TWO AND THREE DIMENSIONAL GEOMETRY BASED ADAPTIVE FINITE ELEMENT ANALYSIS

Rafael A. de Sousa¹, Antonio C. O. Miranda¹, William W.M. Lira², Joaquim B. Cavalcante-Neto³, and Luiz F. Martha¹*

1: Department of Civil Engineering and Tecgraf – Computer Graphics Technology Group, PUC-Rio – Pontifical Catholic University of Rio de Janeiro Rua Marquês de São Vicente, 225 – Rio de Janeiro, RJ – Brazil – 22453-900 e-mail: {rflsousa,amiranda,lfm}@tecgraf.puc-rio.br web: http://www.tecgraf.puc-rio.br

2: Department of Structural Engineering Alagoas Federal University Campus A. C. Simões, BR 104, km 97, Tabuleiro do Martins – Maceió, AL – Brazil – 57072-970 e-mail: william@lccv.ufal.br web: http://www.lccv.ufal.br

> 3: Computer Science Department Ceará Federal University Fortaleza, CE – Brazil e-mail: joaquimb@lia.ufc.br web: http://www.lia.ufc.br

Keywords: Finite Element Method, Mesh Generation, Adaptive Analysis, Geometric Modeling

Resumo. This work presents a methodology for adaptive generation of 2D and 3D finiteelement 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 discratization 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 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.

1. INTRODUÇÃO

Em simulações numéricas usando o Método dos Elementos Finitos (MEF) dois importantes aspectos que devem ser considerados são a geração automática da malha de elementos finitos do modelo e a definição do nível de refinamento associado a esta malha. Este refinamento precisa ser tal que a densidade dos elementos gerados varie conforme o gradiente da solução obtida