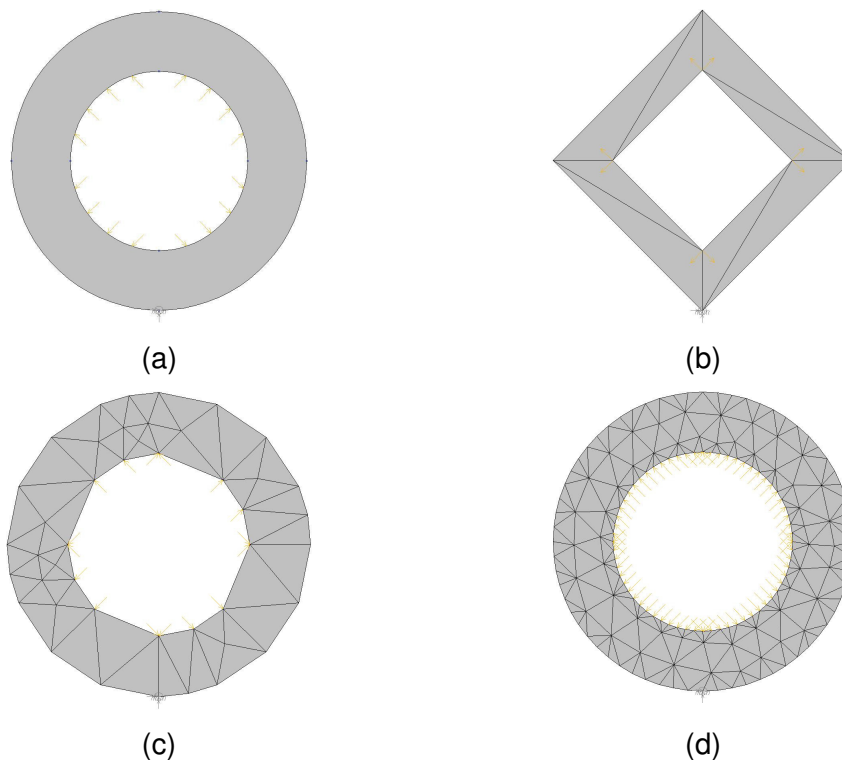


Figura 14 (a) Tubo com pressão interna, (b) Malha grosseira do modelo do tubo, (c) resultado do 1º passo do processo adaptativo usando apenas critérios baseados na estimativa de erro e (d) resultado do 1º passo do processo adaptativo usando os critérios de estimativa de erro e os critérios geométricos.

3.4. Refinamento da octree global baseado na adaptatividade geométrica das superfícies

No caso 3D, um ponto muito importante que deve ser considerado é a geração de malhas em superfícies de forma arbitrária, levando em consideração a curvatura das mesmas. Dessa forma, após o refinamento da *octree* global considerando os critérios da análise de erro e usando-se os critérios da adaptatividade geométrica das curvas, deve-se refinar a *octree* global considerando também as curvaturas das superfícies. Para tanto, usou-se neste trabalho o algoritmo desenvolvido por Miranda e Martha [45], que foi uma extensão da técnica de geração de malhas proposta por Cavalcante-Neto [13].

Nesse algoritmo a malha é gerada no espaço paramétrico da superfície, mas ângulos e distâncias são calculados no espaço Cartesiano 3D. Desta forma, os elementos finitos podem ser distorcidos no espaço paramétrico. Mas quando são mapeados ao espaço Cartesiano 3D devem apresentar boa razão de aspecto. O algoritmo usa um mapeamento de métrica desenvolvido por Tristano [59] para obter as distâncias e distorções corretas entre os dois espaços (Figura 15).

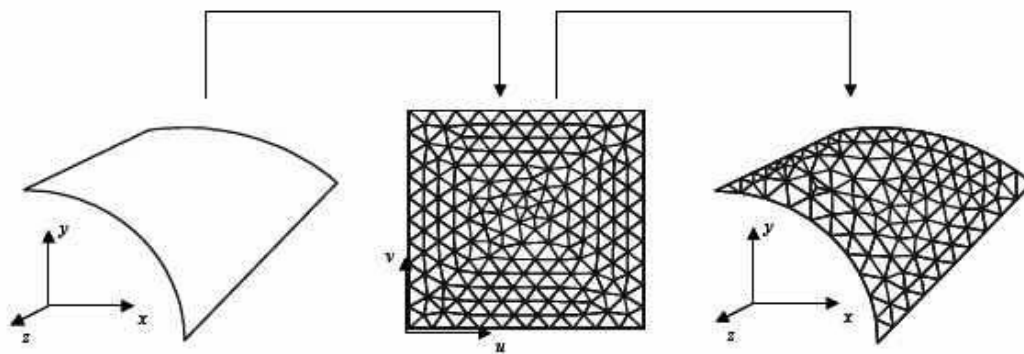


Figura 15 Processo de geração de malhas superficiais paramétricas em 3D (Lira [37]).

Tristano *et al* também desenvolveram um algoritmo de geração de malhas de superfície, porém a principal diferença entre os dois algoritmos está na estratégia de como guardar as informações de métricas entre os espaços Cartesiano e Paramétrico. No trabalho de Tristano [59] usa-se uma triangulação auxiliar, já no algoritmo desenvolvido por Miranda e Martha [45] usa-se uma *quadtree* auxiliar para guardar estas informações.

Esse trabalho mantém a idéia de Miranda e Martha de utilizar a *quadtree* auxiliar das superfícies para guardar as informações de métricas. Uma alternativa seria guardar essas informações na *octree* global. Entretanto, se teria dificuldade em saber que informações estariam associadas a quais superfícies, quando uma mesma célula da *octree* global contivesse trechos de mais de uma superfície. Por essa razão cria-se a *quadtree* auxiliar, mas os valores dos tamanhos característicos h que controlam os tamanhos dos elementos a serem gerados na malha de superfície, continuam sendo guardados na *octree* global.

No trabalho de Tristano [59] também se usa uma *quadtree* no espaço paramétrico, mas ela tem apenas a função de gerar pontos no interior do domínio para depois, com a técnica de Delaunay, gerar a malha. Já no trabalho de Miranda e Martha [45] os pontos no interior do domínio são inseridos concomitantemente aos elementos da malha, utilizando uma técnica da avanço de fronteira que é descrita nas seções 3.7 e 3.8.

Conforme será descrito nessas seções, a descrição do contorno das superfícies serve como entrada de dados para o algoritmo de Miranda e Martha [45]. A definição destes pontos não é tarefa do algoritmo. Portanto, é necessário que as curvas do contorno das superfícies também apresentem uma discretização que considerem as curvaturas presentes. É neste contexto, que se percebe o importante acréscimo de se considerar a adaptatividade geométrica das curvas tal como mostrado na seção anterior.

O refinamento da *octree* global considerando as curvaturas das superfícies é feito da seguinte maneira. Para cada superfície gera-se uma *quadtree* auxiliar com base na discretização das curvas de bordo e de suas curvaturas. Neste estágio não é necessário gerar uma malha associada à superfície, pois o objetivo é apenas o refinamento da *octree* global. Cada *quadtree* auxiliar é refinada conforme o algoritmo de Miranda *et al* da seguinte forma. A subdivisão das curvas de bordo da superfície determina a profundidade local da *quadtree* e posteriormente a *quadtree* é refinada para evitar que células do interior sejam maiores que células que contenham as curvas de bordo. Em seguida a *quadtree* é refinada para forçar uma disparidade entre células vizinhas de no máximo um nível de profundidade na decomposição recursiva. O último passo é o refinamento da *quadtree* para forçar a mínima diferença de curvatura entre células adjacentes.

Neste último passo, calculam-se os vetores normais de cada célula da *quadtree* auxiliar. Os vetores normais de duas células adjacentes, N_A e N_B , são

avaliados no centro geométrico das mesmas. Calcula-se a inclinação (θ) entre as células adjacentes através do produto escalar entre os seus vetores normais.

$$\cos(\theta) = \frac{N_A \cdot N_B}{\|N_A\| \|N_B\|} \quad (35)$$

O valor de $\cos(\theta)$ encontrado é comparado com o valor mínimo $\cos(\theta)_{MIN}$.

$$\cos(\theta) < \cos(\theta)_{MIN} \quad (36)$$

Se o critério da inequação (36) não for atendido, deve-se, então, calcular um novo tamanho de célula H_{NEW} , baseado no tamanho antigo H_{OLD} , $\cos(\theta)$ e $\cos(\theta)_{MIN}$, onde o valor de θ_{MIN} sugerido por Miranda *et al* é de 30°.

$$H_{NEW} = \frac{H_{OLD}}{\cos(\theta)_{MIN}} \cos(\theta) \quad (37)$$

Para ilustrar a geração da *quadtree* auxiliar, a Figura 16 mostra uma superfície 3D sobre a qual será feita a geração da malha com as suas curvas de bordo já tendo sido refinadas segundo os critérios geométricos estabelecidos, de tal forma que os trechos de maior curvatura são mais discretizados.

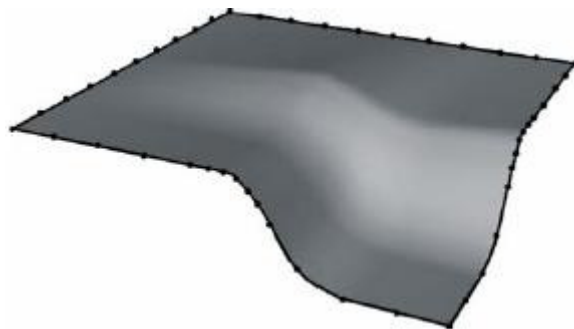


Figura 16 Superfície com as suas curvas de bordo considerando os parâmetros da adaptatividade geométrica das curvas.

A Figura 17 ilustra a importância de se considerar a curvatura da superfície na geração de malhas. A Figura 17.a mostra uma malha sem considerar as curvaturas das curvas de bordo e da superfície. A Figura 17.b mostra a *quadtree* auxiliar construída para realizar a adaptatividade geométrica da superfície e a Figura 17.c mostra a malha quando se considera a curvatura.

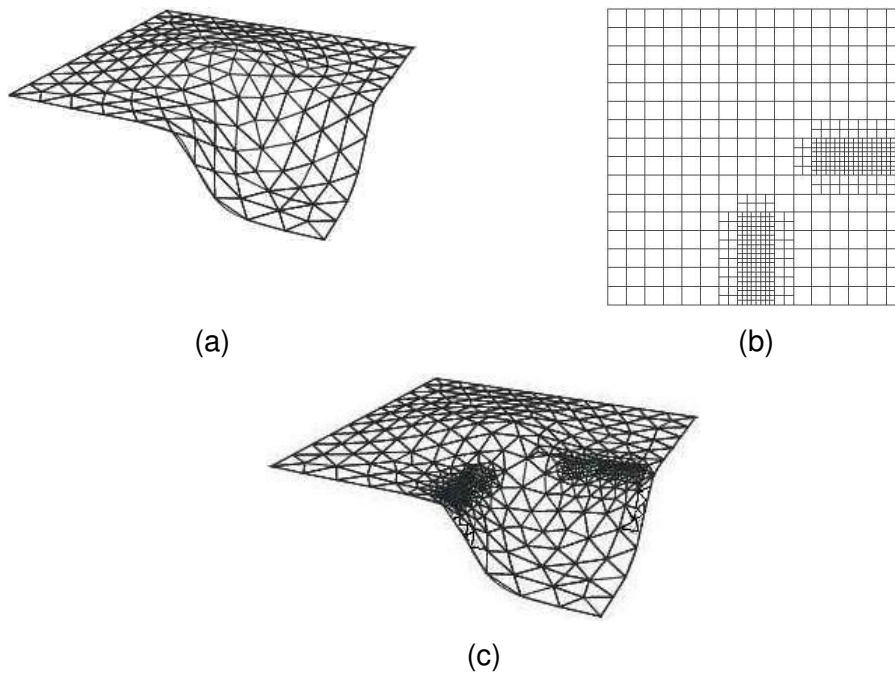


Figura 17 Exemplo de uma malha em uma superfície 3D: (a) sem considerar informações sobre curvaturas; (b) *quadtree* auxiliar construída para realizar a adaptatividade geométrica da superfície da figura (c) e (c) malha da superfície após se considerar as suas curvaturas (Lira [37]).

Finalmente, após a construção da *quadtree* auxiliar da superfície, é realizado o refinamento da *octree* global. Isto é efetuado de forma análoga ao que foi feito para refinar a *octree* global devido ao erro numérico e à curvatura das curvas. Para cada célula da *quadtree*, localiza-se a célula da *octree* global que contém o seu centro. Refina-se a célula da *octree*, forçando que o seu tamanho fique menor ou igual ao tamanho da célula da *quadtree* auxiliar.

3.5. Refinamento da *octree* / *quadtree* global para forçar disparidade de tamanho mínima

Em seguida a *octree* / *quadtree* é refinada para manter, entre duas células vizinhas, apenas um nível de diferença de profundidade na decomposição recursiva, evitando uma má transição entre os elementos da malha a ser gerada. A Figura 18 mostra um exemplo bidimensional com esse ajuste, onde as células marcadas em “x” serão refinadas conforme este critério.

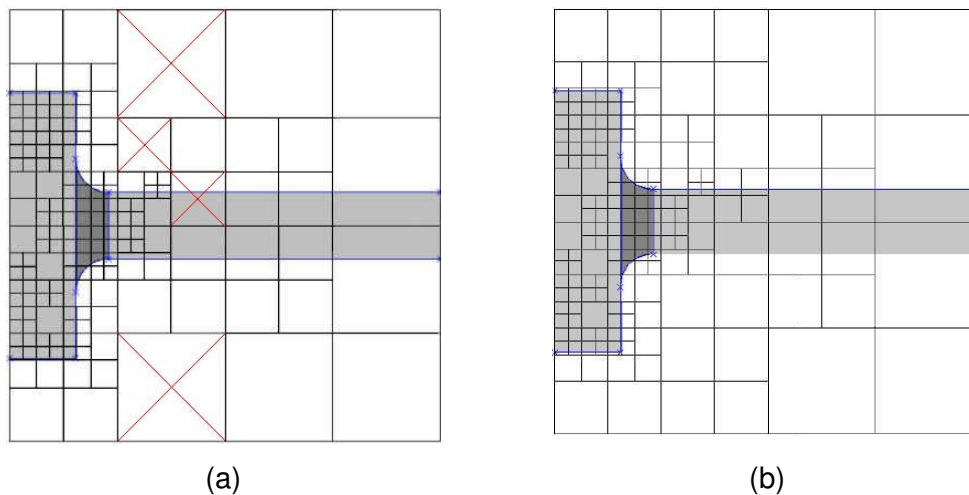


Figura 18 (a) *quadtree* inicial sem refinamento que força a diferença mínima entre as células da octree / *quadtree*, (b) *quadtree* final com refinamento adicional que força a diferença mínima entre as células da octree / *quadtree*.

A Figura 18, também mostra a capacidade que o algoritmo tem de tratar modelos com multiregiões, pois as regiões em cor mais escura, são regiões com propriedades diferentes das de cor mais clara. Isto é muito importante para tratar modelos que possuam características diferentes em regiões diferentes, tal como: relações constitutivas, propriedades térmicas, espessura, etc, variando ao longo do modelo.

3.6. Refinamento das curvas

Após a construção da *octree* / *quadtree* global, o passo seguinte é refinar as curvas do contorno do modelo. Isto é feito através da técnica de Decomposição Espacial Recursiva binária (*binary-tree*) da curva. Desta vez, as novas subdivisões das curvas serão as subdivisões finais do passo corrente do processo adaptativo, pois agora a *octree* / *quadtree* global já contém todos os parâmetros necessários à execução do processo adaptativo.

A decomposição recursiva binária é descrita a seguir. Caso o tamanho da curva seja maior que o tamanho da célula em que o centro da curva está, a curva será dividida pela metade. O processo é repetido para as duas metades restantes e assim sucessivamente até que o critério acima não seja atendido. A Figura 19 ilustra esta técnica para uma curva reta em 2D. A *quadtree* no trecho é mostrada tracejada. No início (Refinamento 0), a curva tem um único segmento de subdivisão, que abrange todo o seu comprimento. No passo seguinte (Refinamento 1), a curva é dividida em dois segmentos iguais. No terceiro passo

(Refinamento 2), os dois segmentos são divididos em outros dois. No passo seguinte (Refinamento 3), apenas os segmentos da parte da direita são divididos. Finalmente, no último passo (Refinamento 4) somente o último segmento é dividido pela metade.

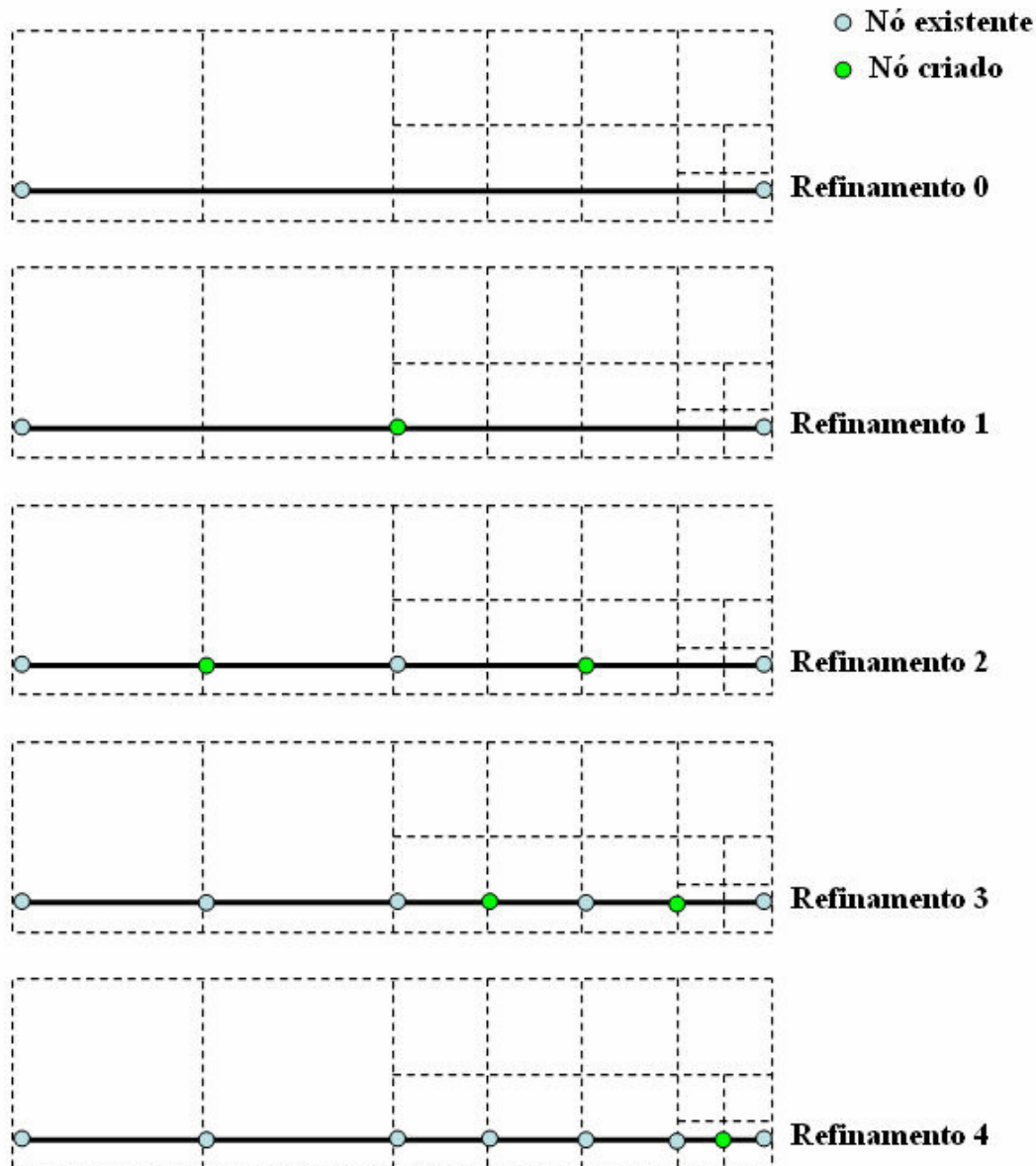


Figura 19 Refinamento de uma curva reta pela técnica de Decomposição Espacial Recursiva binária (*binary-tree*).

Esse uso da *octree* / *quadtree* para refinar a curva e para controlar o tamanho dos elementos é inovador, pois na estratégia anterior de Cavalcante-Neto [12,13] para se refinar as curvas, buscava-se a informação sobre o tamanho característico (h) dos elementos adjacentes à curva, de tal forma que

era necessário, primeiramente, determinar os nós pertencentes à curva para depois se determinar os elementos adjacentes a estes nós.

Esse procedimento não era eficiente, principalmente em 3D, pois exigia uma busca de elemento adjacente para cada segmento da curva. Alguns outros detalhes da estratégia, que não estão relatados aqui, também dificultavam o refinamento binário da curva. A estratégia atual é muito mais eficiente, pois só envolve uma consulta de localização em uma *octree* / *quadtree* global.

Outro ponto muito importante deste algoritmo está na sua capacidade de “desrefinar” regiões em que a análise de erro indique um excessivo grau de refinamento das mesmas. Ou seja, o algoritmo é capaz de indentificar as regiões em que se faz necessário um maior refinamento, bem como as regiões em que o contrário é o que é requerido.

A Figura 20.a mostra uma malha inicial onde todas as curvas do tipo arco de círculo possuem dez subdivisões. Após um passo do processo adaptativo (Figura 20.b) considerando apenas o erro numérico, verifica-se que todas as curvas estavam mais discretizadas que o necessário (segundo os critérios de erro estabelecidos).

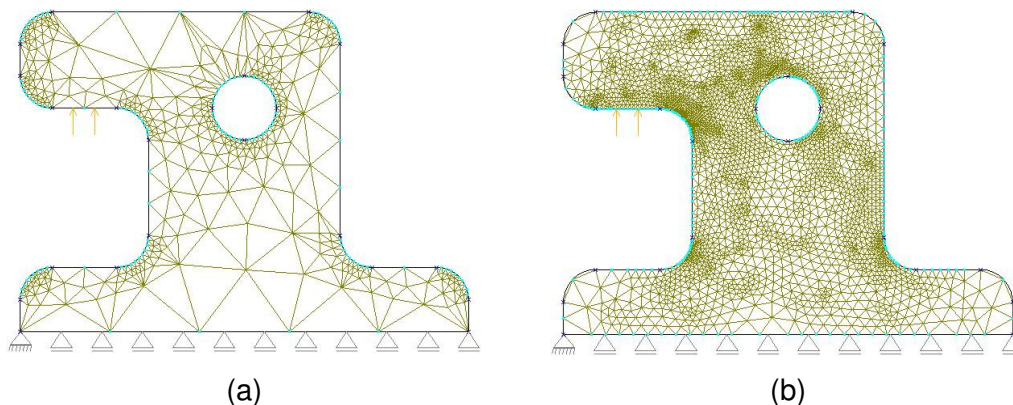


Figura 20 (a) malha inicial com todas as curvas do tipo arco de círculo com dez subdivisões, (b) malha final com suas curvas com o número de subdivisão menor que dez.

Deve-se ressaltar que cada segmento de discretização da curva corresponde a um lado ou aresta de elemento finito. No caso de elementos quadráticos, ainda existirá um nó no meio do segmento que corresponde ao nó de meio de lado ou aresta do elemento.

3.7. Discretização das superfícies

No caso 2D, após o refinamento das curvas, como a *quadtree* global já está gerada e contendo as informações necessárias para definir o tamanho dos elementos gerados na malha de superfície, parte-se para a geração das malhas nas regiões planas. O algoritmo adotado para gerar a malha de elementos triangulares em uma região plana é uma versão simplificada [44] do algoritmo de geração de malhas volumétricas que é descrito na próxima seção. Por isso, o algoritmo 2D não será descrito aqui. A *quadtree* global é utilizada para definir o tamanho dos elementos gerados.

No caso 3D, o algoritmo de geração de malha em uma superfície é semelhante ao algoritmo de geração em uma região plana. Isto porque a geração é feita no espaço paramétrico bidimensional da superfície. A principal diferença é que os tamanhos dos elementos gerados ficam definidos pela *octree* global. Além disso, para compensar as distorções entre as métricas (ângulos e distâncias) dos espaços paramétrico e cartesiano 3D, uma estrutura de dados do tipo *quadtree* auxiliar da superfície é utilizada. O procedimento é idêntico ao que foi descrito na seção 3.4, diferenciando-se apenas no fato de que agora os critérios de adaptatividade geométrica já foram considerados no refinamento da *octree* global. Portanto, o tamanho das células da *quadtree* auxiliar é ditado pela *octree* global.

Uma vez gerada a *quadtree* auxiliar, o algoritmo de geração de malhas de superfície segue o que está descrito no trabalho de Miranda *et al* [44] que, como foi dito, é uma versão simplificada do algoritmo de geração 3D que é descrito na próxima seção.

3.8. Algoritmo de geração de malhas volumétricas

Esta seção descreve o algoritmo de geração de malha volumétrica em uma dada região do modelo que é delimitada por um conjunto de superfícies. O algoritmo foi concebido para atender três requisitos específicos. Primeiramente, o algoritmo deve produzir elementos de boa forma, evitando elementos com aspectos ruins. Apesar do algoritmo não garantir limites nos aspectos de forma dos elementos, são tomados os cuidados necessários a cada passo para gerar os elementos com a melhor forma possível.

O segundo requisito é que o algoritmo deve gerar uma malha que se conforme com a malha triangular no contorno da região. Isto é importante em vários contextos, como por exemplo em propagação de trincas, pois permite que a geração de uma nova malha ocorra localmente em uma região próxima a uma trinca se propagando, ou seja, um número relativamente pequeno de elementos próximos à trinca podem ser removidos criando um vazio na malha; outra aplicação, de interesse direto deste trabalho, é na simulação de problemas com a adaptatividade, onde após uma análise de erro realizada, pode-se remover apenas uma parte localizada da malha, gerando-se igualmente um vazio na mesma e conseqüentemente a necessidade de geração de novos elementos que se conformem com os outros elementos remanescentes da malha anterior e que atendam aos critérios de erro pré-estabelecidos.

O terceiro requisito do algoritmo é que ele tenha a capacidade de criar uma boa transição entre regiões com elementos de tamanhos variáveis. Esse requisito também é importante no contexto de propagação de trincas e na análise adaptativa, pois nestas situações, dependendo do gradiente de tensões, é necessário que se gerem maiores elementos em uma região do que em outra. Porém essa transição deve ser suave, do contrário podem ocorrer problemas numéricos.

O algoritmo trata casos onde as superfícies tridimensionais possuem curvaturas acentuadas. Neste caso, a malha superficial 3D resultante deve ser mais refinada onde houver maior curvatura. Este procedimento evita que elementos vizinhos sejam pontiagudos, o que seria indesejável para o procedimento de análise.

Este algoritmo incorpora aspectos de procedimentos de geração de malhas bastante conhecidos, os quais já foram apresentados no capítulo anterior, incluindo alguns aspectos originais. Ele usa a técnica de avanço de fronteira juntamente com uma árvore octária (*octree*) como orientação local para

o tamanho dos elementos a serem gerados. O método de avanço de fronteira é dividido em duas fases: uma, chamada de fase baseada na geometria, que objetiva a criação de elementos com a máxima qualidade de forma. A outra é denominada de fase baseada na topologia, onde os critérios de formação de elementos ótimos são menos restritivos e o algoritmo tenta criar elementos válidos, usando apenas considerações topológicas. Estas duas adições na técnica de avanço de fronteira são certamente uma das melhores características deste algoritmo. Outro aspecto incluído de grande importância são os chamados procedimentos de “volta-passo”, que são tentativas de fechar a malha de um domínio, quando a técnica de avanço de fronteira tradicional falha. A necessidade deste procedimento surge pelo fato de não ser garantida a discretização de qualquer volume em tetraedros, o que não ocorre quando se usam triângulos na geração de malhas 2D. É importante mencionar que neste algoritmo, os procedimentos de “volta-passo” não são somente usados em um estágio de pós-processamento para melhorar a qualidade da malha, mas também durante a fase de avanço de fronteira, quando um volume menor que não pôde ser subdividido em tetraedros é encontrado. Neste caso, não somente a validação da malha é garantida, como também a qualidade da mesma é melhorada. Neste algoritmo os nós internos são gerados simultaneamente com a geração dos elementos. Este algoritmo também gerencia descontinuidades no domínio ou no contorno do modelo, como nos exemplos que envolvem trincas.

A entrada de dados do algoritmo é a descrição facetada (discretização) do contorno da região onde a malha vai ser gerada, que é obtido por uma lista de nós, definidos por suas coordenadas e por uma lista de faces triangulares, definidas por sua conectividade nodal. Este tipo de entrada de dados pode representar geometrias de qualquer forma, incluindo buracos ou trincas e pode ser facilmente incorporado em qualquer sistema de elementos finitos. Dentro da estratégia adaptativa adotada, a discretização do contorno de cada região, que inclui a discretização das curvas e superfícies da fronteira da região, é feita *a priori*, conforme foi descrito na seção anterior.

A seguir mostra-se os principais passos utilizados pelo algoritmo.

- a) Utilização da *octree* global (gerada conforme citado nas seções anteriores)
- b) Procedimento de avanço de fronteira
 - b.1) Geração de elementos baseada em geometria (elementos ótimos);
 - b.2) Geração de elementos baseada em topologia (elementos válidos);

b.3) Geração de elementos baseada em procedimento “volta-passo” com remoção de elementos que estejam impedindo a técnica de avanço de fronteira fechar a geração da malha dentro de um domínio.

c) Melhoria local da malha

c.1) Suavização Laplaciana com testes de validação;

c.2) Avaliação da qualidade e procedimento “volta-passo” local com remoção de elementos.

3.9. Localização do tamanho característico h para os novos elementos

Esta seção faz uma discussão sobre a maneira de se refinar a *octree* / *quadtree* global em função do erro numérico (seção 3.2). A estratégia adotada consiste em associar o tamanho característico h de cada elemento ao seu centro geométrico, como sugerida por Cavalcante-Neto [12,13]. Outra possibilidade encontrada nos outros trabalhos pesquisados foi a de associar o valor de h aos pontos de Gauss de cada elemento, o que torna o processo um pouco mais lento. A estratégia de associar o valor de h de cada elemento ao seu centro geométrico resulta em malhas com uma transição mais suave entre as regiões com diferentes gradientes de tensões, mas apresenta um problema de dificultar o refinamento de cantos (Figura 21).

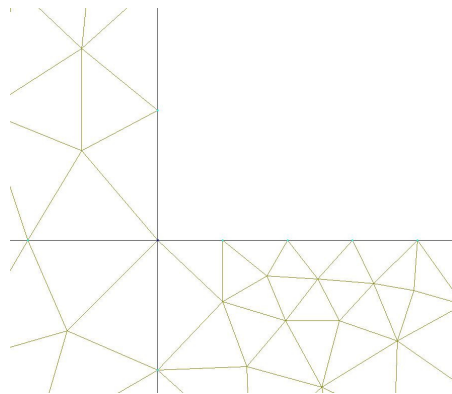


Figura 21 Problema de refinamento em cantos quando o valor de h de cada elemento é associado ao seu centro geométrico.

A Figura 21 mostra que as subdivisões das curvas não vão aumentando do canto onde apresenta maior concentração de tensões. Isto só acontece com aumento dos passos do processo adaptativo.

Como tentativa de evitar este problema tentou-se relacionar o tamanho característico (h) de cada elemento aos seus vértices. Entretanto neste caso, elementos de fronteira de uma região que apresente uma alta concentração de tensão acabam passando para outra região o erro correspondente à região anterior, ocasionando com isto, refinamento excessivo em regiões que não requeiram tal nível de refinamento.

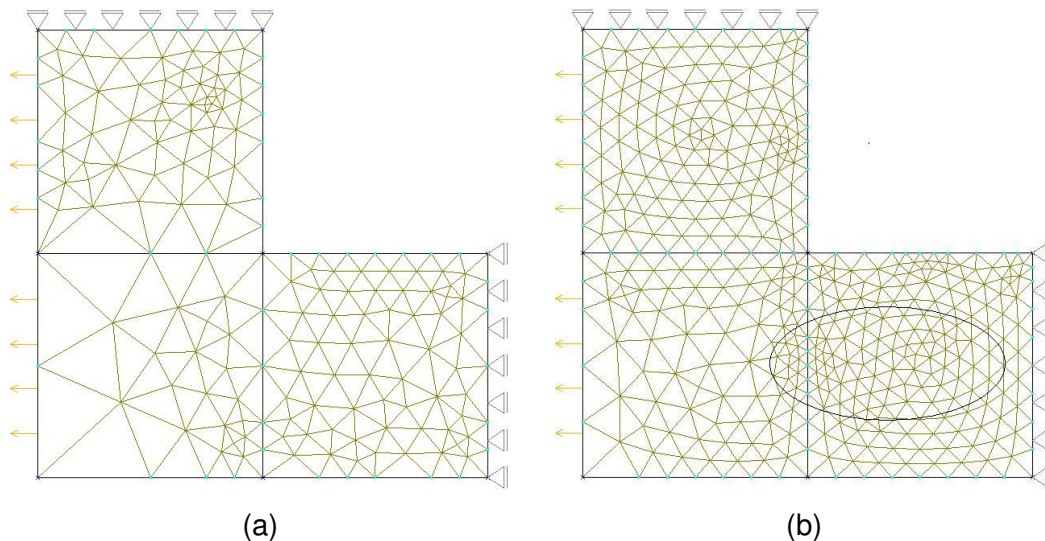


Figura 22 Compara as malhas, referentes ao mesmo passo do processo adaptativo, com o valor de h de cada elemento associado ao seu centro geométrico (a) e com o valor de h associado aos vértices de cada elemento (b).

Pela Figura 22 percebe-se que quando o valor de h é associado ao centro geométrico do elemento gera-se um número menor de elementos e regiões que não apresentam concentrações de tensões elevadas não são refinadas desnecessariamente. Diferentemente, quando o valor de h é associado aos vértices dos elementos, o problema de refinamento dos cantos é amenizado, pois o próprio vértice de canto terá tantos valores de h associados a ele quanto for o número de elementos adjacentes a este vértice, gerando desta forma um novo problema, o refinamento excessivo e desnecessário em determinadas regiões, que não apresentem elevadas concentrações de tensões, o que está destacado na Figura 22.b.

Portanto, em ambas possibilidades de associação do valor de h aos elementos foram encontrados problemas. Por isso optou-se pela estratégia adotada por Cavalcante-Neto.

3.10. Considerações finais

Esta seção visa justificar o duplo refinamento das curvas e da geração das *quadrees* auxiliares, para o caso 3D, proposto para o processo adaptativo (vide-Figura 11).

A idéia central do processo adaptativo em 3D é a criação de uma estrutura *octree* global para armazenar a distribuição espacial dos tamanhos característicos dos elementos finitos que serão gerados em cada passo do processo. Esta estrutura é construída em três passos, levando em conta o erro numérico (primeira etapa), curvatura das curvas (segunda etapa) e curvatura das superfícies (terceira etapa). A *octree* final de cada passo do processo adaptativo é utilizada para discretizar curvas, superfícies e regiões (geração da malha).

Para se levar em conta as curvaturas das superfícies constrói-se uma primeira versão de uma estrutura *quadtree* auxiliar para cada superfície, seguindo o procedimento proposto por Miranda *et al* [45]. Ocorre que para se criar a *quadtree* auxiliar é preciso partir de um refinamento das curvas de bordo da superfície. Por isso, é necessário, fazer um refinamento prévio das curvas, que leva em consideração o erro numérico (informação contida na *octree* global inicial) e as curvaturas das curvas. Esse refinamento prévio de curvas não é utilizado para a discretização final das curvas, pois neste estágio ainda não se levou em conta as curvaturas de todas as superfícies na construção final da *octree* global.

Por outro lado, a primeira versão da *quadtree* auxiliar de cada superfície não levou em conta o erro numérico no interior da superfície e a influência das outras superfícies. Por isso esta versão é descartada sem que se gere uma malha na superfície. Ela é apenas utilizada para refinar a *octree* global considerando a curvatura de cada superfície. Somente após a construção da *octree* global e de se obter a discretização final das curvas é que se cria a *quadtree* auxiliar final de cada superfície. Esta versão final é utilizada para discretizar a superfície.

4 IMPLEMENTAÇÕES DO PROCESSO ADAPTATIVO

Este capítulo apresenta as implementações do novo processo adaptativo tanto para os problemas 2D quanto para os 3D. Entende-se por processo adaptativo todo o procedimento necessário para a geração de malhas adaptativas. Portanto estão incluídos neste contexto a estratégia adaptativa (h), os métodos de suavização ($ZZ-HC$ [62] e SPR [63-66]), os estimadores de erro e as técnicas de geração de malhas.

Antes é importante esclarecer alguns detalhes. Por exemplo, o processo de simulação de um problema físico é dividido em três partes: representação geométrica e topológica do modelo (módulo de modelagem), solução do problema (módulo de análise) e interpretação dos resultados (módulo de visualização).

Tanto para o problema 2D quanto para o 3D, usou-se somente um programa de análise, denominado *FEMOOP* [43], mas os outros dois módulos são diferentes, dependendo da dimensão do problema. Para o caso 2D, o programa responsável pelo módulo de modelagem e pelo módulo de visualização é o *MTOOL* [47]. Para o caso 3D, o *MG* [48] é o programa responsável pelo módulo de modelagem e o *POS-3D* [16] é responsável pelo módulo de visualização. Em 2D, os módulos de modelagem e visualização integrados em um único programa (*MTOOL* [47]), que se comunica automaticamente com o módulo de análise (*FEMOOP* [43]). Esta integração ainda não foi feita para o caso 3D. Entretanto, para se fazer o processo adaptativo não é necessário o módulo de visualização. Portanto, o processo adaptativo foi implementado no *MTOOL* [47] (2D) e no *MG* [48] (3D). Outra diferença entre os dois casos é que o *MTOOL* foi desenvolvido usando-se a linguagem de programação C e o *MG* [48] foi desenvolvido em C++, fundamentada no paradigma de programação orientada a objetos.

Outro aspecto muito importante na simulação de problemas físicos é o gerenciamento dos atributos. Nos dois casos, o módulo responsável por esta parte é o ESAM (*Extensible System Attributes Management*)^{*}. O gerenciamento

^{*} para maiores detalhes ver [36].

de atributos é importante para que os modeladores sejam capazes de atender aos diversos tipos de análise em engenharia, como, por exemplo, estrutural, naval, termodinâmica, etc. Os atributos para cada tipo de simulação devem poder ser configurados sem a necessidade de se alterar o código-fonte do modelador.

O *ESAM* também possibilita que uma interface seja definida para a captura dos dados referentes aos atributos do *MTOOL* [47] e do *MG* [48]. Existe, ainda, um mecanismo que permite ao usuário do *MTOOL* [47] ou do *MG* [48] implementar a rotina responsável pela geração do arquivo de dados, referente a comunicação entre os pré-processadores e os módulos de análise. Assim, os novos atributos incorporados ao programa de análise podem ser facilmente inseridos aos pré-processadores, a partir da definição de uma interface para capturar os seus dados, bem como a inclusão, pelo próprio usuário do sistema, dos dados referentes a esses atributos na rotina que gera o arquivo de dados lido pelo programa de análise. No *ESAM*, a configuração dos atributos é feita através de arquivos que contém instruções a linguagem de programação extensível LUA [29] (linguagens extensíveis são usadas para estender ou configurar uma aplicação para fins particulares). Esta linguagem é interpretada, permitindo que a configuração dos atributos seja feita sem a necessidade de recompilação do sistema. Este sistema utiliza uma abordagem conhecida como modelagem baseada em geometria. Neste tipo de abordagem, os atributos são associados às entidades geométricas do modelo e a discretização desta geometria (malha de elementos finitos) herda automaticamente estes atributos. Este tipo de modelagem oferece diversas vantagens, como, por exemplo, um melhor suporte para a automação do processo de modelagem, incluindo geração automática de malhas e conseqüentemente facilitando a utilização do processo adaptativo, pois os atributos são associados às entidades geométricas do modelo, não precisando serem redefinidos durante o refinamento da malha.

4.1. *MTOOL* (2D)

Nesta seção, apresenta-se de forma sucinta o programa *MTOOL* [47], bem como todos os outros componentes usados no seu desenvolvimento.

O *MTOOL* [47] é um programa utilizado para simulações numéricas de problemas bidimensionais pelo Método dos Elementos Finitos. Este sistema integra, em um único ambiente, os módulos de pré-processamento (sistema gráfico interativo e configurável para geração, edição e manipulação de malhas de elementos finitos bidimensionais) e pós-processamento, responsáveis pela geração do modelo numérico para análise pelo MEF e visualização de resultados, com o programa de análise numérica *FEMOOP* [43] (análises numéricas quasi-estáticas, com não-linearidade física, de modelos de estado plano de tensão, estado plano de deformação e aximétricos). Este sistema é extensível e configurável, permitindo que novas funcionalidades sejam facilmente implementadas.

Este sistema foi desenvolvido utilizando-se o sistema de interface com o usuário IUP/LUA* que permite a obtenção de uma interface atrativa, configurável e eficiente para execução das tarefas. Sob o sistema de interface e o sistema para geração de malhas propriamente dito, encontra-se o sistema gráfico CD [19], que garante a portabilidade multi-plataforma da visualização. Outra ferramenta importante utilizada no desenvolvimento do *MTOOL* [47] é a biblioteca, HED [42], que manipula estruturas de dados topológicas para o gerenciamento de subdivisões planares. Além disso, foi utilizado um sistema configurável para o gerenciamento de atributos, *ESAM*, escrito na linguagem interpretada LUA [29].

O *MTOOL* [47] é composto por cinco módulos responsáveis pelas diversas tarefas envolvidas em um processo de simulação (modelagem geométrica, especificação dos atributos, geração da malha de elementos finitos, análise numérica e visualização dos resultados), e por um módulo principal responsável pelo gerenciamento e integração dos outros módulos (Figura 23).

* <http://www.tecgraf.puc-rio.br/iup>

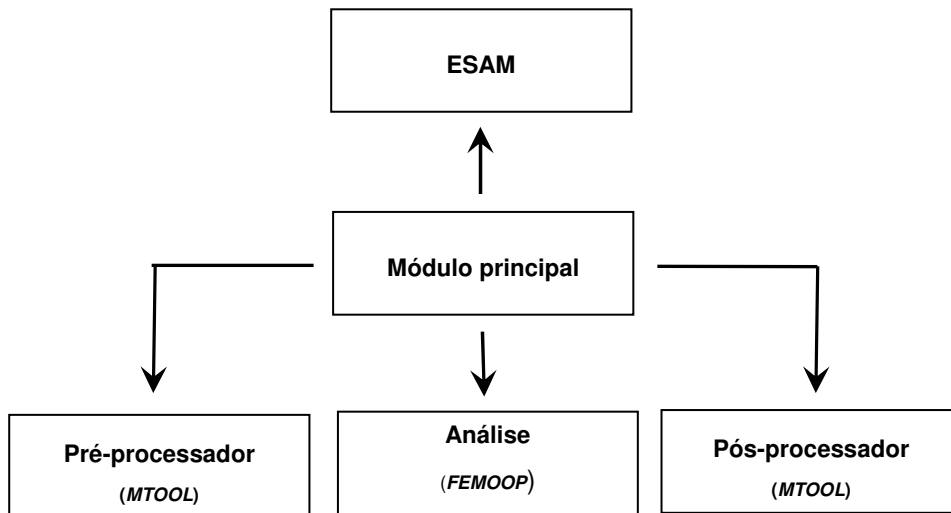


Figura 23 Fluxograma mostrando o funcionamento do *MTOOL*.

O *MTOOL* [47] (*Two-dimensional Mesh Tool*) possui dois ambientes bem distintos. O primeiro é responsável pela definição da geometria e topologia do modelo. Este ambiente é suportado pela estrutura de dados topológica *HED* (*half-edge data structure*) [42], que permite assegurar a consistência do modelo durante a sua criação. É possível, por exemplo, identificar interseções entre as curvas, determinar se uma região (face) foi definida ou dividida, verificar os efeitos indiretos de uma determinada atribuição (por exemplo, a troca do número de subdivisões de uma curva elimina as malhas adjacentes, caso contrário o modelo ficaria inconsistente), etc. A implementação desta estrutura de dados topológica é complexa e ocupa um espaço de memória relativamente grande para a sua realização. Entretanto, a biblioteca de rotinas *HED* [42] oferece uma interface de alto nível (nível abstrato), abstraindo a complexidade desta estrutura de dados. O segundo ambiente do *MTOOL* [47] apresenta uma representação única de malhas de elementos finitos convencional, obtida a partir da conversão das malhas do modelo de regiões, definido no ambiente de edição do modelo. No segundo ambiente, o usuário pode realizar operações típicas de modelos de elementos finitos, como, por exemplo, consultar a incidência nodal de cada elemento finito gerado na conversão. No entanto, não existem funções para edição da malha, pois não se tem o suporte de uma estrutura de dados topológica. Para isso, o usuário pode retornar ao ambiente de edição do modelo, realizar as alterações necessárias e fazer uma nova conversão para o modelo de elementos finitos.

O módulo de visualização fornece informações qualitativas e quantitativas sobre a malha e os resultados referentes à mesma. A análise quantitativa é feita através de gráficos de resposta e funções de consulta, que fornecem informações tais como número de nós ou incidência do elemento. A análise qualitativa consiste basicamente na representação visual de resultados, que podem ser de dois tipos: resultados escalares, como componentes de tensão ou campo de temperatura, e resultados vetoriais, tais como campo de deslocamentos ou campo de velocidades. Os resultados de campos escalares podem ser fornecidos nos pontos nodais, de forma suavizada ou não, ou nos pontos de Gauss. Para auxiliar na visualização do modelo, existem funções para manipular o campo de visão, tais como a possibilidade de mudar os limites da janela de visualização ou fazer um *zoom* em todo o modelo ou em alguma face do mesmo. O programa ainda permite visualizar a animação do modelo com a resposta de um determinado caso e campo ao longo de diversos passos. A entrada de dados referente à malha de elementos finitos e aos resultados obtidos a partir da análise numérica é feita através de arquivos neutros.

O módulo principal é responsável pelo gerenciamento e integração dos demais módulos do *MTOOL* [47] (*MTOOL*, *ESAM*, *FEMOOP*).

Neste módulo são chamadas as funções de inicialização referentes aos outros módulos do sistema. Outra tarefa realizada por este módulo consiste na conversão dos arquivos gerados pelo programa *FEMOOP* [43] contendo os resultados da análise, que possuem formatos próprios, para que possam ser interpretados pelo pós-processador *MTOOL* [47], que utiliza o formato do arquivo de dados *NeutralFile*^{*}.

^{*} ver (<http://www.tecgraf.puc-rio.br/neutralfile>)

4.1.1. Adaptatividade no *MTOOL*

Para que o uso da adaptatividade pudesse ser disponibilizado pelo programa *MTOOL* [47], foram implementados módulos responsáveis pelo refinamento da *binary-tree* e a conseqüente subdivisão das curvas e módulos responsáveis pelo refinamento da *quadtree* e o consecutivo controle do tamanho dos elementos que serão gerados nas malhas de superfície.

Como o *MTOOL* [47] é dividido em dois ambientes (o ambiente de modelagem e o ambiente de malha) foi necessário separar o processo adaptativo nestes dois ambientes. Desta forma, após se selecionar o método de suavização e determinar a razão de erro máxima η^* , passa-se para o ambiente de malha e se faz a análise por elementos finitos. Em seguida executa-se a opção da análise adaptativa com ou sem a adaptatividade geométrica.

O processo adaptativo no *MTOOL* [47] está composto basicamente por duas funções, que são: *_MtoolAdaptiveQt* e *_MtoolAdaptiveMesh*.

A função *_MtoolAdaptiveQt* tem fundamentalmente as seguintes tarefas.

- 1) Ler as informações do erro de cada elemento e associa o valor de h ao centro geométrico de cada elemento (*MtoolGetError*);
- 2) Calcular em qual das coordenadas Cartesianas está a maior dimensão do sólido, guardando-as (*MtoolGetBoundingBox*);
- 3) Construir a *quadtree* global (*Adapt_CreateQuadtree*).

As tarefas da função *_MtoolAdaptiveQt* podem ser visualizadas na Figura 24.

```
// variáveis da função
vars = xmax, ymax, xmin, ymin, n_elem, n_nodes, eta, inf_el_eachnode,
      par0, par1, par2, par3

// captura o parâmetro que verifica se será feita a adaptatividade geométrica
par0 = 0 ou 1

// testa se é para ser feita a adaptatividade geométrica
Se par0 = 1 capturam-se os outros parâmetros necessários para
a adaptatividade geométrica.

// tolerância mínima da distância entre as subdivisões estipulada pelo usuário
// do programa
```



```

    par1 = dMIN;
// tolerância máxima da distância entre as subdivisões estipulada pelo usuário
// do programa
    par2 = dMAX;
// tolerância angular estipulada pelo usuário do programa
    par3 = angTOL
fim Se

Se par0 = 0 não se fará a adaptatividade geométrica e inicializam-se
os outros parâmetros como zero.
    par1 = 0; // tolerância mínima da distância entre os pontos
    par2 = 0; // tolerância máxima da distância entre os pontos
    par3 = 0; // tolerância angular
fim Se

// captura o parâmetro que verifica se será feita a adaptatividade baseada em
// erro
par4 = 0 ou 1

// preenche o vetor inf_el_eachnode e retorna o valor do erro corrente
eta = MtoolGetError (n_elem, n_nodes, inf_el_eachnode);

// calcula as coordenadas da Bound Box (coordenadas cartesianas máximas e
mínimas)
MtoolGetBoundingBox (n_elem, xmax, xmin, ymax, ymin);

// Cria a quadtree no ambiente de malha
Adapt_CreateQuadtree (xmax, ymax, xmin, ymin, n_elem, inf_el_eachnode,
                    n_nodes, par0, par1, par2, par3, par4);

// retorna o valor do erro corrente
(eta)

```

Figura 24 Pseudo código da função *_MtoolAdaptiveQt*.

Na função *MtoolGetError* mostrada da Figura 24, calcula-se o valor de h discutido na seção 3.2, que é o valor que controlará o refinamento da *quadtree*, da *binary-tree* e o tamanho dos elementos triangulares que serão gerados no interior da malha.

A Figura 25 mostra o trecho de código com o cálculo de h quando se usa elementos lineares T3. No caso quadrático (T6), o trecho de código para o cálculo de h é praticamente o mesmo. A única diferença está no instante de calcular a área existente e no valor de p usado para obter o valor de h , pois conforme a equação (23), o valor de p varia de acordo com o grau do polinômio de aproximação, ou seja $p = 1$ para o T3 e $p = 2$ para o T6.

```
// variáveis da função
vars = norm_beb, norm_bub, eta_barra, beb, bub, i, k, a, hold, hnew, fact, erro
// calcula a área do elemento triangular T3 (a) através dos valores
// nodais dos vértices do elemento P1(v1x,v1y), P2(v2x,v2y) e P3(v3x,v3y)
a = 0.5*det[(v1x*v2y)+(v2x*v3y)+(v3x*v1y)-(v2y*v3x)-(v1y*v2x)-(v1x*v3y)]

// valor de L após se igualar às áreas existente e ideal
hold = sqrt( 4 * a / sqrt(3) );

// lê do arq '.pos' o valor de RATIO_ERRO para cada elemento
De i = 0 até i < nelem faz-se i = i + 1
    // lê o RATIO_ERRO do elemento corrente ( $\varepsilon_i$ )
    error = ResEIScalarField.nod[i]
    // lê o NORM_ERRO do elemento corrente ( $\|\bar{e}\|$ )
    beb = ResEIScalarField.nod[i]
    // lê o NORM_TENSOR do elemento corrente ( $\|\hat{u}\|$ )
    bub = ResEIScalarField.nod[i]
    // atualiza os valores das variáveis
    norm_bub = norm_bub + bub*bub;
    norm_beb = norm_beb + beb*beb;
fim De.
// preenche o valor do erro relativo corrente
eta_barra = sqrt(norm_beb)/sqrt((norm_bub)+(norm_beb))
eta = (eta_barra*100)
```

```

// calcula o novo valor de 'h'
fact = pow( error, 1.0/1 );
hnew = hold / fact;
// preenche o novo valor de 'h'
s = hnew

```

Figura 25 Pseudo-código da função que preenche o valor de h para elementos do tipo T3.

Na função *Adapt_CreateQuadtree* mostrada na Figura 24, constrói-se a *quadtree* global, considerando ou não a adaptatividade geométrica, o que é indicado pelo valor da variável *par0* e considerando ou não a adaptatividade baseada no erro numérico, o que é indicado pelo valor da variável *par4*.

```

// variáveis da função
vars = n_pts // indica o número de pontos adicionados a curva devido os
              // critérios da adaptatividade geométrica
inf_pts // vetor que guardará as coordenadas Cartesianas do centro
         // geométrico de cada trecho e o tamanho do mesmo
i // contador
// parâmetros da função
nelem // número de elementos na malha
inf_msh // vetor que guardará as coordenadas Cartesianas do centro
         // geométrico de cada elemento da malha e o valor de h
Se par0 = 1 constrói-se a quadtree global levando em consideração a
adaptatividade geométrica
Para cada curva do sólido
  SdvAdaptRefineStretchGeom(par1, par2, par3, n_pts, inf_pts);
  // passa as informações de todos os trechos de cada curva para a construção
  // da quadtree
  De i = 0 até i < n_pts faz-se i = i + 1
    AddVertexSize (inf_pts)
  Fim De;
fim Para;
fim Se;

```

```

Se par4 = 1 constrói-se a quadtree global levando em consideração a
adaptatividade baseada no erro numérico
    // passa as informações de todos os elementos da malha para
    // a construção da quadtree
    De i = 0 até i < nelem faz-se i = i + 1
        AddVertexSize (inf_msh);
    Fim De;
fim Se;

```

Figura 26 Pseudo-código da função que constrói a *quadtree* considerando ou não a adaptatividade baseada no erro numérico e na adaptatividade geométrica.

Já a função *_MtoolAdaptiveMesh* tem como tarefa:

1) Chamar a função *Adapt_CreateMesh* que efetivamente gera a nova malha conforme os parâmetros de erro numérico e geométricos.

As tarefas da função *_MtoolAdaptiveMesh* podem ser visualizadas na Figura 27.

```

// Gera a nova malha no ambiente geométrico
Adapt_CreateMesh ( );

// Desaloca as memórias que foram alocadas durante a execução da função
Adapt_End ( );

```

Figura 27 Pseudo código da função *_MtoolAdaptiveMesh*.

Na Figura 27, uma função que merece destaque é a *Adapt_CreateMesh*, pois é nesta função que se subdivide as curvas do modelo e gera-se a nova malha de superfície conforme a *quadtree* global.

4.2. MG (3D)

O *MG* [48] é um pré-processador que foi originalmente desenvolvido para geração de superfícies (cascas) em modelos de elementos finitos, sendo, na seqüência, estendido para considerar também malhas sólidas. O *MG* [48] incorpora duas capacidades importantes em modelagem 3D pelo MEF. A primeira está relacionada com a interface com o usuário e procedimentos gráficos interativos para gerar malhas em superfícies e a segunda relaciona-se com a interseção de superfícies paramétricas

O algoritmo para interseção de superfícies usado no *MG* [48] é baseado em um esquema para interseção de superfícies paramétricas onde as malhas de superfícies existentes são usadas como suporte para a definição inicial das curvas de interseção. Uma estrutura de dados auxiliar, definida no espaço paramétrico de cada superfície, permite a construção das curvas de *trimming* (curvas resultantes da interseção entre superfícies e que possuem representação nos espaços paramétricos destas e no espaço Cartesiano) sem a necessidade de se realizar buscas globais.

O *MG* [48] foi inicialmente útil para a representação de modelos geométricos, porém sua estrutura de dados original não foi baseada em qualquer conceito formal de modelagem geométrica. Em várias ocasiões, a consistência geométrica de um modelo era realizada com a intervenção do usuário. Por exemplo, não existia a capacidade de detectar automaticamente quando um volume no espaço é definido por um conjunto de retalhos de superfícies (ou seja, superfícies limitadas pelas suas curvas de bordo). O usuário devia indicar a construção explicitamente, o que pode não ser uma tarefa trivial na modelagem de um problema real de engenharia. Essa capacidade é particularmente importante na geração de malhas formadas por elementos finitos sólidos, onde algumas vezes é desejável ter várias regiões com mapeamentos distintos, cada uma definida com suas respectivas malhas de contorno geradas por diferentes tipos de algoritmos.

Lira em 2002 [37] propôs uma técnica que permite o reconhecimento automático de regiões sólidas, adotando uma metodologia que é baseada em uma representação topológica completa de uma subdivisão espacial, chamada *Complete Geometric Complex (CGC)* [14].

A representação *CGC* é implementada como uma biblioteca de classes, no contexto da programação orientada a objetos (POO), provendo um conjunto de operadores de alto nível que manipulam uma subdivisão espacial. Estes

operadores recebem como parâmetros de entrada as informações geométricas dos retalhos de superfícies que são, então, inseridas na subdivisão espacial. Estas informações geométricas são automaticamente transformadas em informações topológicas requeridas pelos operadores de baixo nível. A implementação original da *CGC* [14] tratava apenas de retalhos de superfícies planas. No trabalho de Lira isso foi estendido para superfícies curvas e paramétricas.

4.2.1. Adaptatividade no *MG*

O processo adaptativo foi inserido no *MG* [48] através de módulos que foram adicionados ao programa com as seguintes tarefas. Os responsáveis por todo o gerenciamento do processo adaptativo (classe *mgAdaptive*), os responsáveis pelo refinamento da *binary-tree* e a conseqüente subdivisão das curvas e os responsáveis pelo refinamento da *octree* e o consecutivo controle do tamanho dos elementos que serão gerados nas malhas de superfície e volumétrica.

Como o *MG* [48] constitui apenas a parte do pré-processamento, pode-se implementar todo o processo adaptativo apenas no ambiente de malha.

Partindo-se do princípio que já se tem o modelo com todos os seus atributos preenchidos, seleciona-se o método de suavização e se determina a máxima razão de erro η^* . Na seção que trata dos arquivos de saída está a opção de executar a análise adaptativa com ou sem a adaptatividade geométrica.

O processo de comunicação entre os programas é feito através de arquivos neutro. O *MG* [48] fornece como dado de saída um arquivo neutro que será lido pelo *FEMOOP* [43] (módulo de análise) que por sua vez fornece um outro arquivo de saída (ver Figura 28) sobre o qual se obtêm as informações de erro que serão utilizadas para o cálculo dos novos tamanhos dos elementos.

```
// variáveis da função
vars = xmax, ymax, xmin, ymin, n_elem, n_nodes, eta, inf_el_eachnode,
      par0, par1, par2, par3,
      eta      // retorna o erro corrente
      adpt     // objeto da classe mgAdaptive
      et = 0,  // 0=linear,      1=quadratic
      ek = 1,  // 0=shell,      1=solid
```

```

    ei = 0      // 0=all model, 1=selected
    command // variável com a extensão certa para chamar o Femoop
    f          // referência para arquivo neutro
// captura o parâmetro que verifica se será feita a adaptatividade geométrica
par0 = 0 ou 1

// testa se é para ser feita a adaptatividade geométrica
Se par0 = 1 capturam-se os outros parâmetros necessários para
a adaptatividade geométrica.
// tolerância mínima da distância entre as subdivisões estipulada pelo usuário
// do programa
    par1 = dMIN;
// tolerância máxima da distância entre as subdivisões estipulada pelo usuário do
// programa
    par2 = dMAX;
// tolerância angular estipulada pelo usuário do programa
    par3 = angTOL
fim Se;

Se par0 = 0 não se fará a adaptatividade geométrica e inicializam-se
os outros parâmetros como zero.
    par1 = 0; // tolerância mínima da distância entre os pontos
    par2 = 0; // tolerância máxima da distância entre os pontos
    par3 = 0; // tolerância angular
fim Se;

// captura o parâmetro que verifica se será feita a adaptatividade baseada em
// erro
par4 = 0 ou 1

// manipula o nome do arquivo para mandá-lo de forma adequada
// (extensão correta) ao programa de análise constrói a malha
building ( et, ek, ei )

// salva o arquivo usando o Esam.
createMesh( mesh )
writeMesh ( f )

```

```

// chama o programa de análise (femoop)
roda(command);

// executa o processo adaptativo
mgRegenMeshes(mesh, f, eta, par0, par1, par2, par3, par4))

// destroi a malha.
kill();
// fecha o arquivo '.pos'
fclose( f );
// chama a função em Lua que mostra o resultado de erro da análise
ResultError (eta);

```

Figura 28 Pseudo código no *MG* que gerencia a gravação e leitura dos arquivos neutros.

O método *mgRegenMeshes* da classe *mgAdaptive* é que realmente gerencia todo o processo adaptativo. Este método é mostrado detalhadamente na Figura 29.

```

// variáveis da função
vars = new_size, n_elements, xmax, ymax, zmax, xmin, ymin, zmin, oct,
      new_tpts, pra_val

// obtém o número de elementos da malha
n_elements = getNumberOfElems( );

// cria a Bound Box
mgGetBoundingBox(n_elements, xmax, xmin, ymax, ymin, zmax, zmin);

// Inicia a octree passando as coordenadas Cartesianas máximas
OctreeBegin (xmax, ymax, zmax, xmin, ymin, zmin);

// preenche os valores new_size de todos os elementos da malha e o valor do
// erro corrente eta
mgGetErroSizeEachElement (mesh, new_size, eta));

```



```

// refina a octree global considerando o erro numérico e com os critérios de erro
// geométrico das curvas, caso esta opção tenha sido selecionada, porém as
// mesmas não são subdivididas neste estágio.
mgCreateOctree (oct, xmax, ymax, zmax, xmin, ymin, zmin, mesh,
                new_size, par0, par1, par2, par3);

// Caso se opte pela adaptatividade geométrica as curvas subdividas
Se par0 = 1
  Para cada curva do sólido
    // subdividem-se as curvas com os critérios de erro e de adaptatividade
    // geométrica passando-se o número de pontos (new_pts) e as suas
    // coordenadas paramétricas (par_val).
    segmentSetSubd(new_pts, par_val);
  fim Para;
fim Se;

// refina a octree global com os valores passados pela quadtree auxiliar
// considerando a curvatura das mesmas
mgRefineOctSurface(oct);

// método responsável pela subdivisão das curvas e pela geração das novas
// malhas de superfícies e volumétricas
mgAdapt_CreateMesh ( mesh );

```

Figura 29 Pseudo código no MG que mostra o método *mgRegenMeshes*.

Todos os métodos presentes na Figura 29 são da classe *mgAdaptive*. O método *mgGetBoundingBox* da Figura 29 é responsável por obter as maiores coordenadas cartesianas do modelo, informação que será usada para a construção da *octree*.

O método *mgGetErroSizeEachElement* é responsável por ler as informações necessárias do arquivo neutro para a estimativa de erro e por calcular os novos valores de h para cada elemento da malha, conforme a equação 34, guardando estas informações na estrutura de dados *new_size* (ver Figura 29).

O método *mgCreateOctree* recebe os valores extremos do modelo preenchidos no método *mgGetBoundingBox*, cria a *octree* global e a refina de acordo com os dados contidos na variável *new_size* e com os parâmetros da adaptatividade geométrica das curvas.

O método *mgRefineOctSurface* recebe, como parâmetros de entrada, as curvas de todo modelo subdivididas conforme os critérios de erro numérico e os parâmetros da adaptatividade geométrica das curvas. Este método tem a função de refinar a *octree* global conforme os critérios de curvatura em cada superfície, conforme o algoritmo de Miranda *et al* (vide – seção 3.4).

O último método que aparece na Figura 29, dentro do método *mgRegenMeshes*, é o *mgAdapt_CreateMesh*. Ele é responsável pelo refinamento das curvas, pela geração das novas malhas de superfície e pela geração da malha volumétrica. Este método pode ser visto na Figura 30.

A Figura 30 mostra em forma de pseudo código as principais funcionalidades do método *mgAdapt_CreateMesh* da classe *mgAdaptive*.

```
// variáveis do método
seg    // é um ponteiro para um objeto da classe mgSegment
octree // é um objeto da classe mgOctree
p2     // é um ponteiro para um objeto da classe mgPatch2d
tetra  // é um ponteiro para um objeto da classe mgTetra
n_pts  // número de pontos da subdivisão da curva
c_par  // coordenadas paramétricas dos pontos das subdivisões de cada curva

// para cada curva do modelo gera-se a nova subdivisão usando as informações
// de erro contidas na octree, preenchendo-se o número de pontos com suas
// respectivas coordenadas paramétricas através do seguinte método
// da classe mgAdaptive
mgPointsOnEdgeOCT(seg, 1, octree, n_pts, c_par)

// de posse destas informações geram-se os novos pontos da subdivisão através
// do método segmentMakeUse da classe mgSegment
segmentSetSubd(new_pts, par_val)

// com as novas subdivisões das curvas geram-se as novas malhas de superfície
// através do método patch2dReplaceMesh da classe mgPatch2d
```

```
patch2dReplaceMesh( mgGenMesh2d::CONTRACTION )  
  
// com as novas subdivisões das curvas e com as novas malhas de superfície  
// geradas geram-se as novas malhas volumétricas através do método  
// RebuildTetra da classe mgTetra  
RebuildTetra ( )
```

Figura 30 Pseudo código no *MG* que mostra o método *mgAdapt_CreateMesh*, que é o método que gerencia todo o processo adaptativo.

5 EXEMPLOS

Neste capítulo são apresentados vários exemplos com soluções conhecidas objetivando validar a eficiência do novo processo adaptativo. Os resultados obtidos nestes exemplos são comparados com os resultados obtidos por Cavalcante-Neto [12,13], já que o trabalho desenvolvido pelo referido autor foi usado como base para a elaboração deste trabalho. Outros exemplos analisados neste trabalho também foram feitos por outros autores: Zienkiewicz [66] e A Merrouche [1].

É importante enfatizar que as comparações feitas neste capítulo têm apenas o intuito de mostrar o aspecto das malhas com os resultados do processo adaptativo conforme a evolução do processo, mostrando também o comportamento do critério de convergência para todos os casos (equação 17). Portanto, não cabe avaliar, através dos resultados, qual método é mais eficiente ou rápido, pois em alguns exemplos não se sabe qual foi à técnica de geração de malha, nem o método de suavização e nem o critério de erro adotado pelos outros autores, não conseguindo, desta forma, compatibilizar as características iniciais do problema: número de nós (NN), número de elementos (NE) e o número de graus de liberdade (GDL). Além disto, qualquer alteração na discretização das curvas do modelo resultarão em malhas diferentes, mesmo que as condições iniciais tenham sido compatíveis.

O capítulo está dividido em duas seções: exemplos em 2D, rodados no *MTOOL* [47] e exemplos em 3D, rodados no *MG* [48]. Em todos os exemplos 2D foram usados os seguintes valores para o erro máximo permitido: $\eta^* = 10\%$ para elementos lineares e $\eta^* = 5\%$ para elementos quadráticos, conforme o que é recomendado na literatura. Para os exemplos 3D, usou-se somente elementos quadráticos e adotou-se dois valores diferentes de η^* . Em alguns casos usou-se $\eta^* = 5\%$ e em outros $\eta^* = 8\%$, conforme os valores encontrados de onde os exemplos foram obtidos.

É importante ressaltar que pelo fato de não ser o objetivo deste trabalho comparar os métodos de suavização existentes, foi utilizado apenas no primeiro

exemplo do caso 2D o método *SPR* [63-66]. Neste mesmo exemplo se faz uma comparação entre os dois métodos, mostrando a pequena diferença entre ambos, justificando, desta forma, apenas o uso do *Z2-HC* [62] nos demais exemplos.

Nas últimas malhas de todos os exemplos apresentou-se, apenas por motivos ilustrativos, as malhas seguintes caso o critério de convergência não tivesse sido atingido no passo anterior do processo adaptativo.

5.1. Exemplos em 2D (*MTOOL*)

Foram usados seis exemplos com o objetivo de validar a implementação do processo adaptativo no *MTOOL* [47]. Em todos os exemplos se fez uma análise para elementos lineares (T3) e outra para elementos quadráticos (T6).

Pode ser observada uma nítida diferença entre os aspectos das malhas geradas para elementos lineares e quadráticos, sendo que para os lineares (T3) a malha apresenta um alto grau de refinamento em regiões que não apresentam grande concentração de tensões. Outros trabalhos na literatura também destacam este fato.

O terceiro exemplo foi oportuno para mostrar a distribuição da razão de erro (ξ_i) de cada elemento das malhas resultantes de cada passo do processo adaptativo. A tabela de cores referente a máxima tensão principal vai do azul escuro (compressão) ao vermelho escuro (tração). Já a tabela de cores referente a razão de erro ξ_i vai do azul (valores negativos) ao vermelho (valores positivos).

5.1.1. Peça mecânica com um furo circular

O primeiro exemplo é a análise de uma peça mecânica com um furo circular. Este exemplo é importante para mostrar a influência da adaptatividade geométrica na convergência feita em conjunto com a análise de erro. Neste exemplo também se faz uma comparação entre os métodos de suavização que foram disponibilizados no programa. Considera-se para a solução deste problema estado plano de deformação com o módulo de elasticidade $E = 10^5$, e coeficiente de Poisson $\nu = 0.25$ e espessura $e = 1$. A Figura 31 abaixo mostra as características do problema.

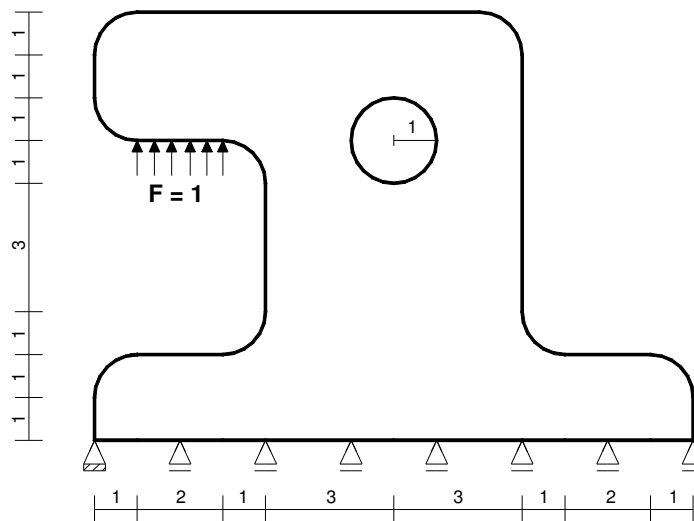


Figura 31 Peça mecânica com um furo circular.

Como os parâmetros de entrada da adaptatividade geométrica são estabelecidos pelo usuário, adotaram-se os seguintes valores para os exemplos tamanho máximo (d_{MAX}) de segmento da discretização da curva igual à metade do comprimento da curva, tamanho mínimo (d_{MIN}) igual ao próprio comprimento da curva e a tolerância angular (ang_{TOL}) igual a dois graus.

Neste exemplo ao se usar elementos do tipo (T3) foram necessários três passos do processo adaptativo. Já para elementos do tipo quadrático (T6) foram necessários dois passos do processo adaptativo. Nos dois casos a convergência independe do uso da adaptatividade geométrica. Mas mesmo assim, percebe-se através do valor de $\bar{\eta}$ e do número de elementos gerados que o processo converge mais rapidamente quando se considera o efeito geométrico.

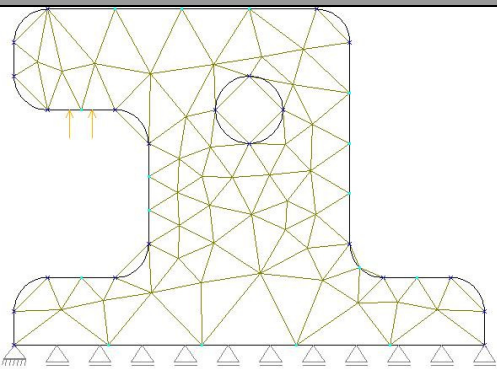
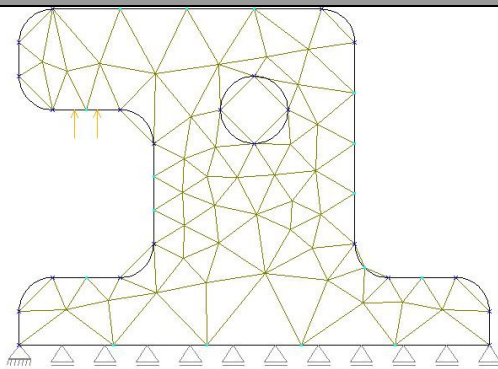
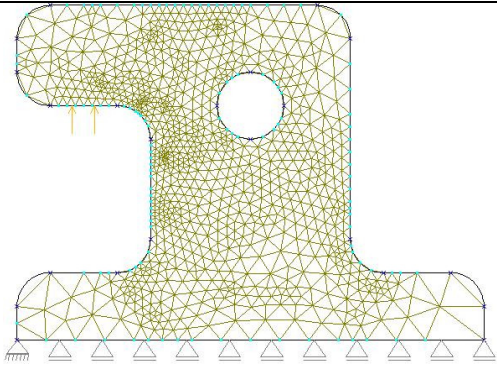
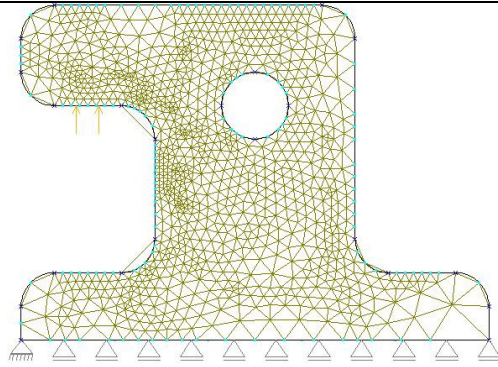
É evidente que a influência da adaptatividade geométrica varia conforme os valores estipulados para os parâmetros de entrada.

A Figura 32 apresenta as malhas com as suas características e os resultados da análise de erro para elementos lineares (T3) utilizando-se os dois métodos de suavização apresentados neste trabalho (*Z2-HC* [62] e *SPR* [63-66]). Nesta figura não se está utilizando a adaptatividade geométrica.

A Figura 33 mostra os aspectos das malhas com os elementos lineares (T3) usando-se a adaptatividade geométrica durante o processo adaptativo.

A Figura 34.b ilustra a malha resultante após se realizar um passo do processo adaptativo usando-se somente a adaptatividade geométrica, passando como parâmetros de análise os dados mostrados na Figura 34.a.

A Figura 35 mostra os aspectos das malhas deste exemplo usando elementos quadráticos (T6) e sem usar a adaptatividade geométrica. Já a Figura 36 mostra o aspecto das malhas com os elementos quadráticos (T6) usando-se a adaptatividade geométrica durante o processo adaptativo.

<i>Z2-HC</i>	<i>SPR</i>
	
(a) NN = 74, NE = 110 GDL = 141, $\bar{\eta} = 35.28\%$	(a) NN = 74, NE = 110 GDL = 141, $\bar{\eta} = 35.40\%$
	
(b) NN = 955, NE = 1772 GDL = 1890, $\bar{\eta} = 10.98\%$	(b) NN = 1067, NE = 1985 GDL = 2112, $\bar{\eta} = 11.77\%$

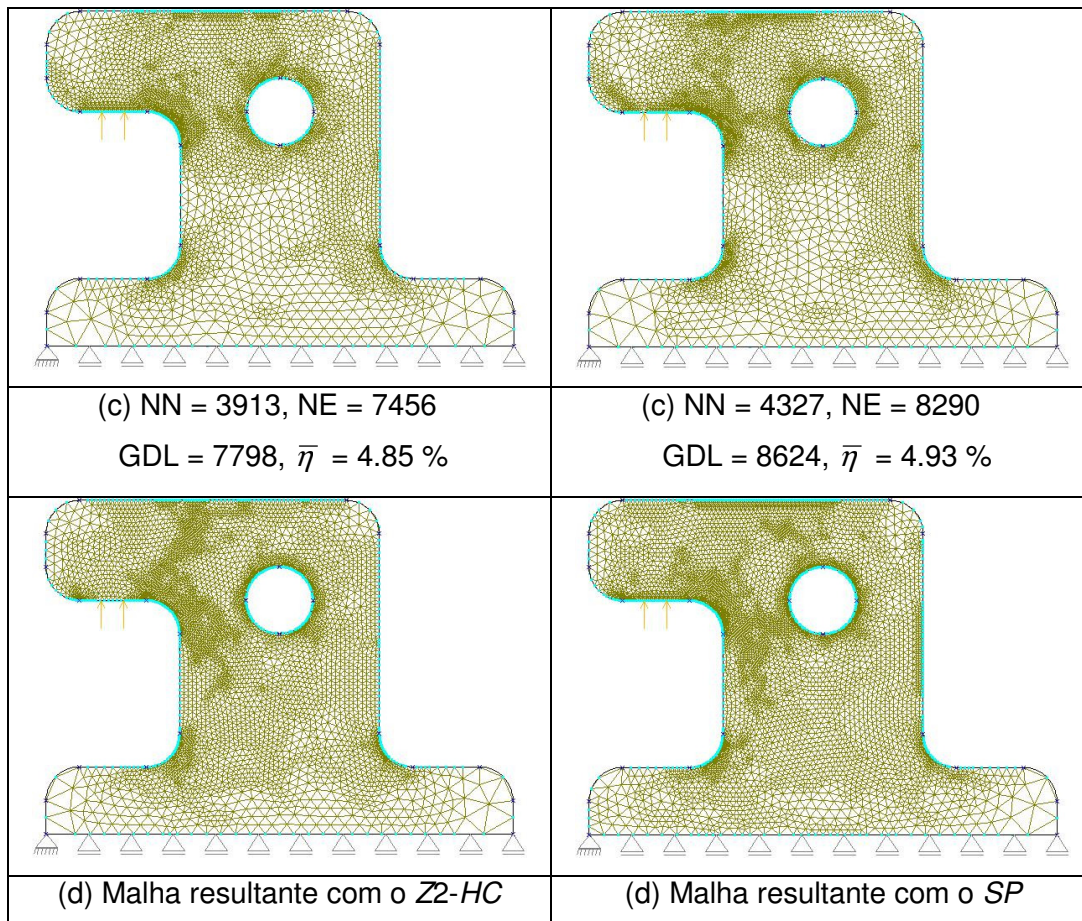


Figura 32 Comparação entre os métodos de suavização *Z2-HC* e *SPR*, utilizando-se elementos lineares (T3) usando apenas a adaptatividade numérica: malha inicial (a), malhas seguintes (b,c,d).

Na Figura 32 percebe-se que o método *Z2-HC* [62], que é menos complexo matematicamente e por isso é bem conhecido no meio técnico apresenta resultados bem próximos ao *SPR* [63-66]. Desta forma, todos os outros exemplos serão apresentados apenas usando-se o *Z2-HC* [62] como método de suavização, pois também, não é o objetivo deste trabalho comparar os métodos de suavização existentes.

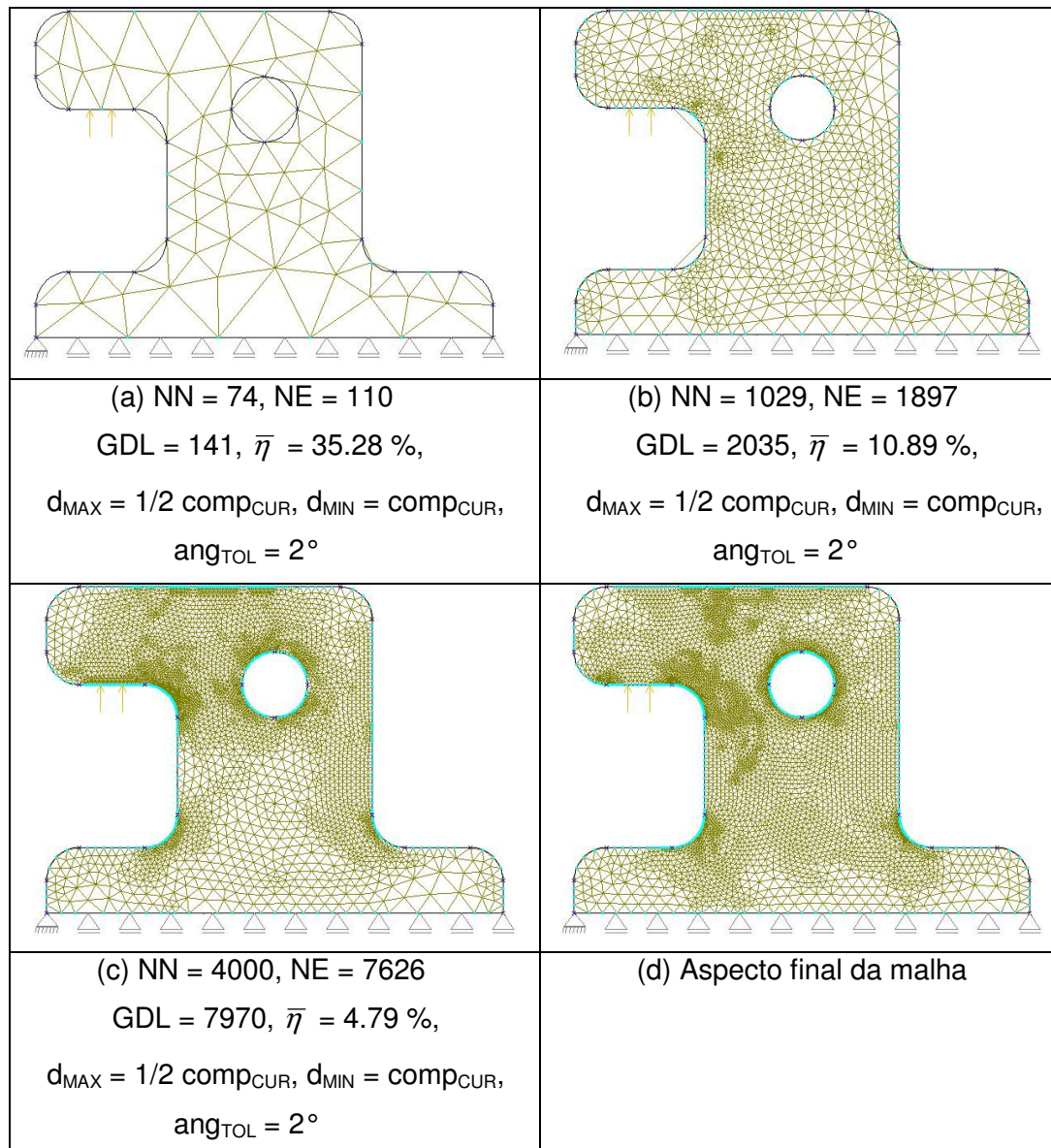


Figura 33 Exemplo 1 com T3 usando a adaptatividade geométrica e numérica: malha inicial (a), malhas seguintes (b,c,d).

Nos resultados da Figura 33 pode-se perceber que as curvas de bordo próximas a regiões que não apresentam grande concentrações de tensões foram mais refinadas do que as mesmas curvas quando se usou somente a adaptatividade numérica. Isto acontece justamente pelo uso da adaptatividade geométrica, pois desta forma se gera um número maior de elementos na malha e consecutivamente a análise de erro converge mais rapidamente.

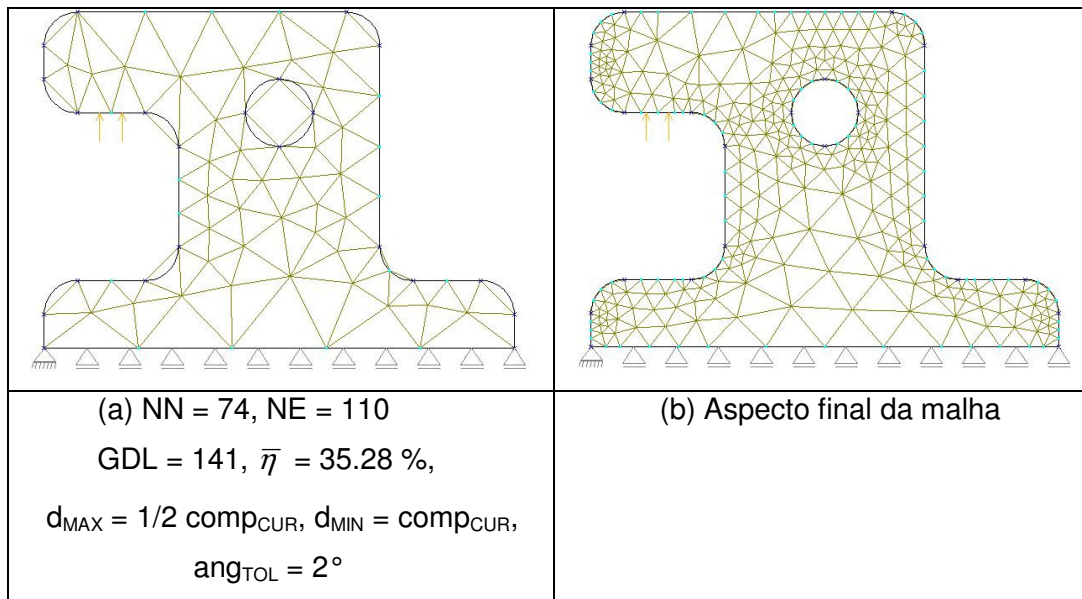


Figura 34 Exemplo 1 com T3 e usando apenas a adaptatividade geométrica: malha inicial (a), malha seguinte (b).

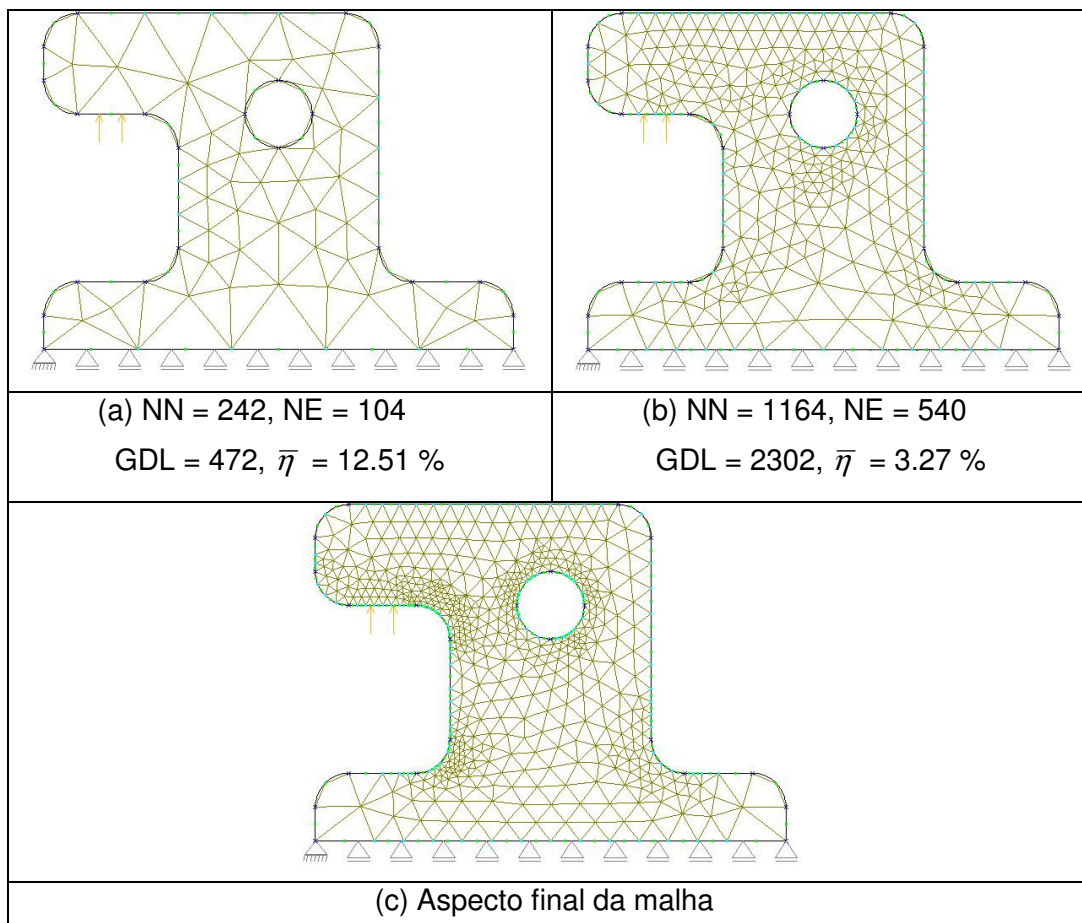


Figura 35 Exemplo 1 com T6 sem a adaptatividade geométrica mas com a adaptatividade numérica: malha inicial (a), malhas seguintes (b,c).

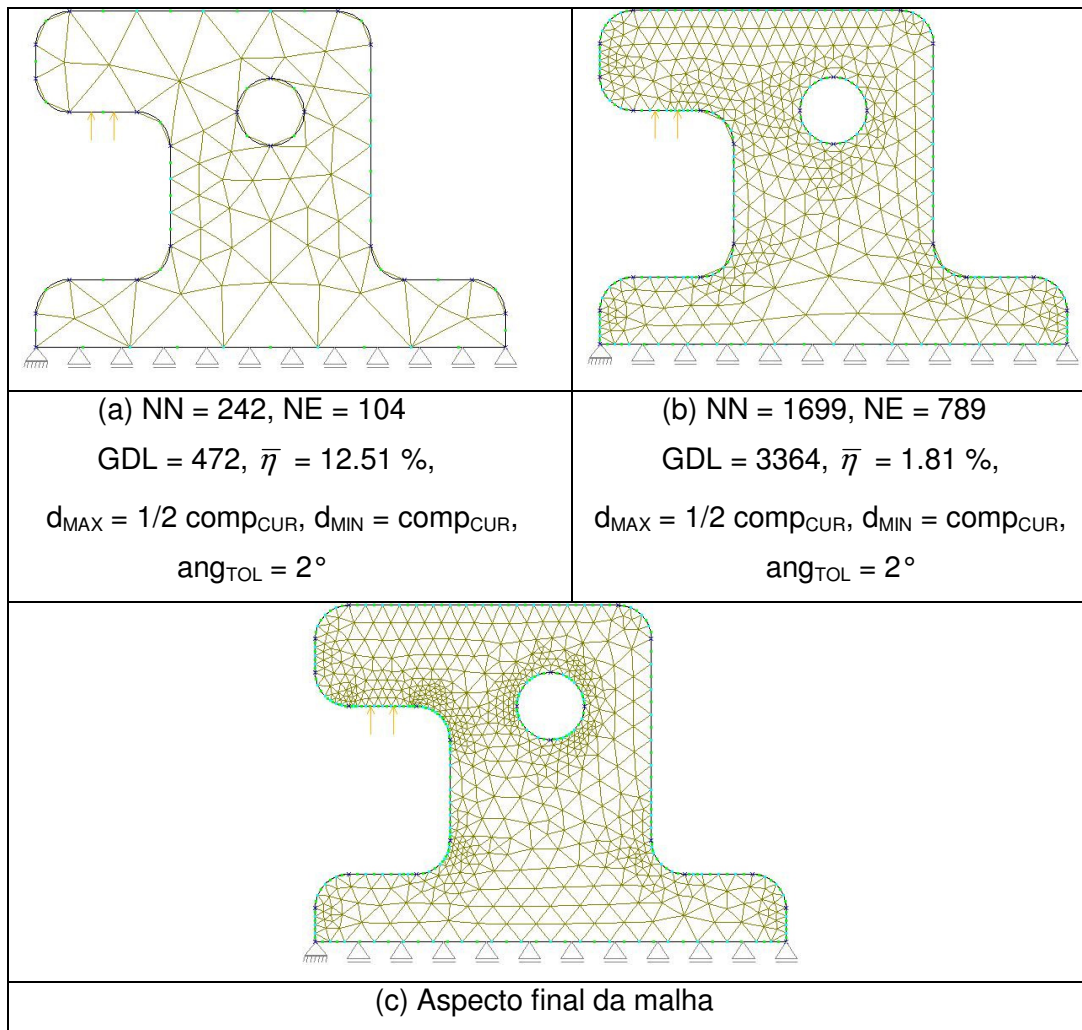


Figura 36 Exemplo 1 com T6 usando a adaptatividade geométrica e numérica: malha inicial (a), malhas seguintes (b,c).

Observando a Figura 36.b e c com a Figura 36.b e c, percebe-se que a principal diferença entre as malhas, na sua respectiva ordem do processo adaptativo, é o grau de refinamento das curvas do tipo arco de círculo. Dessa forma, nas malhas em que se considerou a adaptatividade geométrica, obteve-se um maior número de elementos e conseqüentemente uma convergência mais rápida.

5.1.2. Placa quadrada com furo quadrado

O segundo exemplo analisado é uma placa quadrada com um furo quadrado, submetida a um carregamento lateral unitário. Devido às condições de simetria apenas um quarto da placa foi modelado. Este exemplo é mostrado para demonstrar a aplicabilidade do processo adaptativo proposto para multiregiões. Considerou-se estado plano de tensões, com módulo de elasticidade $E = 10^5$, coeficiente de Poisson $\nu = 0.3$ e espessura $e = 1$. Neste exemplo, considerou-se três regiões, para que fosse possível ter condições iniciais iguais as que foram obtidas por Cavalcante-Neto [13]. A Figura 37 mostra as características do problema.

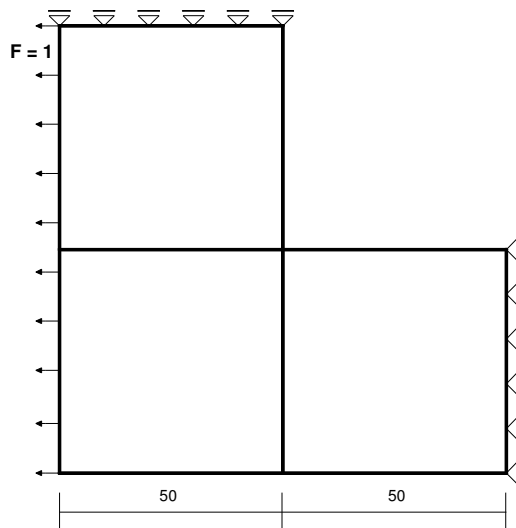


Figura 37 Placa quadrada com furo quadrado.

Para este exemplo foram necessários três passos do processo adaptativo usando o T3 e apenas dois passos usando o T6 para satisfazer a equação (17). No trabalho de Cavalcante-Neto [12], também foram necessários três passos para os dois tipos de elementos satisfazerem a referida equação.

A Figura 38 e a Figura 39 mostram os aspectos das malhas com os elementos lineares (T3) e quadráticos (T6) com as suas características, número de nós (NN), número de elementos (NE) e graus de liberdade (GDL).

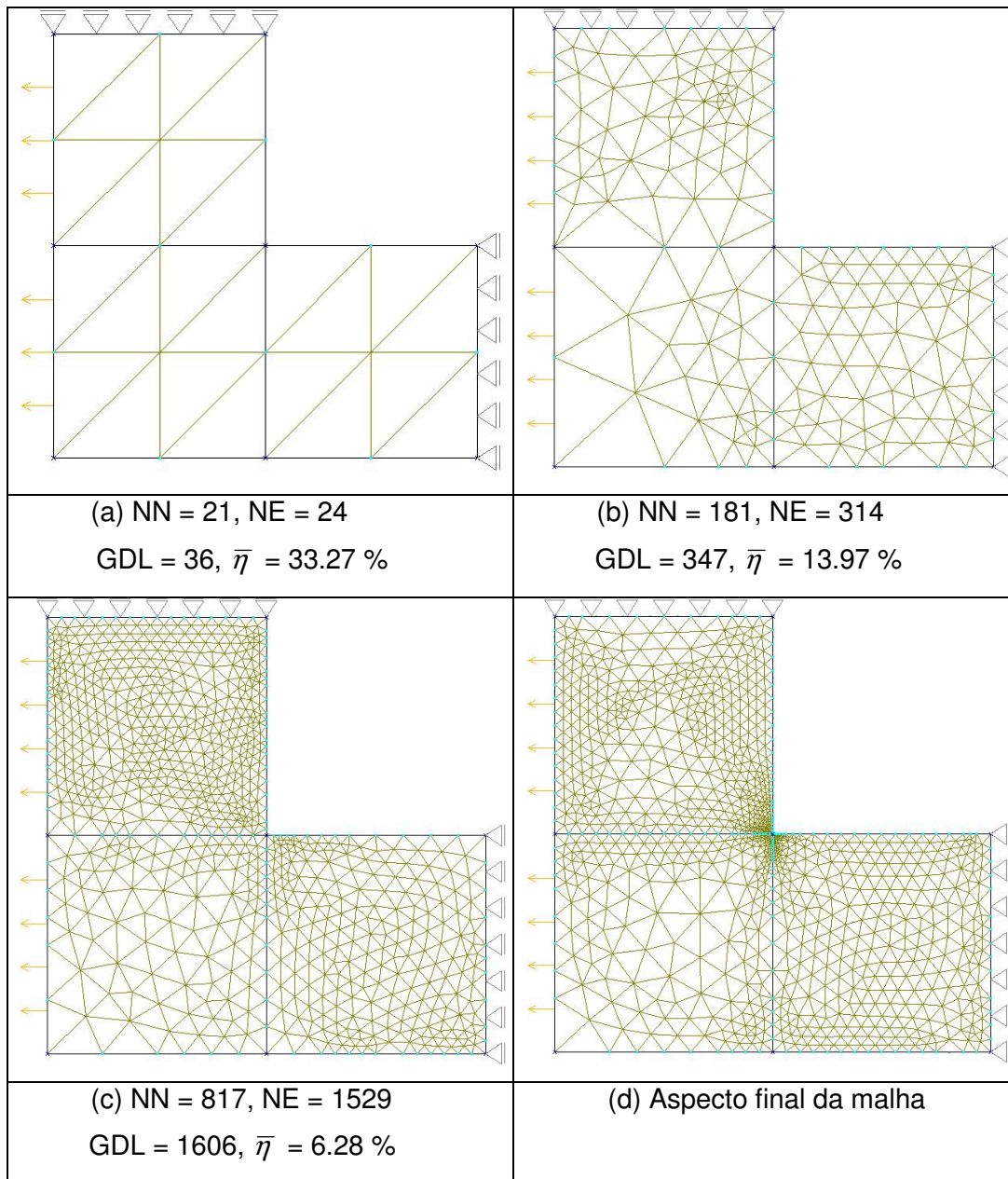


Figura 38 Exemplo 2 com T3: malha inicial (a), malha seguinte (b,c,d).

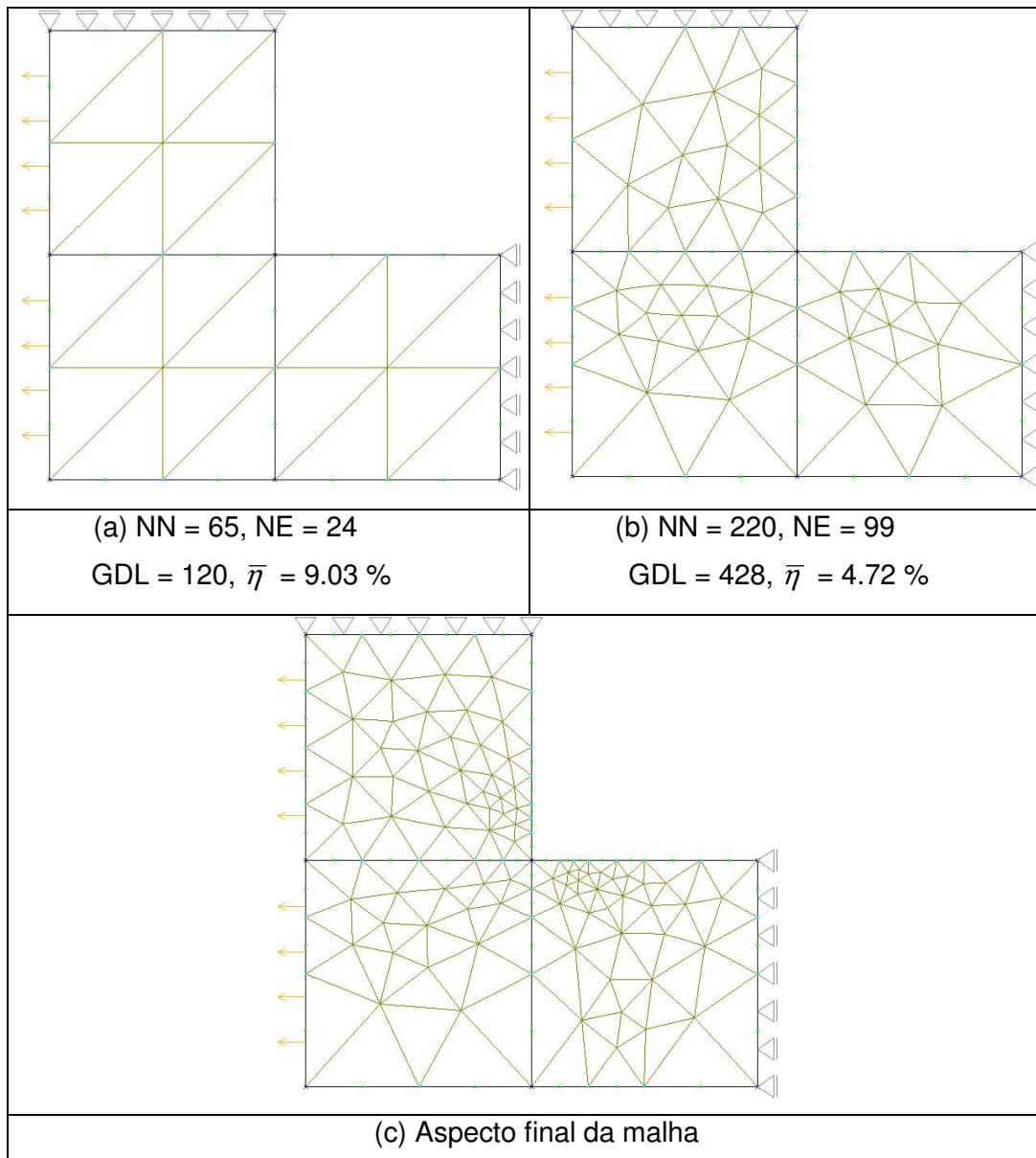


Figura 39 Exemplo 2 com T6: malha inicial (a), malhas seguintes (b,c).

A Tabela 1 compara os resultados obtidos deste trabalho com os de Cavalcante-Neto [12].

Tabela 1 Comparação dos resultados para a placa quadrada com furo quadrado.

	Cavalcante-Neto	Processo adaptativo proposto
1º	Elemento linear (T3)	
NN	21	21
NE	24	24
GDL	36	36
$\bar{\eta}$	33.27 %	33.27 %
2º	Elemento linear (T3)	
NN	240	181
NE	413	314
GDL	464	347
$\bar{\eta}$	13.45 %	13.97 %
3º	Elemento linear (T3)	
NN	693	817
NE	1265	1529
GDL	1365	1606
$\bar{\eta}$	7.77 %	6.28 %
1º	Elemento quadrático (T6)	
NN	65	65
NE	24	24
GDL	120	120
$\bar{\eta}$	9.02 %	9.03 %
2º	Elemento quadrático (T6)	
NN	141	220
NE	60	99
GDL	272	428
$\bar{\eta}$	5.77 %	4.72 %

Na Tabela 1, a primeira coluna contém os números correspondentes ao passo do processo adaptativo.

Os valores do erro corrente $\bar{\eta}$ obtidos pelo processo adaptativo proposto reduzem mais rapidamente. Isto pode ser visto pelos valores de $\bar{\eta}$ em cada passo do processo. Apenas no segundo passo do caso linear foi que a formulação proposta apresentou um valor de $\bar{\eta}$ maior que o resultado obtido por Cavalcante-Neto.

5.1.3. Viga curta em balanço

O terceiro exemplo é uma viga curta em balanço sob carregamento distribuído unitário no seu bordo superior. Considera-se para a solução deste problema estado plano de deformação com módulo de elasticidade $E=1$, coeficiente de Poisson $\nu=0.3$ e espessura $e=1$. A Figura 40 mostra as características do problema.

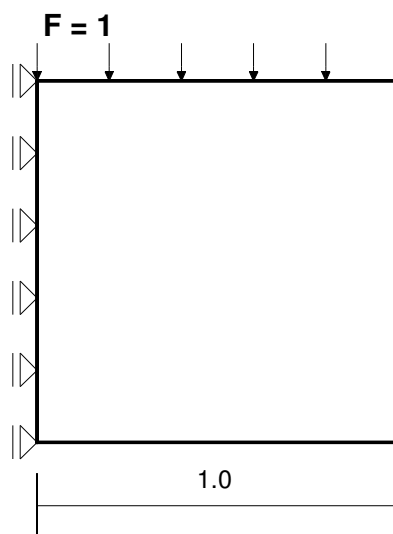


Figura 40 Viga curta em balanço.

Para este exemplo foram necessários três passos do processo adaptativo tanto usando o T3 quanto o T6 para satisfazer a equação (17). No trabalho de Cavalcante-Neto [12] também foram necessários três passos para que as malhas com os dois tipos de elementos pudessem satisfazer a referida equação.

A Figura 41 e a Figura 43 mostram os aspectos das malhas com os elementos lineares (T3) e quadráticos (T6) com as suas características, número de nós (NN), número de elementos (NE) e graus de liberdade (GDL).

Este exemplo também é muito bom para ilustrar como ocorre a distribuição das razões de erro (ξ_i) entre os elementos da malha conforme a evolução do processo adaptativo. Desta forma, a Figura 42 mostra esta distribuição de erro conforme uma tabela de cores referente aos valores de (ξ_i) para cada elemento. Como o processo adaptativo se propõe a homogeneizar estes valores para todos os elementos, espera-se que no final do processo a figura esteja praticamente de uma cor só.

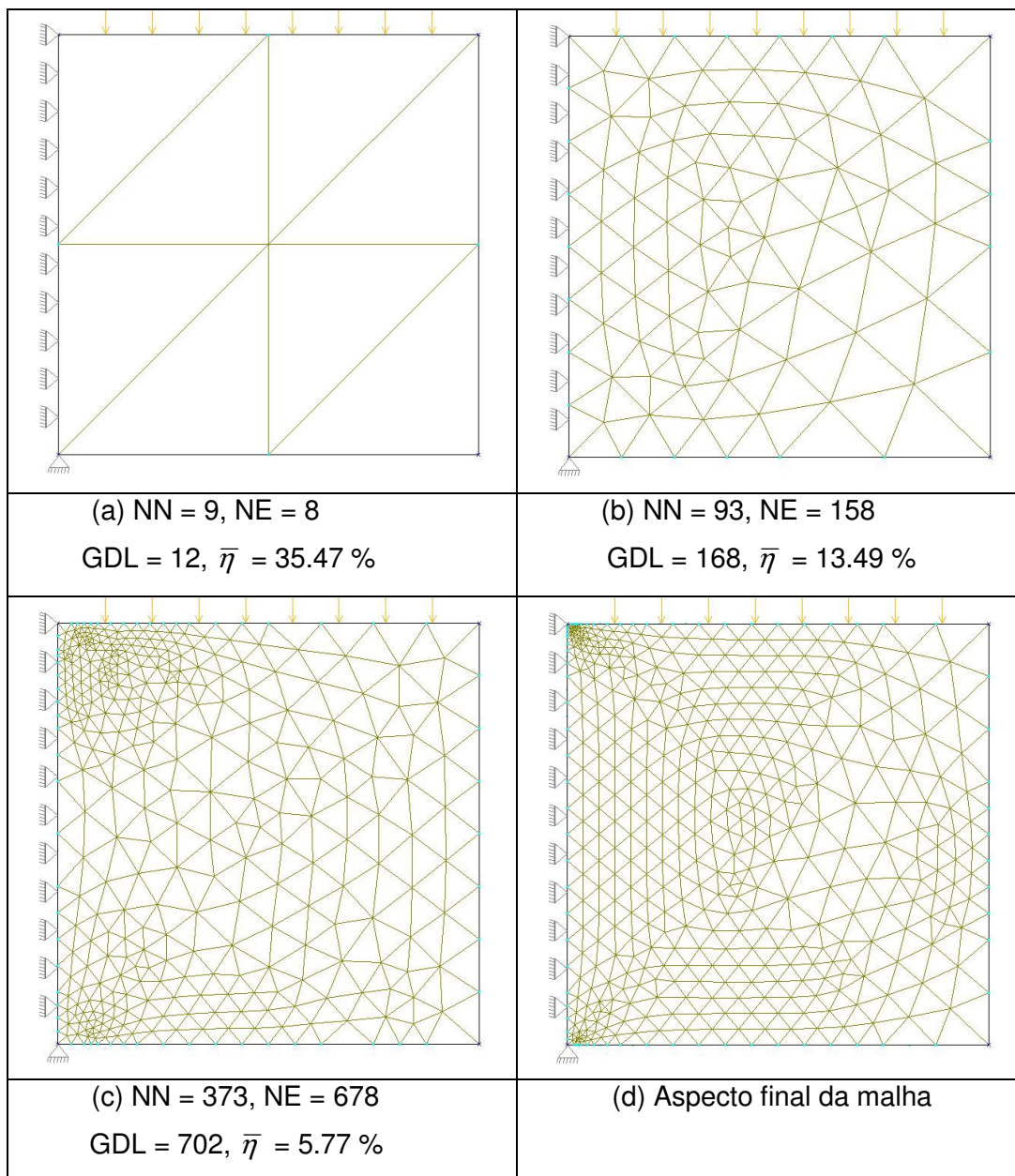


Figura 41 Exemplo 3 com T3: malha inicial (a), malhas seguintes (b,c,d).

Na Figura 41.b a malha não está mostrando maior discretização nos cantos que apresentam concentração de tensões (canto esquerdo superior e inferior), o que só começa a ser verificado a partir da Figura 41.c, quando o critério de convergência já é atendido, conforme a equação (17).

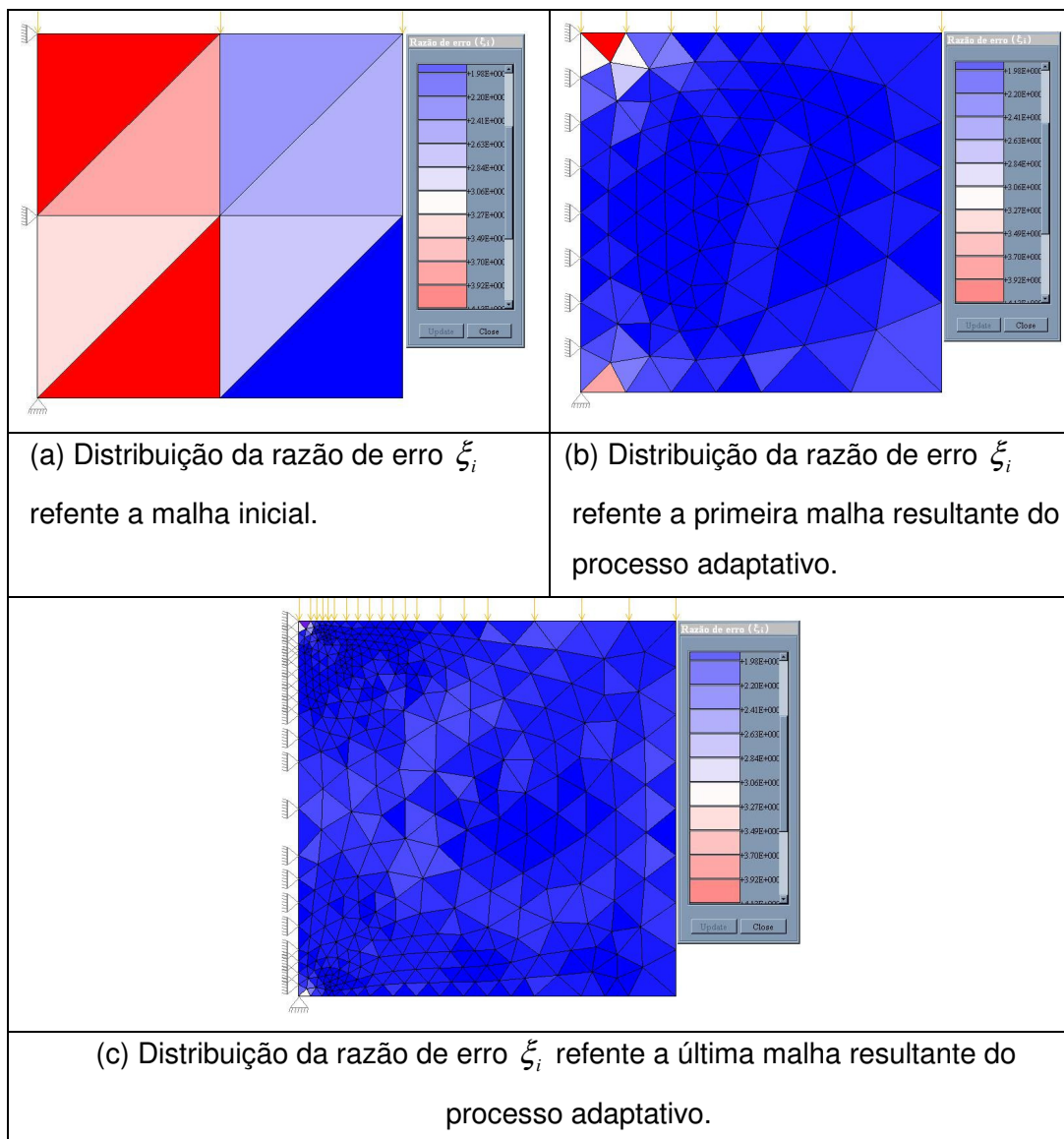


Figura 42 Mostra a distribuição da razão de erro ξ_i entre os elementos da malha durante os passos do processo adaptativo.

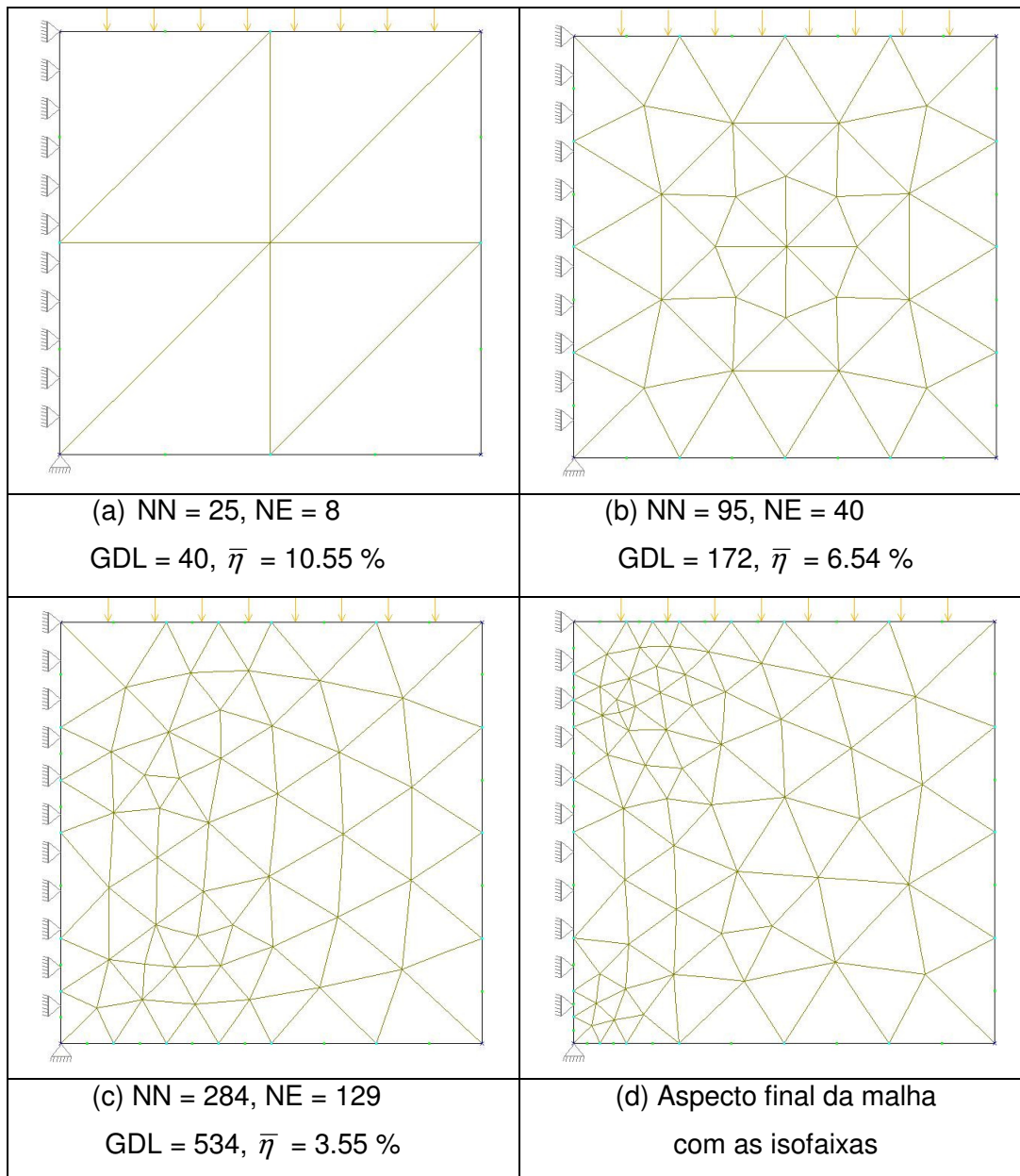


Figura 43 Exemplo 3 com T6: malha inicial (a), malhas seguintes (b,c,d).

Na Figura 43.b a malha não está apresentando maior discretização nos cantos que apresentam concentração de tensões (canto esquerdo superior e inferior), o que só é observado a partir da Figura 43.d quando já foi satisfeito o critério de convergência, conforme a equação (17).

A Tabela 2 compara os resultados obtidos deste trabalho com os de Cavalcante-Neto [12].

Tabela 2 Comparação dos resultados para a viga curta em balanço.

	Cavalcante-Neto	Processo adaptativo proposto
1º	Elemento linear (T3)	
NN	9	9
NE	8	8
GDL	12	12
$\bar{\eta}$	35.19 %	35.47 %
2º	Elemento linear (T3)	
NN	128	93
NE	214	158
GDL	230	168
$\bar{\eta}$	11.80 %	13.49 %
3º	Elemento linear (T3)	
NN	525	373
NE	940	678
GDL	970	702
$\bar{\eta}$	5.64 %	5.77 %
1º	Elemento quadrático (T6)	
NN	25	25
NE	8	8
GDL	40	40
$\bar{\eta}$	11.08 %	10.55 %
2º	Elemento quadrático (T6)	
NN	68	95
NE	27	40
GDL	118	172
$\bar{\eta}$	6.15 %	6.54 %
3º	Elemento quadrático (T6)	
NN	141	284
NE	60	129
GDL	256	534
$\bar{\eta}$	3.83 %	3.55 %

Os valores do erro corrente $\bar{\eta}$ obtidos pelo processo adaptativo proposto reduzem mais lentamente para o caso linear e mais rapidamente para o caso quadrático. Isto pode ser visto pelos valores de $\bar{\eta}$ em cada passo do processo. Apenas no segundo passo, do caso quadrático, foi que o valor de $\bar{\eta}$ da formulação proposta foi maior do que o valor obtido por Cavalcante-Neto.

5.1.4. Placa com furo quadrado com dois materiais

O quarto exemplo é a mesma placa do exemplo 2, mas considerando dois materiais diferentes. Este exemplo é mostrado para demonstrar a aplicabilidade do processo adaptativo proposto para regiões de materiais distintos. A diferença é dada no módulo de elasticidade das regiões, onde uma ficou sendo o triplo da outra, tal que $E_1 = 10^5$, $E_2 = 3 \cdot 10^5$, o coeficiente de Poisson $\nu = 0.3$ e espessura $e = 1$ para ambos os materiais. A Figura 44 mostra as características do problema.

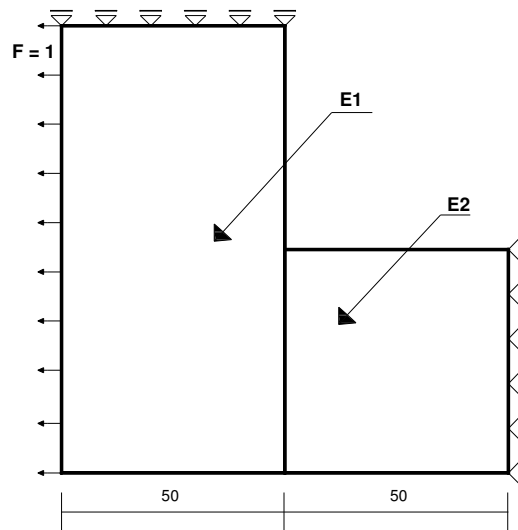


Figura 44 Placa quadrada com furo quadrado com dois materiais.

Para este exemplo foram necessários três passos do processo adaptativo tanto usando o T3 quanto usando o T6 para satisfazer a equação (17). No trabalho de Cavalcante-Neto [12] foram necessários três passos para que as malhas com os dois tipos de elementos pudessem satisfazer a referida equação.

A Figura 45 e a Figura 46 mostram os aspectos das malhas com os elementos lineares (T3) e quadráticos (T6) com as suas características, número de nós (NN), número de elementos (NE) e graus de liberdade (GDL).

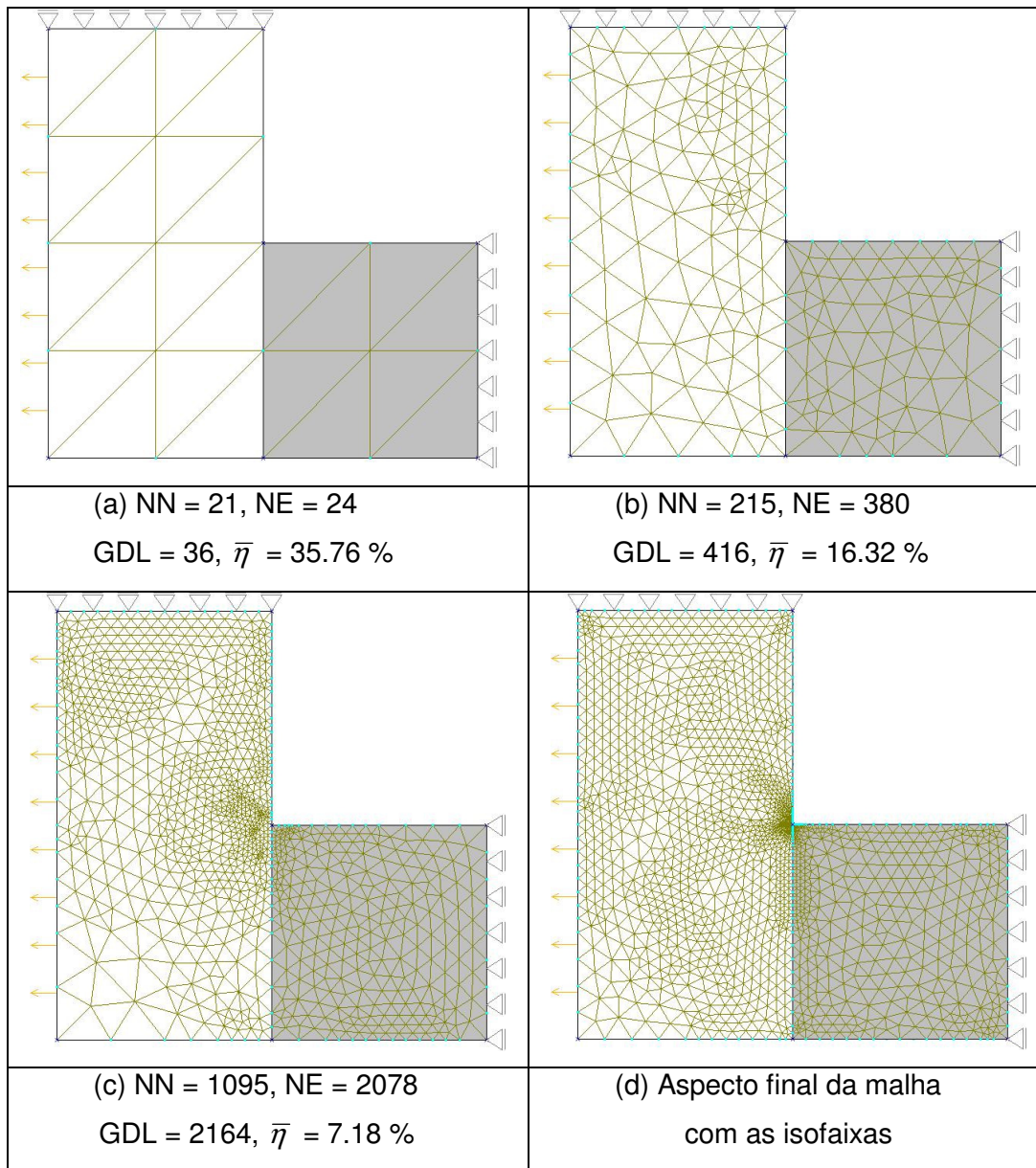


Figura 45 Exemplo 4 com T3: malha inicial (a), malhas seguintes (b,c,d).

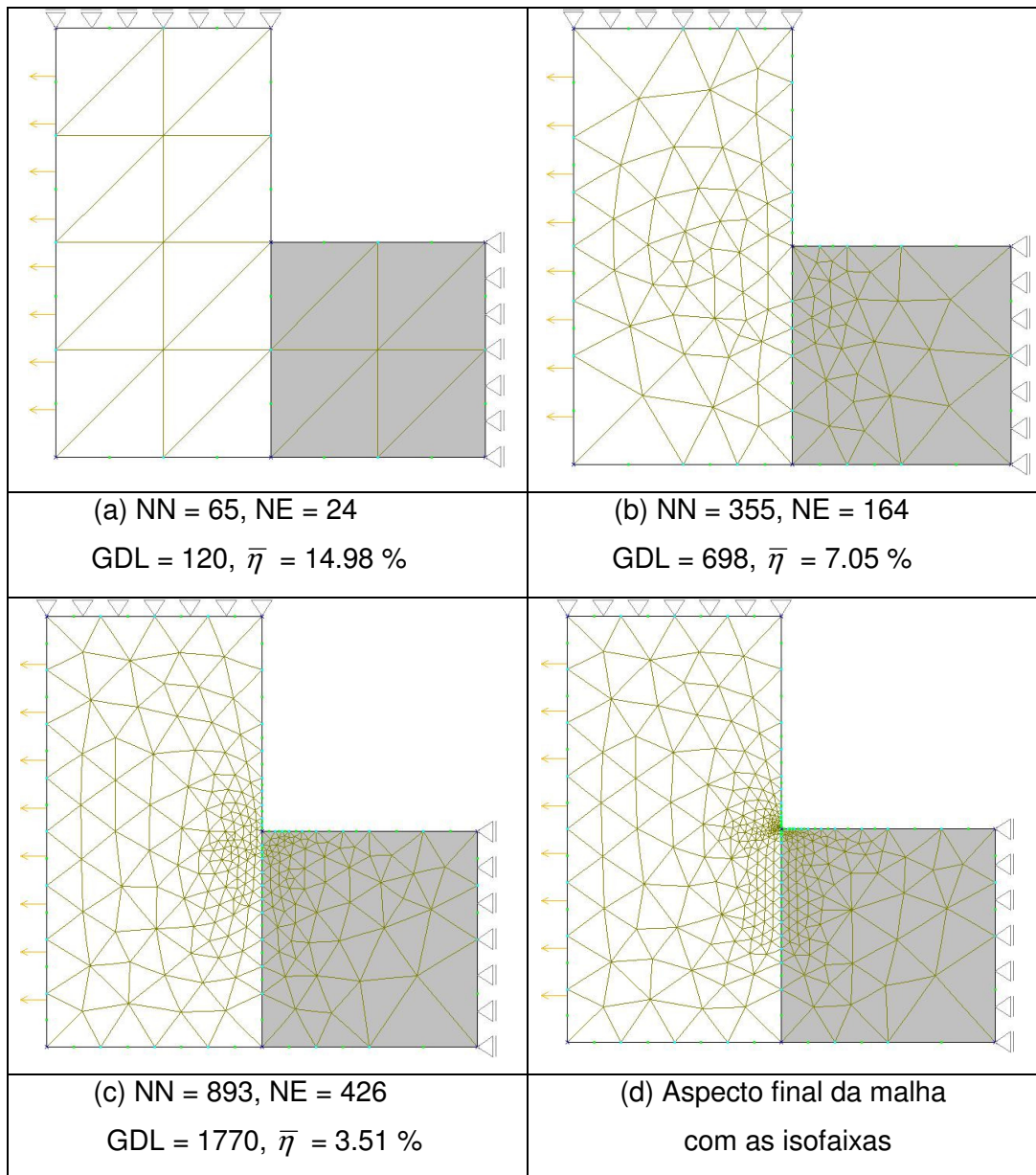


Figura 46 Exemplo 4 com T6: malha inicial (a), malhas seguintes (b,c,d).

A Tabela 3 compara os resultados obtidos deste trabalho com os de Cavalcante-Neto [12].

Tabela 3 Comparação dos resultados para a placa quadrada com furo quadrado com dois materiais diferentes.

	Cavalcante-Neto	Processo adaptativo proposto
1º	Elemento linear (T3)	
NN	21	21
NE	24	24
GDL	36	36
$\bar{\eta}$	35.76 %	35.76 %
2º	Elemento linear (T3)	
NN	286	215
NE	503	380
GDL	558	416
$\bar{\eta}$	14.41 %	16.32 %
3º	Elemento linear (T3)	
NN	1077	1095
NE	1999	2078
GDL	2135	2164
$\bar{\eta}$	7.33 %	7.18 %
1º	Elemento quadrático (T6)	
NN	65	65
NE	24	24
GDL	120	120
$\bar{\eta}$	19.37 %	14.98 %
2º	Elemento quadrático (T6)	
NN	308	355
NE	141	164
GDL	606	698
$\bar{\eta}$	6.57 %	7.05 %
3º	Elemento quadrático (T6)	
NN	629	893
NE	296	426
GDL	1246	1770
$\bar{\eta}$	3.26 %	3.51 %

Nesta tabela pode-se perceber que para o caso linear, ambas as formulações fornecem o mesmo valor de $\bar{\eta}$ para o primeiro passo, no segundo passo o valor obtido de $\bar{\eta}$ foi maior para a formulação proposta. Já para o terceiro passo, o valor de $\bar{\eta}$ obtido pela formulação proposta, foi menor.

No caso quadrático apenas no primeiro passo o valor de $\bar{\eta}$ obtido pela formulação proposta foi menor que o valor encontrado por Cavalcante-Neto.

5.1.5. Placa complexa com três furos circulares e um quadrado

O quinto exemplo é a análise de uma placa complexa com três furos circulares e um quadrado sob um carregamento lateral unitário. Este exemplo é importante para mostrar a eficácia do processo adaptativo proposto para problemas com geometria complexa. Considera-se para a solução deste problema estado plano de deformação com $E = 10^5$, $\nu = 0.3$ e espessura $e = 1$. A Figura 47 mostra as características do problema.

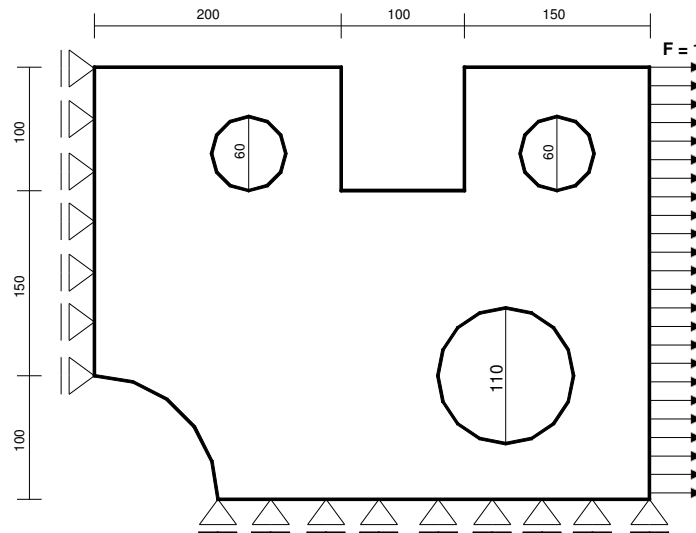


Figura 47 Placa complexa com três furos circulares e um quadrado.

Para este exemplo foram necessários três passos do processo adaptativo quando se usou o T3 e dois passos quando se usou o T6 para satisfazer a equação (17). No trabalho de Cavalcante-Neto [12] também foram necessários três passos para satisfazer a referida equação. Esse autor só analisou este problema para o caso quadrático (T6).

A Figura 48 e a Figura 49 mostram os aspectos das malhas com os elementos lineares (T3) e quadráticos (T6) com as suas características, número de nós (NN), número de elementos (NE) e graus de liberdade (GDL).

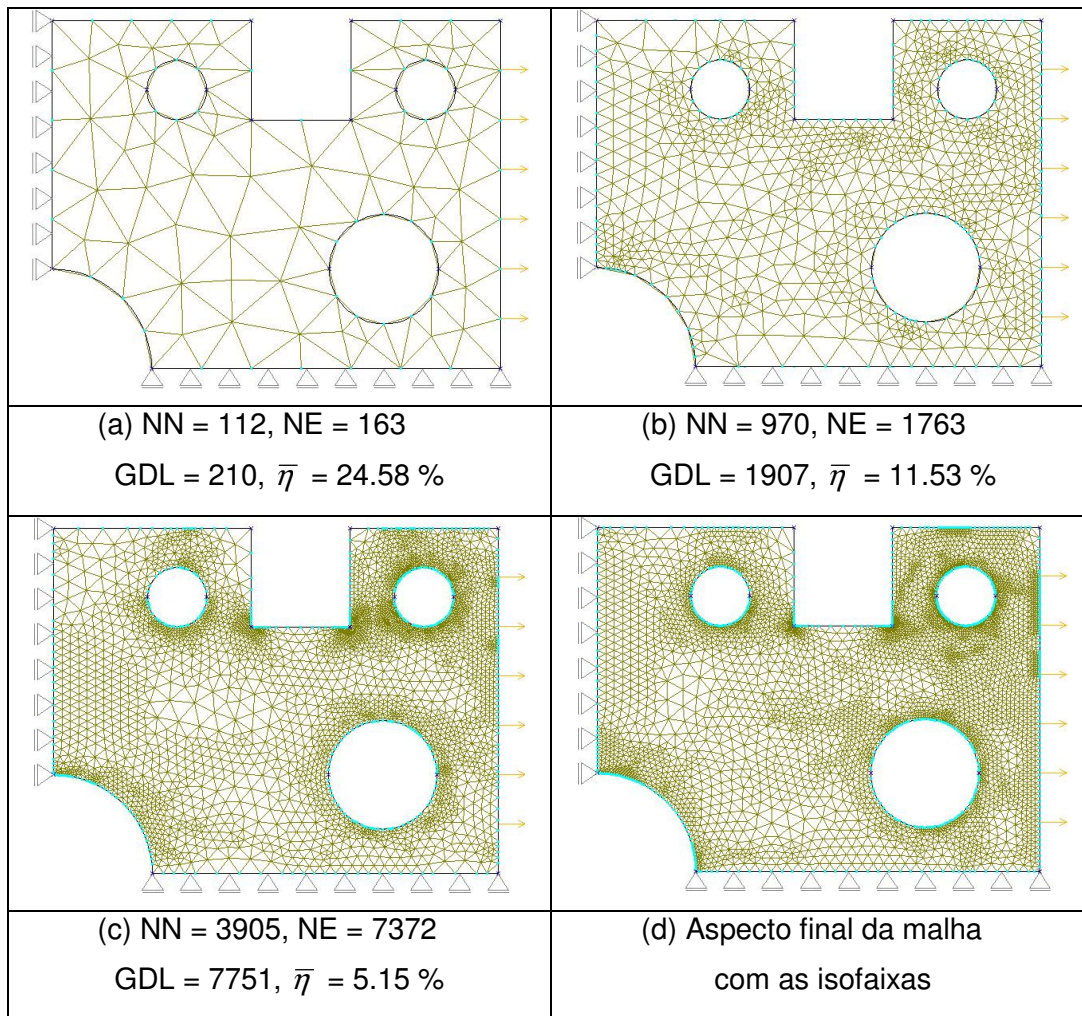


Figura 48 Exemplo 5 com T3: malha inicial (a), malhas seguintes (b,c,d).

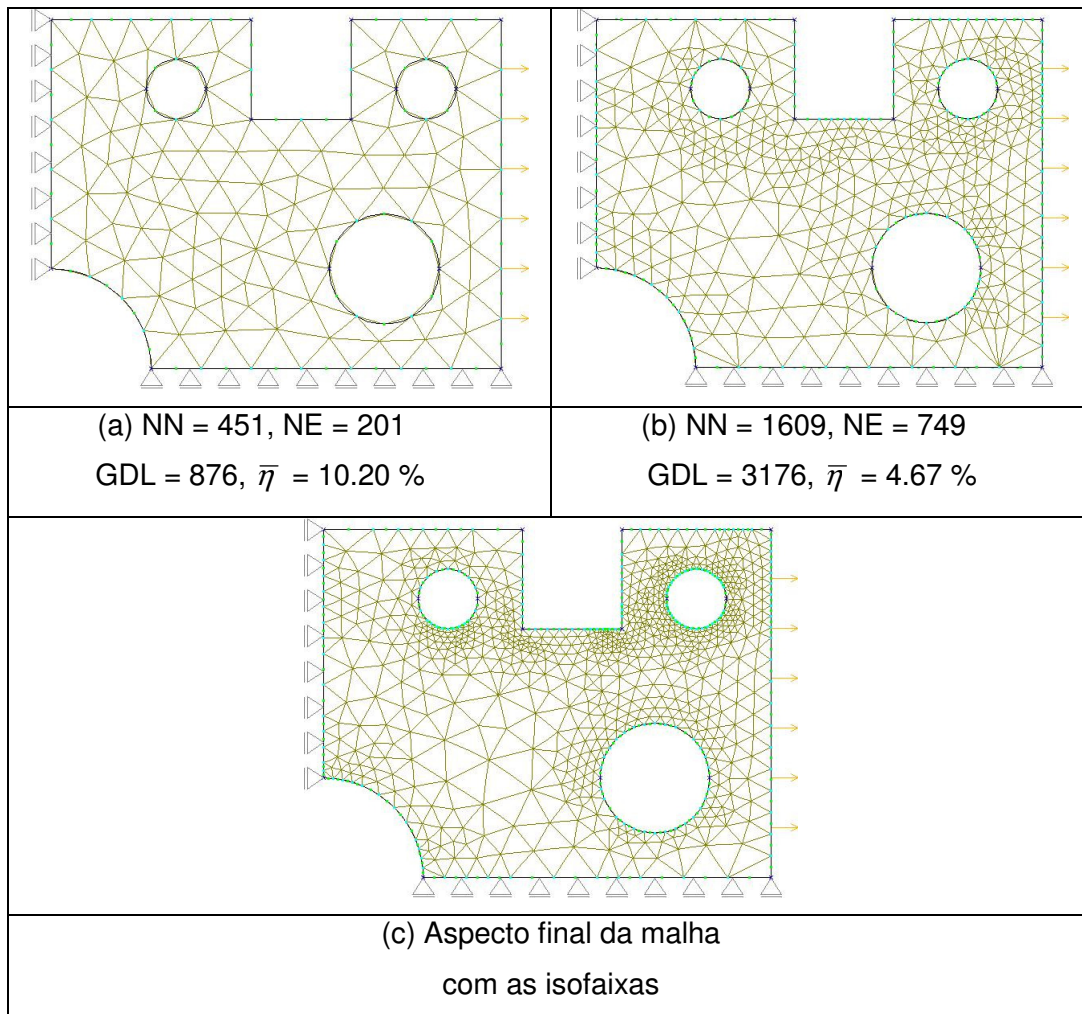


Figura 49 Exemplo 5 com T6: malha inicial (a), malhas seguintes (b,c).

A Tabela 4 compara os resultados obtidos para o caso quadrático deste trabalho com os de Cavalcante-Neto [12].

Tabela 4 Comparação dos resultados para a placa complexa com três furos circulares e um quadrado.

	Cavalcante-Neto	Processo adaptativo proposto
1º	Elemento linear (T6)	
NN	275	451
NE	113	201
GDL	524	876
$\bar{\eta}$	12.47 %	10.20 %
2º	Elemento linear (T6)	
NN	953	1609
NE	431	749
GDL	1878	3176
$\bar{\eta}$	5.71 %	4.67 %

Percebe-se que no processo adaptativo proposto os valores de $\bar{\eta}$ obtidos a cada passo são menores do que os obtidos por Cavalcante-Neto, porém como as características iniciais não foram equivalentes, não se pode afirmar com certeza em qual caso se obteve a convergência de forma mais rápida.

5.1.6. Barragem

O sexto exemplo é a análise de uma barragem bidimensional submetida a pressão hidrostática. Este exemplo é importante para mostrar a eficácia da utilização do processo adaptativo proposto para problemas com geometria complexa. Este exemplo também foi analisado por Zienkiewicz [66]. Considera-se para a solução deste problema estado plano de deformação com $E = 1$, $\nu = 0.3$ e espessura $e = 1$. A Figura 50 mostra as características do problema.

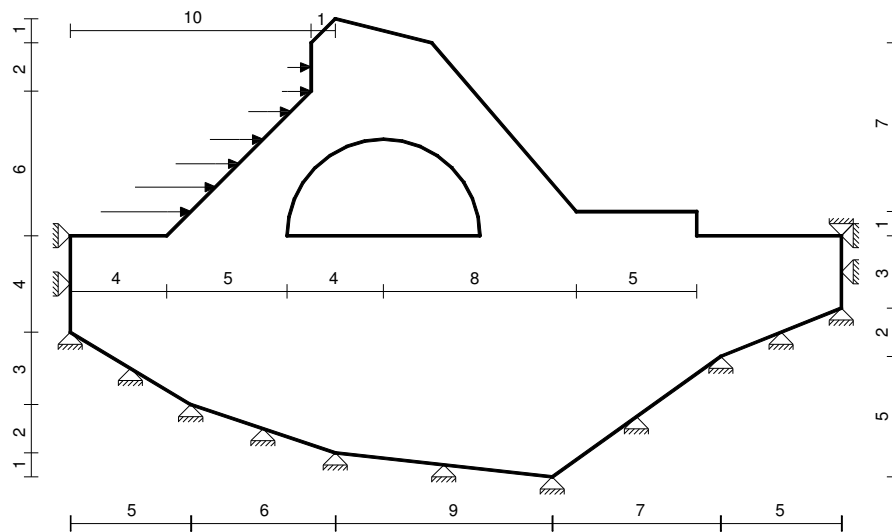


Figura 50 Barragem bidimensional.

Para este exemplo foram necessários três passos do processo adaptativo tanto quando se usou elementos do tipo T3 como quando se usou o T6. O mesmo número de passos foi obtido por Zienkiewicz [66] para elementos quadráticos, porém o número de elementos gerados pelo método proposto é bem maior, fato que pode ser observado na Tabela 5.

A Figura 51 e a Figura 52 mostram os aspectos das malhas com os elementos lineares (T3) e quadráticos (T6) com as suas características, número de nós (NN), número de elementos (NE) e graus de liberdade (GDL).

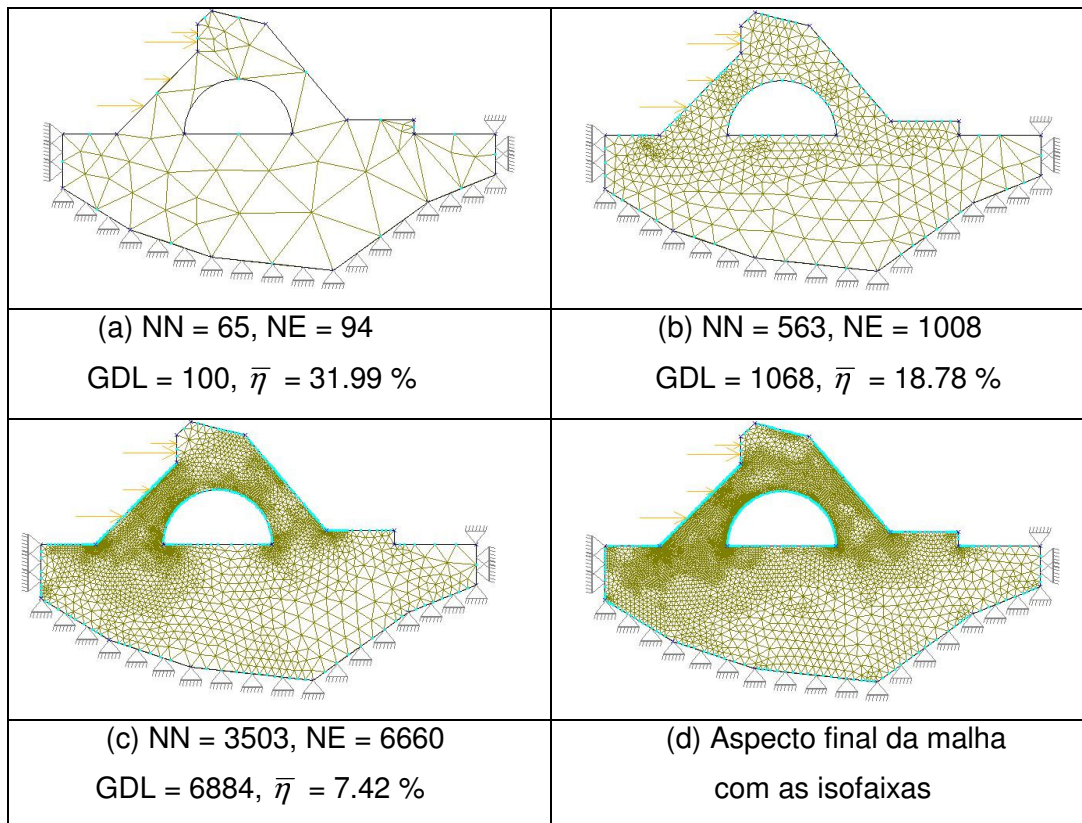


Figura 51 Exemplo 6 com T3: malha inicial (a), malhas seguintes (b,c,d).

A Figura 51.c mostra a malha do processo adaptativo. Nesta mesma figura, pode-se observar a influência da qualidade do elemento linear, pois mostra um refinamento em regiões em que não ocorrem altas concentrações de tensões, o que ocorre em menor intensidade para os elementos quadráticos, conforme a Figura 52.b.

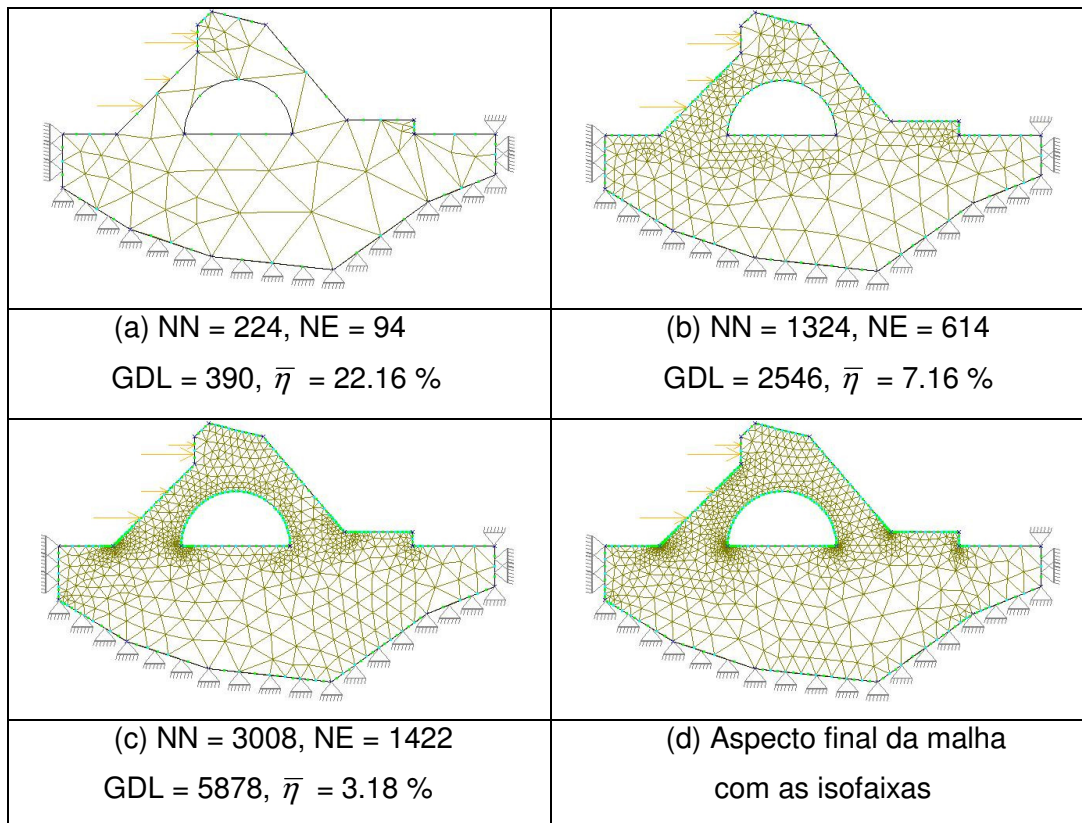


Figura 52 Exemplo 6 com T6: malha inicial (a), malhas seguintes (b,c,d).

A Tabela 5 compara os resultados obtidos para o caso quadrático deste trabalho com os de Zienkiewicz [66].

Tabela 5 Comparação dos resultados do exemplo bidimensional da barragem, mostrando os valores obtidos por Zienkiewicz e o método proposto.

	Zienkiewicz	Processo adaptativo proposto
1º	Elemento quadrático (T6)	
NN	—	224
NE	—	94
GDL	728	390
$\bar{\eta}$	16.50 %	22.16 %
2º	Elemento quadrático (T6)	
NN	—	1324
NE	—	614
GDL	1764	2546
$\bar{\eta}$	4.90 %	7.16 %
3º	Elemento quadrático (T6)	
NN	—	3008
NE	—	1422
GDL	—	5878
$\bar{\eta}$	—	3.18 %

Percebe-se que no processo adaptativo proposto os valores de $\bar{\eta}$ obtidos a cada passo são maiores do que os obtidos por Zienkiewicz. Mas como as características iniciais não foram equivalentes, não se pode afirmar com certeza em qual caso se obteve a convergência de forma mais rápida.

5.2. Exemplos em 3D (MG)

Foram usados cinco exemplos com o objetivo de validar a implementação do processo adaptativo no MG [48]. Em todos os exemplos se fez apenas análise para elementos quadráticos (TETRA10).

Deve-se ressaltar que os exemplos que também foram analisados por Cavalcante-Neto [13], não foram analisados de forma completamente automática, requerendo do usuário interferências em cada passo para a construção das malhas.

5.2.1. Viga curta em balanço

O primeiro exemplo é uma viga curta em balanço, fixa em uma extremidade e submetida a uma pressão distribuída vertical na sua extremidade livre. Este problema também foi analisado por Cavalcante-Neto [13]. Adotou-se o valor de $\eta^* = 5\%$, considerando para a solução deste problema módulo de elasticidade $E = 1$ e coeficiente de Poisson $\nu = 0.3$. A Figura 53 mostra as características do problema.

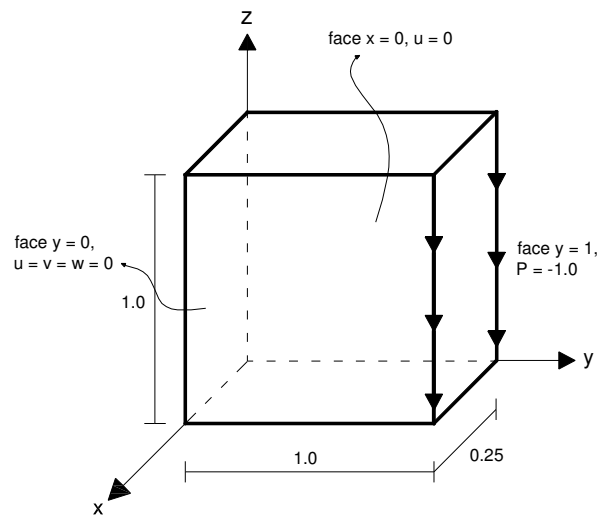


Figura 53 Viga curta em balanço.

Para este exemplo foram necessários dois passos do processo adaptativo. O mesmo número de passos foi obtido por Cavalcante-Neto [13], porém o número de elementos gerados pelo método proposto é bem maior, fato que pode ser observado na Tabela 6.

A Figura 54 mostra o aspecto das malhas para cada passo do processo adaptativo com os elementos quadráticos (TETRA10) e com as suas características, número de nós (NN), número de elementos (NE) e graus de liberdade (GDL).

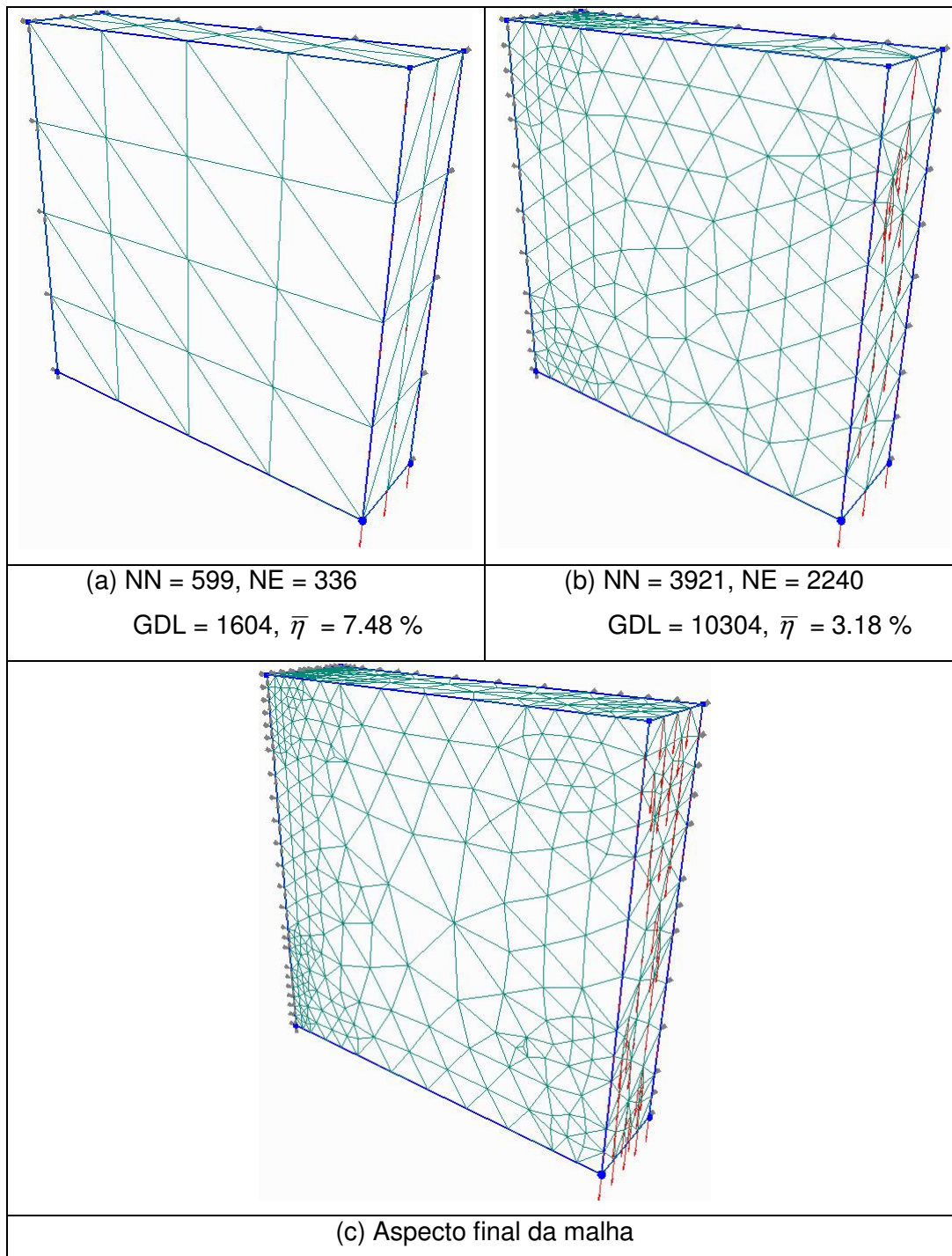


Figura 54 Exemplo 1 com TETRA10: malha inicial (a), malhas seguintes (b,c).

A Tabela 6 compara os resultados obtidos para o caso quadrático deste trabalho com os de Cavalcante-Neto [13].

Tabela 6 Comparação dos resultados do exemplo da viga curta em balanço 3D.

	Cavalcante-Neto	Processo adaptativo proposto
1º	Elemento quadrático (TETRA10)	
NN	386	599
NE	177	336
GDL	807	1604
$\bar{\eta}$	7.90 %	7.48 %
2º	Elemento quadrático (TETRA10)	
NN	1122	3921
NE	609	2240
GDL	2625	10304
$\bar{\eta}$	4.20 %	3.18 %

Na Tabela 6 a primeira coluna contém os números correspondentes ao passo do processo adaptativo e os dados com as características das malhas.

Percebe-se que no processo adaptativo proposto os valores de $\bar{\eta}$ obtidos a cada passo são menores do que os obtidos por Cavalcante-Neto. Porém o número de elementos é maior. Portanto, como as características iniciais não foram equivalentes, não se pode afirmar com certeza em qual caso se obteve a convergência de forma mais rápida.

5.2.2. Placa com furo circular

O segundo exemplo é uma placa fina com um furo circular, submetida a uma tensão axial. Devido à simetria, apenas um quarto da placa foi analisado. Este problema também foi analisado por Cavalcante-Neto [13]. Adotou-se o valor de $\eta^* = 5\%$, considerando para a solução deste problema módulo de elasticidade $E = 10E + 06$ e coeficiente de Poisson $\nu = 0.33$. A Figura 55 mostra as características do problema.

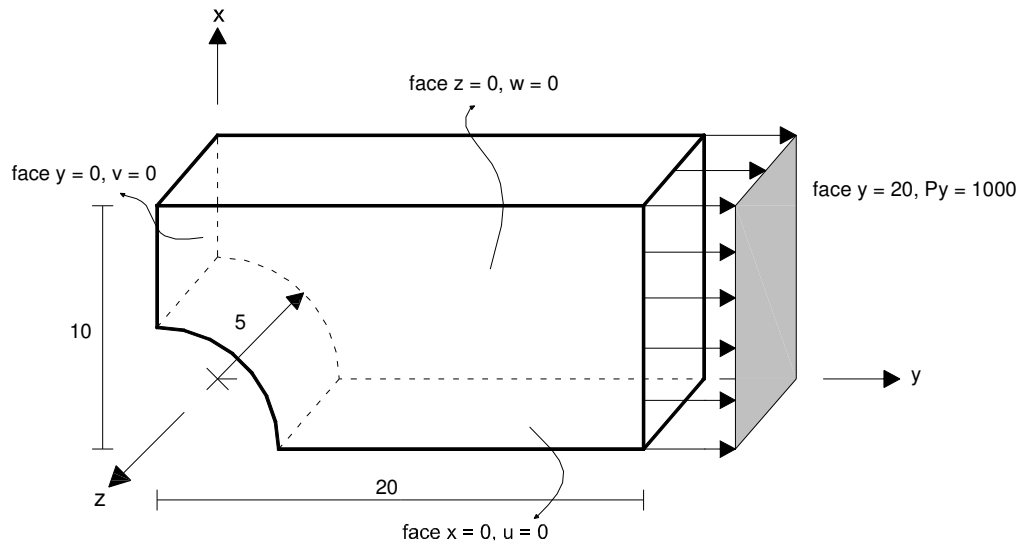


Figura 55 Placa com furo circular.

Para este exemplo foram necessários dois passos do processo adaptativo, um passo a menos do que obteve Cavalcante-Neto [13]. Porém, o número de elementos gerados pelo método proposto é bem maior, fato que pode ser observado na Tabela 7.

A Figura 56 mostra o aspecto das malhas para cada passo do processo adaptativo com os elementos quadráticos (TETRA10) e com as suas características, número de nós (NN), número de elementos (NE) e graus de liberdade (GDL).

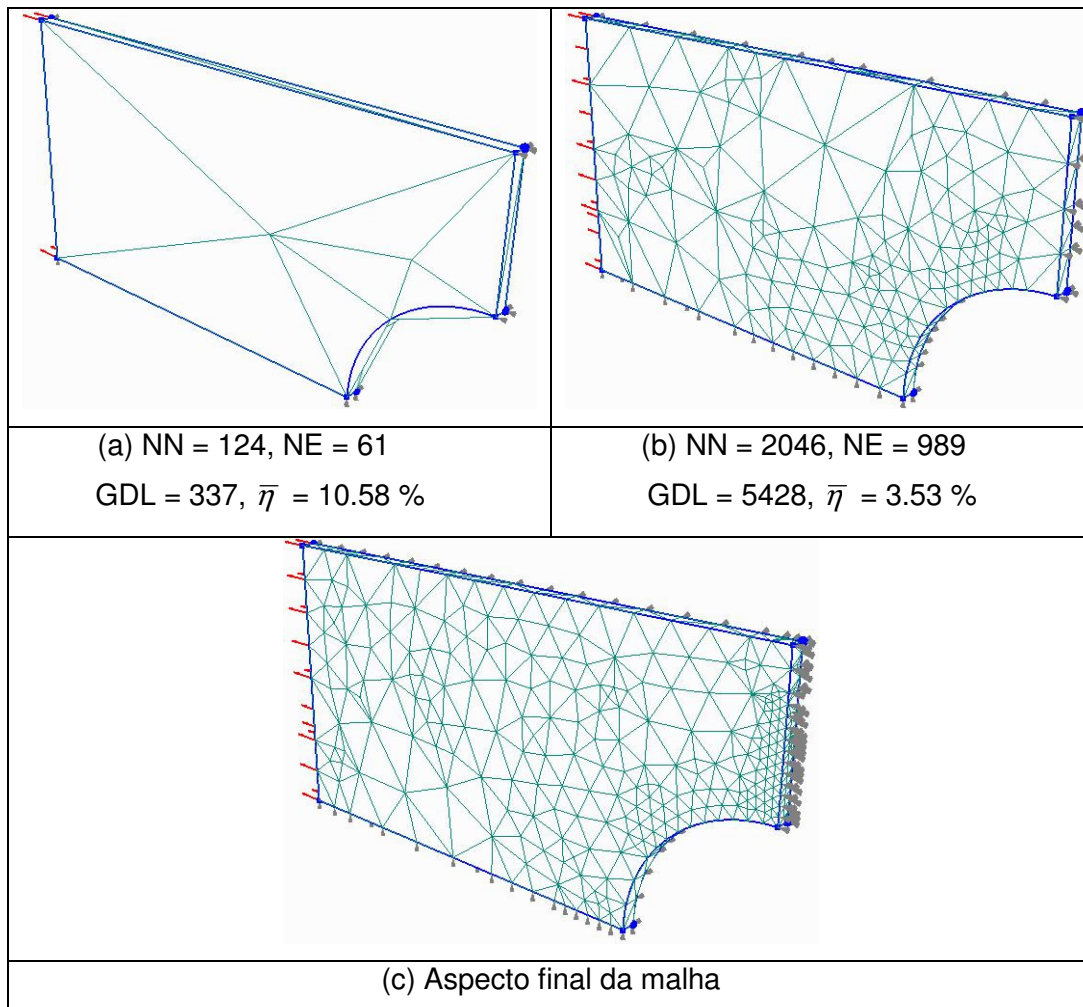


Figura 56 Exemplo 2 com TETRA10: malha inicial (a), malhas seguintes (b,c).

A Tabela 7 compara os resultados obtidos para o caso quadrático deste trabalho com os de Cavalcante-Neto [13].

Tabela 7 Comparação dos resultados do exemplo da placa 3D com furo circular.

	Cavalcante-Neto	Processo adaptativo proposto
1º	Elemento quadrático (TETRA10)	
NN	45	124
NE	12	61
GDL	54	337
$\bar{\eta}$	10.50 %	10.58 %
2º	Elemento quadrático (TETRA10)	
NN	313	2046
NE	160	989
GDL	684	5428
$\bar{\eta}$	6.50 %	3.53 %

Como as características iniciais não foram equivalentes, não se pode afirmar em qual caso se obteve a convergência de forma mais rápida. Até porque no primeiro passo a formulação de Cavalcante-Neto apresenta o valor de $\bar{\eta}$ menor, mas no passo seguinte já apresenta o valor de $\bar{\eta}$ maior.

5.2.3. Sólido com uma cavidade cilíndrica

O terceiro exemplo é um sólido com uma cavidade cilíndrica, submetida a uma pressão axial. Este problema também foi analisado por A Merrouche [1], adotando o valor de $\eta^* = 8\%$, considerando para a solução deste problema módulo de elasticidade $E = 1$, coeficiente de Poisson $\nu = 0.3$. A Figura 57 mostra as características do problema.

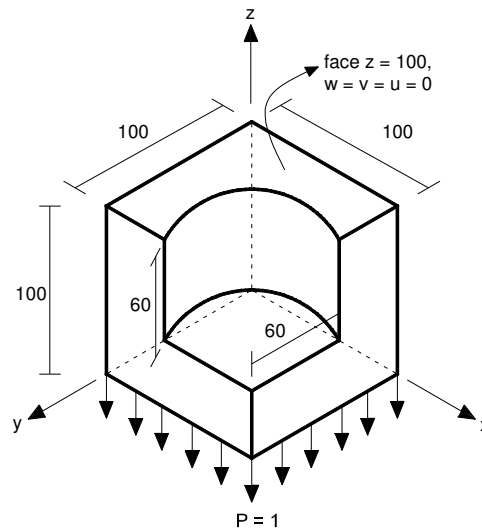


Figura 57 Sólido com cavidade cilíndrica.

Para este exemplo foram necessários dois passos do processo adaptativo. A Merrouche [1] obteve convergência após cinco passos.

A Figura 58 mostra o aspecto das malhas para cada passo do processo adaptativo com os elementos quadráticos (TETRA10) e com as suas características, número de nós (NN), número de elementos (NE) e graus de liberdade (GDL).

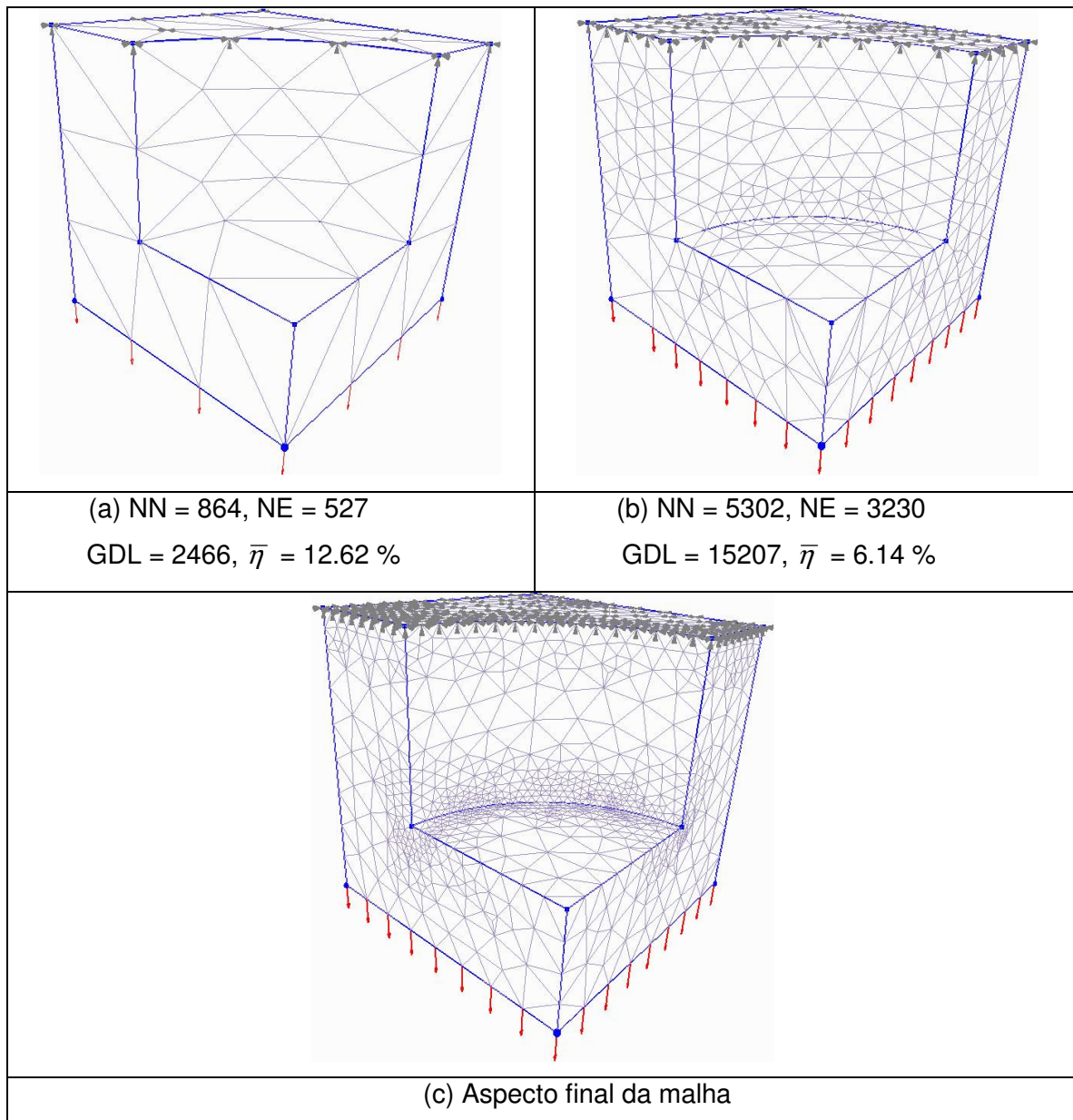


Figura 58 Exemplo 3 com TETRA10: malha inicial (a), malhas seguintes (b,c).

A Tabela 8 compara os resultados obtidos para o caso quadrático deste trabalho com os de A Merrouche [1].

Tabela 8 Comparação dos resultados do exemplo do sólido com uma cavidade cilíndrica.

	A Merrouche	Processo adaptativo proposto
1º	Elemento quadrático (TETRA10)	
NN	—	864
NE	140	527
GDL	—	2466
$\bar{\eta}$	23.50 %	12.62 %
2º	Elemento quadrático (TETRA10)	
NN	—	5302
NE	—	3230
GDL	—	15207
$\bar{\eta}$	—	6.14 %

No trabalho de A Merrouche [1], só são apresentados os números de elementos e o erros correntes $\bar{\eta}$ correspondentes ao passo inicial e final. Do último passo os valores obtidos foram os seguintes: $NE = 26643$ e $\bar{\eta} = 7.50\%$.

Como as características iniciais não foram equivalentes, não se pode afirmar em qual caso se obteve a convergência de forma mais rápida.

5.2.4. Eixo cilíndrico

O quarto exemplo é um eixo cilíndrico, uniformemente carregado em uma extremidade e preso em outra. Devido a simetria, apenas um quarto do problema foi modelado. Este problema também foi analisado por A Merrouche [1], adotando o valor de $\eta^* = 8\%$, considerando para a solução deste problema módulo de elasticidade $E = 1$, coeficiente de Poisson $\nu = 0.3$. A Figura 59 mostra as características do problema.

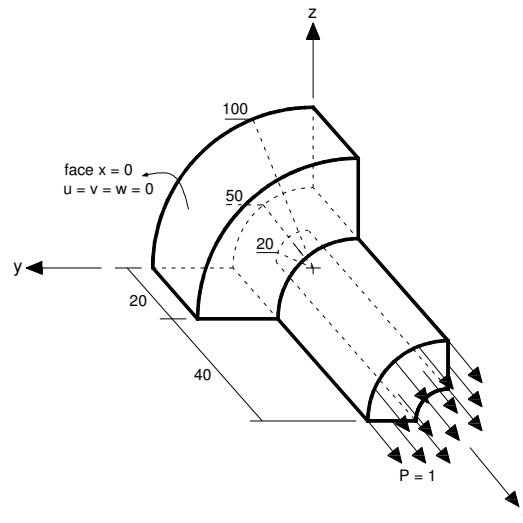


Figura 59 Eixo cilíndrico.

Para este exemplo foi necessário apenas um passo do processo adaptativo. A Merrouche [1] obteve convergência após quatro passos.

A Figura 60 mostra o aspecto das malhas para cada passo do processo adaptativo com os elementos quadráticos (TETRA10) e com as suas características, número de nós (NN), número de elementos (NE) e graus de liberdade (GDL).

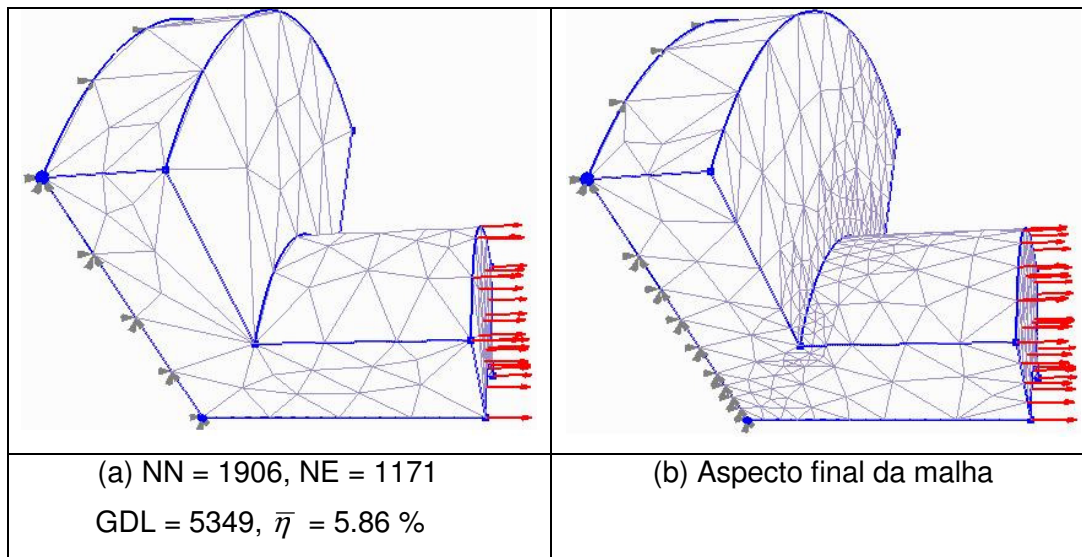


Figura 60 Exemplo 4 com TETRA10: malha inicial (a), malha seguinte (b).

A Tabela 9 compara os resultados obtidos para o caso quadrático deste trabalho com os resultados obtidos por de A Merrouche [1].

Tabela 9 Comparação dos resultados do exemplo do eixo cilíndrico.

	A Merrouche	Processo adaptativo proposto
1º	Elemento quadrático (TETRA10)	
NN	—	1906
NE	184	1171
GDL	—	5349
$\bar{\eta}$	21.50 %	5.86 %

No trabalho de A Merrouche [1], só são apresentados os números de elementos e o erros correntes $\bar{\eta}$ correspondentes ao passo inicial e final. Do último passo os valores obtidos foram os seguintes: $NE = 32985$ e $\bar{\eta} = 7.00\%$.

Como as características iniciais não foram equivalentes, não se pode afirmar em qual caso se obteve a convergência de forma mais rápida.

5.2.5. Pórtico misto 3D

O quinto exemplo é um pórtico misto, composto por três barras, sendo cada barra composta de um material diferente. O pórtico é uniformemente carregado em uma extremidade e preso em outra. Como o exemplo tem apenas a finalidade de mostrar a aplicabilidade do processo adaptativo para problemas com multiregiões, adotaram-se propriedades hipotéticas para os materiais. Sendo assim, atribui-se os seguintes valores para os módulos de elasticidade: $E_1 = 1$, $E_2 = 2$ e $E_3 = 3$ e os seguintes valores para os coeficientes de Poisson $\nu_1 = 0.2$, $\nu_2 = 0.3$ e $\nu_3 = 0.4$. As dimensões, as condições de contorno e atributos do problema são mostrados na Figura 61. Adotou-se o valor de $\eta^* = 5\%$.

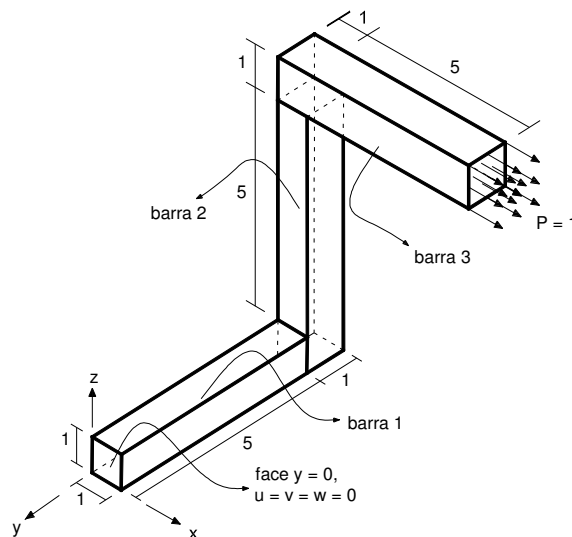


Figura 61 Pórtico misto 3D.

Para este exemplo foram necessários três passos do processo adaptativo.

A Figura 62 mostra o aspecto das malhas para cada passo do processo adaptativo com os elementos quadráticos (TETRA10) e com as suas características, número de nós (NN), número de elementos (NE) e graus de liberdade (GDL).

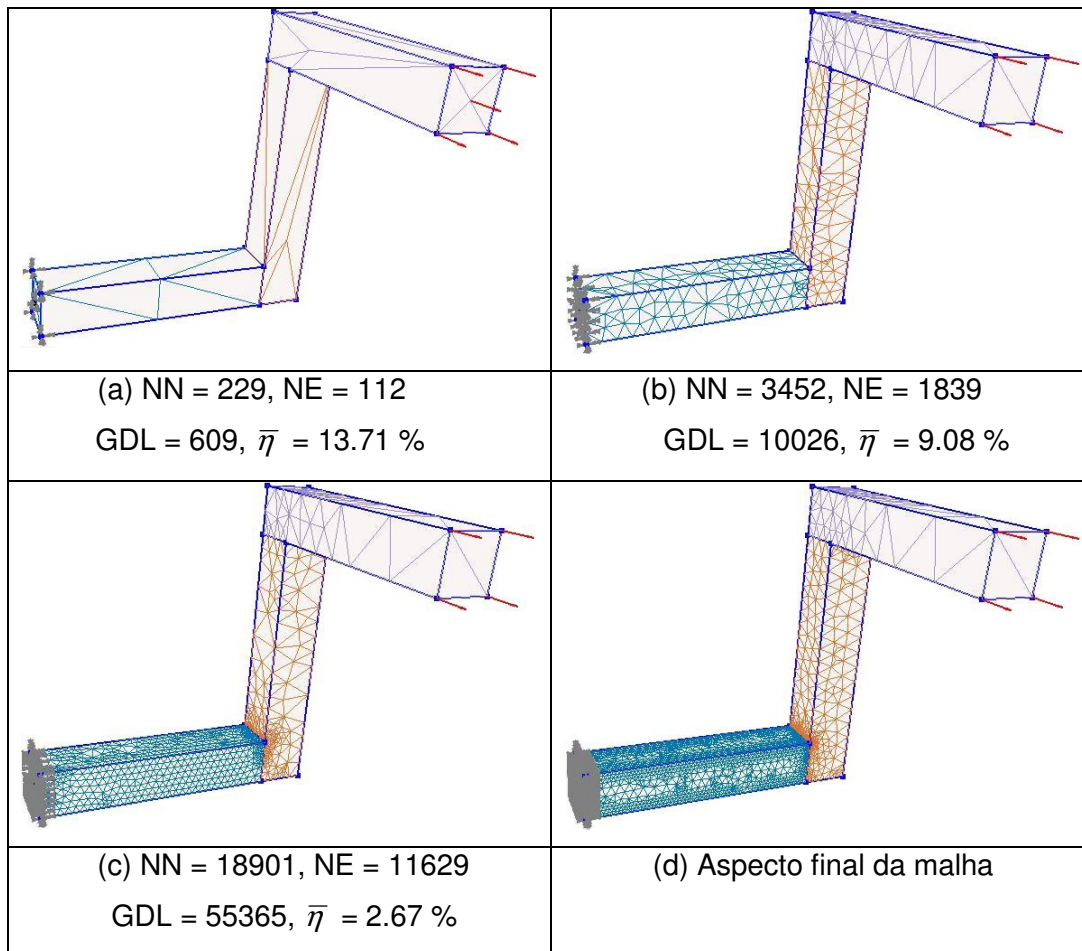


Figura 62 Exemplo 5 com TETRA10: malha inicial (a), malhas seguintes (b,c,d).

É possível observar na Figura 62 um gradativo refinamento nas regiões onde ocorrem elevadas variações de tensões, como por exemplo a barra 1 que está submetida à flexão, cisalhamento e torção é bem mais refinada que as outras. A barra 2 que está submetida à flexão e cisalhamento é pouco menos refinada que a barra 1 e a barra 3 que está submetida apenas a tração foi a menos refinada durante o processo adaptativo.

A Tabela 10 mostra os resultados obtidos em cada passo do processo adaptativo.

Tabela 10 Mostra os resultados obtidos para o exemplo do pórtico misto.

Processo adaptativo proposto	
1º	Elemento quadrático (TETRA10)
NN	229
NE	112
GDL	609
$\bar{\eta}$	13.71 %
2º	Elemento quadrático (TETRA10)
NN	3452
NE	1839
GDL	10026
$\bar{\eta}$	9.08 %
3º	Elemento quadrático (TETRA10)
NN	18901
NE	11629
GDL	55365
$\bar{\eta}$	2.67 %

6 CONCLUSÃO

Este trabalho abordou um novo processo adaptativo para geração de malhas de elementos finitos em 2D e 3D. Entende-se por processo adaptativo, todos os componentes necessários para a geração de novas malhas com a menor intervenção possível do usuário. Por isto foram apresentadas todas as principais partes integrantes do processo. Os modeladores geométricos usados foram o *MTOOL* [47] (2D) e o *MG* [48] (3D) com as respectivas estruturas de dados topológica *HED* [42] (2D) e *CGC* [14] (3D). Os modeladores foram desenvolvidos no Tecgraf/PUC-RIO (Grupo de Tecnologia em Computação Gráfica).

Foram mostradas as principais técnicas de estimativa de erro (em recuperação de *patches* e em resíduos), as técnicas de suavização (*Z2-HC* [62], *SPR* [63-66]), as estratégias adaptativas (de extensão h , r , p , ou combinação entre elas) e a parte referente à geração de malhas (estruturadas, não-estruturadas e híbridas). Como neste trabalho trabalhou-se com as malhas não-estruturadas, pois elas se adequam melhor a domínios arbitrários e são de fácil automação, apresentou-se as principais técnicas para este tipo de malha, que são: Decomposição Especial Recursiva, Triangulação de Delaunay e Avanço de Fronteira.

O processo adaptativo, deste trabalho, está fundamentado no refinamento independente de curvas, superfícies e sólidos, usando uma estimativa de erro *a-posteriori* baseada nas normas de erro, métodos de suavização do tipo *Z2-HC* [62] e *SPR* [63-66] e usando, de forma hierárquica, as técnicas de geração de malha de Decomposição Espacial Recursiva *binary-tree* (1D), *quadtrees* (2D) e *octrees* (3D) juntamente com Avanço de Fronteira. Foi tomado como referência o trabalho apresentado por Cavalcante-Neto [12,13], o qual não era totalmente automático, pois era necessário se discretizar manualmente as curvas próximas às regiões de grande concentrações de tensões (3D). Para o caso plano (2D) era necessária uma estrutura de dados para guardar relações de adjacência entre curvas e elementos, o que dificultava o processo de refinamento das curvas. A técnica de Decomposição Espacial Recursiva tinha apenas a função de gerar pontos no interior do domínio.

Neste trabalho, estendeu-se a função da técnica de Decomposição Espacial Recursiva, o que foi uma de suas principais contribuições, pois agora, além de garantir a boa transição entre os elementos da malha, esta estrutura guarda as informações da razão de erro, que são obtidas do resultado de análise de elementos finitos. Com esse novo enfoque, os tipos de estruturas usadas na Decomposição Espacial Recursiva são nomeadas de *octree* / *quadtree* global. Portanto, não é mais necessário obter as relações de adjacência entre elementos e curvas para refinar as curvas, bastando apenas consultar as informações guardadas na *octree* / *quadtree* global.

Outra contribuição deste trabalho consiste no fato de se fazer todo o processo adaptativo baseado no erro numérico, mas obedecendo à geometria do modelo através da utilização da adaptatividade geométrica de curvas e superfícies. Os parâmetros da adaptatividade geométrica para as curvas são definidos por três variáveis: tamanho mínimo (d_{MIN}) e tamanho máximo (d_{MAX}) de um segmento de subdivisão e uma tolerância angular (ang_{TOL}). Conforme o preenchimento destas variáveis, tem-se um nível diferente de refinamento para as curvas, porém os trechos de maior curvatura geralmente são mais refinados (vide exemplo 5.1.1). No caso 3D, também é feita a adaptatividade geométrica de superfícies, através do uso do algoritmo desenvolvido por Miranda *et al* [45]. A influência da adaptatividade geométrica no processo adaptativo é feita durante o refinamento da *octree* / *quadtree* global.

Através do uso da *octree* / *quadtree* global o processo adaptativo para modelos com multi-regiões (vide exemplos 5.1.2, 5.1.4 e 5.2.5) fica facilitado, pois as malhas de superfície e volumétricas (3D) são geradas individualmente para cada região, usando-se apenas a *octree* / *quadtree* global como parâmetro de controle do tamanho dos elementos que são gerados nas malhas no decorrer do processo adaptativo.

Acrescentado ao aspecto inovador da *octree* / *quadtree* global e ao uso da adaptatividade geométrica de curvas e superfícies para a geração de malhas, está o fato deste trabalho concretizar um objetivo, que há muito tempo é almejado pelo grupo de pesquisa no qual ele está inserido (Tecgraf/PUC-RIO), pois utilizando-se como base as técnicas desenvolvidas por Cavalcante-Neto [12,13] e o algoritmo de Miranda *et al* [45] automatizou-se o processo de geração de malhas adaptativas em 2D e 3D. De tal forma, que os únicos dados de entrada para se realizar tal processo são o(s) sólido(s) / superfície(s) com as suas respectivas malhas iniciais associadas. Uma vez feita a análise (*FEMOOP* [43]), o processo refina as regiões em que ocorrem altas concentrações de

tensões e “desrefina” regiões em que o refinamento existente é desnecessário. Para ilustrar este fato e a eficácia do método, apresentou-se no capítulo 5, vários exemplos conhecidos no meio técnico, onde foi possível verificar que todos os requisitos necessários para validar tal procedimento adaptativo foram atendidos.

6.1. Vantagens do Método

Como vantagens do processo adaptativo proposto pode-se citar.

- a criação da *octree* / *quadtree* global, facilitando o processo de refinamento das curvas do modelo e o controle na geração dos elementos da nova malha de elementos finitos;
- o fato de que durante o processo adaptativo se respeite à geometria do modelo, através da utilização da adaptatividade geométrica das curvas e superfícies;
- tratar problemas que envolvem multi-regiões de forma simples e direta;
- o processo não refina somente regiões em que ocorrem altas concentrações de tensões como também “desrefina” as regiões onde o refinamento existente é desnecessário.

6.2. Desvantagens do Método

Em relação às desvantagens do processo adaptativo proposto apresentado, pode-se citar:

- A necessidade de se utilizar duas vezes os algoritmos de adaptatividade geométrica para curvas e superfícies. No primeiro estágio, estes algoritmos são usados apenas para refinar a *octree* / *quadtree* global. No segundo estágio é que efetivamente se subdivide as curvas e gera-se as malhas de superfície.

6.3. Sugestões para Trabalhos Futuros

Como sugestões para trabalhos futuros em continuação ao que foi desenvolvido neste trabalho, apresentam-se:

- Analisar de forma mais consistente as vantagens e desvantagens de se associar o tamanho característico h de cada elemento finito aos seus vértices ou aos centros de gravidade dos mesmos;
- Procurar considerar a adaptatividade geométrica das superfícies no refinamento da *octree* global sem precisar utilizar os algoritmos de adaptatividade geométrica duas vezes;

Referências Bibliográficas

- [1] A Merrouche, Selman, A.; Nnopf-Lenoir, C., 1998. '3D Adaptive mesh refinement', Commun. Numer. Meth. Engng, 14, 397-407.
- [2] Acharjee, S.; Zabaras, N., 2006. 'The continuum sensitivity method for the computational design of three-dimensional deformation processes'. Comput. Methods Appl. Mech. Engng., 195, 6822-6842
- [3] Aymone, J. L. F. 'Mesh motion techniques for the ALE formulation in 3D large deformation problems'. Krieger, New York.
- [4] Babuska, I.; Rheinboldt, C., 1978. 'A-posteriori error estimates for the finite element method'. Int. J. Numer. Meth. Engng., 12, 1597-1615.
- [5] Babuska, I.; Rheinboldt, C., 1979. 'Adaptive approaches and reability estimates in finite element analysis'. Comput. Methods Appl. Mech. Engng., 17-18. 519-540.
- [6] Batista, V. H. F., 2005. 'Geração de malhas não estruturadas tetraédricas utilizando um método de avanço de fronteira'. Dissertação de Mestrado. COPPE / Universidade Federal do Rio de Janeiro. Brasil.
- [7] Beall, M. W.; Shephard, M. S., 1997. 'A general topology-based mesh data structure'. Int. J. Numer. Meth. Engng. , 40, 1573-1596.
- [8] Boroomand, B.; Zienkiewicz, O. C., 1997. 'Recovery by equilibrium in patches (REP)'. J. Numer. Meth. Engng. 40, 137-164.
- [9] Boussetta, R.; Coupez, T.; Forment, L., 2006. 'Adaptive remeshing based on a posteriori error estimation for forging simulation'. Comput. Methods Appl. Mech. Engng., 6626-6645.
- [10] Brauchli, H. J.; Oden, J. T., 1971. 'On the calculation of consistent stress distributions in finite elements applications'. Int. J. Numer. Meth. Engng. 3, 317-325.
- [11] Carey, G. F., 1997. 'Computational grids: Generation, Adaptation, and Solution Strategies'. 1st ed. Taylor & Francis. Washington, D.C., USA, 1997.

- [12] Cavalcante-Neto, J. B., 1994. 'Simulação auto-adaptativa baseada em enumeração espacial recursiva de modelos bidimensionais de elementos finitos'. Dissertação de Mestrado. Departamento de Engenharia Civil. PUC-RIO, Brasil.
- [13] Cavalcante-Neto, J. B., 1998. 'Geração de malha e estimativa de erro para modelos tridimensionais de elementos finitos com trincas'. Tese de Doutorado. Departamento de Engenharia Civil. PUC-RIO, Brasil.
- [14] Cavalcanti, P. R.; Carvalho, P. C. P.; Martha, L. F. C. R. ; 1997. 'Non-manifold Modeling: Na approach base don spatial subdivision'. Computer-Aided Design, v. 29, n.3, 209-220.
- [15] Cavendish, J. C.; Field, D. A.; Frey, W. H., 1985. 'An approach to automatic three-dimensional finite element mesh generation'. Int. J. Numer. Meth. Engng., v.21, n.2, 329-347.
- [16] Celes Filho, W.; Martha, L. F. C. R.; Gatass, M., 1991. 'na efficient data structure for solid finite element analysis pos-processor (inportuguese)'. XI Brazilian Congress of Mechanic Engineering, 105-108.
- [17] Chew, L. P.; 1989. 'Constrained Delaunay triangulations'. Algorithmica, v.4,n.2, 97-108.
- [18] Choi, W. -Y.; Son, I. -H., Im, Y. -T., 2004. 'Locally refined tetrahedral mesh generation based on advancing front technique with optimization and smoothing scheme'. Commun. Numer. Meth. Engng, 20, 681-688.
- [19] Canvas Draw.TECGRAF / PUC-RIO. '<http://www.tecgraf.puc-rio.br/cd>'.
- [20] De Veubeke, B. F., 'Displacement and equilibrium models in finite element method, in: O.C. Zienkiewicz, G.S. Holister (Eds.), Stress Analysis', John Wiley & Sons, Chichester, 1965, Cap.9, 1491-1506.
- [21] De Veubeke, B. F. 'Bending and stretching of plates. Special models for upper and lower bounds, in: Proc. First Conf. Matrix Methods in Structural Mechanics', vol. AFFDL-TR-66-80, Wright Patterson Air Force Base, Ohio, October 1966, 863-886.
- [22] De Veubeke, B. F; Zienkiewicz, O. C., 1967. 'Strain energy bounds in finite element analysis by slab analogy', J. Strain Anal. 2, 265-271.
- [23] Frey, P.; Sarter, B.; Gautherie, M., 1994. 'Fully automatic mesh generation for 3-D domains based upon voxels sets'. Int. J. Numer. Meth. Engng. , 37, 2735-2753.

- [24] Golias, N. A.; Dutton, R. W., 1997. 'Delaunay triangulation and 3D adaptive mesh generation'. *Finite Elements in Analysis and Design*, 25, 331-341.
- [25] Grosso, R.; Lüring, C.; Ertl, T., 1997. 'The Multilevel Finite Element Method for Adaptive Mesh Optimization and Visualization of Volume Data' Computer Graphics Group. Universität Erlangen-Nürnberg, Germany.
- [26] Hinton, E.; Campbell, J. S., 1974. 'Local and Global smoothing of discontinuous finite element functions using a least squares method'. *Int. J. Numer. Meth. Engng.*, 8, 461-480.
- [27] Hoffman, C. M., 1989. 'Geometric & Solid Modeling: An Introduction'. Purdue University, Indiana.
- [28] Hughes, T. J. R.; Cottrell, J. A.; Bazilevs, Y., 2005. 'Isogeometric analysis: CAD, finite elements, NURBS, exact geometric and mesh refinement'. *Comput. Methods Appl. Mech. Engrg.*, 194, 4135-4195.
- [29] I, Roberto; De Figueiredo, L. H.; C., Waldemar. 2006. 'LUA 5.1. Manual'
- [30] Kim, K. D.; Yang, D. Y.; Jeong, J. H., 2006. 'Adaptive refinement techniques based on tetrahedral and hexahedral grids for finite element analysis of mold filling in casting processes'. *Comput. Methods Appl. Mech. Engrg.*, 195, 6799-6821.
- [31] Lee, C. K.; Lo, S. H., 1997. 'Automatic adaptive refinement finite element procedure for 3D stress analysis'. *Finite Element in Analysis and Design*, 25, 135-166.
- [32] Lee, C. K.; Lo, S. H., 1999. 'A full 3D finite element analysis using adaptive refinement and PCG solver with back interpolation' *Comput. Methods Appl. Mech. Engrg.*, 170, 39-64.
- [33] Lee, C. K.; Wong, S. M.; Lie, S. T., 2001. 'On increasing the order and density of 3D finite element meshes'. *Commun. Numer. Meth. Engng*, 17, 55-68.
- [34] Lee, C. K., 2003. 'Automatic metric 3D surface mesh generation using subdivision surface geometrical model. Part 1: Construction of underlying geometrical model'. *Int. J. Numer. Meth. Engng.*, 56, 1593-1614.
- [35] Lee, C. K.; Xu, Q. X., 2005. 'A new automatic adaptive 3D solid mesh generation scheme for thin-walled structures'. *Int. J. Numer. Meth. Engng.*, 62, 1519-1558.
- [36] Lira, W. W. M., 1998. 'Um sistema integrado configurável para simulações em mecânica computacional'. Dissertação de Mestrado. Departamento de Engenharia Civil. PUC-RIO, Brasil.

- [37] Lira, W. W. M., 2002. 'Modelagem geométrica para elementos finitos usando multi-regiões e superfícies paramétricas'. Tese de Doutorado. Departamento de Engenharia Civil. PUC-RIO, Brasil.
- [38] Lo, S. H., 1985. 'A new mesh generation scheme for arbitrary planar domains'. Int. J. Numer. Meth. Engng. , 21, 1403-1426.
- [39] Lo, S. H., 1989. 'Delaunay triangulation of non-convex planar domains'. Int. J. Numer. Meth. Engng. , 28, 2695-2707.
- [40] Lo, S. H., 1991. 'Automatic mesh generation and adaptation by using contours'. Int. J. Numer. Meth. Engng. , 31, 689-707.
- [41] Mackerle, J., 2001. 'Error estimates and adaptive finite element methods'. Engineering Computations. v. 18, n. 5/6, 802-914.
- [42] Mäntylä, M., 1998. 'An introduction to solid modeling computer'. Science Press, Rockville, Maryland.
- [43] Martha, L. F. C. R.; Menezes, I. F. M.; Lages, E. N.; Parente, Jr.; Pitangueira, R. L. S.; 1996. 'Na OOP class organization for materially nonlinear finite element analysis'. Anais do XVII CILAMCE / Congresso Íbero-Latino Americano sobre Métodos Computacionais para Engenharia, 229-232.Veneza, Itália.
- [44] Miranda, A. C. O.; Cavalcante-Neto, J. B.; Martha, L. F. C. R., 1999. 'An algorithm for Two-dimensional Mesh Generation for Arbitrary Regions with Cracks'. Proceedings of SIBGRAPI'99, XII Brazilian Symposium on Computer Graphics, Image Proceedings and Vision, Campinas, Brazil, IEEE Computer Society, 29-38.
- [45] Miranda, A. C. O.; Martha, L. F. C. R., 2002. 'Mesh generation on high-curvature surfaces based on a background quadtree structure'. 11th International Meshing Roundtable Albuquerque, New Mexico: Sandia National Laboratories, v.11, 333-341.
- [46] Meyer, M.; Desbrun, M.; Schröder, P.; Barr, A. H., 2002. 'Discrete differential geometric operators for triangulated 2 - Manifold'. In visualization and mathematics III, H. -C. Hege and K. Polthier eds, 35-57.
- [47] MTool – Bidimensional Mesh Tool – Manual do usuário, PUC-RIO, 1992.
- [48] MG – Mesh Generation, manual do usuário, versão 3.0, PUC-RIO, 1996.

- [49] Peixoto, A.; Velho, L., 2003. 'Extração de malhas adaptativas com operações de simplificação e refinamento'. Tese de Doutorado. Departamento de Informática. PUC-RIO, Brasil.
- [50] Peraire, J.; Vahdati, M.; Morgan, K., *et al*, 1987. 'Adaptive remeshing for compressible flow computations'. *Journal of Computational Physics*, 72, 449-466.
- [51] Prazeres, P. G. C., 2005. 'Desenvolvimento de elementos finitos híbridos para análise de problemas dinêmicos usando superposição modal avançada'. Dissertação de Mestrado, Departamento de Engenharia Civil. PUC-RIO, Brasil.
- [52] Richardson, L. F., 1910. 'The approximate arithmetical solution by finite differences of physical problems. *Trans. Roy. Soc. (London)*', 210, 307-357.
- [53] Schreiner, J.; Asirvatham, A.; Praun, E.; Hoppe, H.; 2004. 'Inter_surface Mapping'. *ACM transactions on Graphics / SIGGRAPH*.
- [54] Schreiner, J.; Scheideger, C. E.; Silva, C. T., 2006. 'High-Quality Extraction of Isosurfaces from Regular and Irregular Grids'. *IEEE Transactions on Visualization and Computer Graphics*, v.12, n.5.
- [55] Schreiner, J.; Scheideger, C. E.; Fleishman, S.; Silva, C. T., 2006. 'Direct (Re)Meshing for efficient Surface Processing'. *Computer Graphics forum / Eurographics*, v.25, n.3.
- [56] Souza, D. L., 2004. 'Geração adaptativa de malha baseada em erro de curvatura'. Dissertação de Mestrado, Departamento de Computação. Universidade Federal do Ceará. Brasil.
- [57] Sydenstricker, R. M.; Coutinho, A. L. G. A.; Martins, M. A. D.; Landau, L.; Alves, J. L. D., 2004. 'Uma estimativa de erro a-posteriori para a análise mecânica de estruturas de materiais heterogêneos'. VI Simpósio Mineiro de mecânica computacional.
- [58] Thompson, J. F.; Weatherill, N. P., 1992. 'Structured and Unstructured grid generation'. *Critical Reviews in Biomedical Engineering*, v.20, n1-2, 73-120.
- [59] Tristano, J. R.; Owen, S. J.; Canann, S. A., 1998. 'Advancing front surface mesh generation in parametric space using Riemannian Surface Definition'. 7th International Meshing Roundtable Proceedings, 429-445.
- [60] Yerry, M. A.; Shephard, M. S., 1984. 'Automatic three dimensional mesh generation by the modified-octree technique'. *Int. J. Numer. Meth. Engng.* , v.20, n.11, 1965-1990.

- [61] Yerry, M. A.; Shephard, M. S., 1985. 'Automatic Mesh Generation for three-dimensional solids'. *Computer & Structures*, v.20, n.1-3, 31-39.
- [62] Zienkiewicz, O. C.; Zhu, J. Z., 1987. 'A simple error estimator and adaptive procedure for practical engineering analysis'. *Int. J. Numer. Meth. Engng.* , 24, 337-357.
- [63] Zienkiewicz, O. C.; Zhu, J. Z., 1992. 'The Superconvergent patch recovery (*SPR*) and adaptive finite element refinement'. *Comput. Methods Appl. Mech. Engng.*, 101, 207-224.
- [64] Zienkiewicz, O. C.; Zhu, J. Z., 1992. 'The Superconvergent patch recovery (*SPR*) and a posteriori error estimates. Part. 1: The recovery technique'. *Int. J. Numer. Meth. Engng.* , 33, 1331-1364.
- [65] Zienkiewicz, O. C.; Zhu, J. Z., 1992. 'The Superconvergent patch recovery (*SPR*) and a posteriori error estimates. Part. 2: Error estimates and adaptivity'. *Int. J. Numer. Meth. Engng.* , 33, 1365-1382.
- [66] Zienkiewicz, O. C.; Taylor, R. L., 2000. 'The finite element method'. Fifth ed. Vols1 e 2, Butterworth_Heinemann.
- [67] Zienkiewicz, O. C., 2006. 'The background of error estimation and adaptivity in finite element computations'. *Comput. Methods Appl. Mech. Engng.* 195, 207-213.
- [68] Zuo, J.; Deng, X.; Sutton, M. A., 2005. 'Advances in tetrahedral mesh generation for modeling of three-dimensional regions with complex, curvilinear crack shapes'. *Int. J. Numer. Meth. Engng.* , 63, 256-275.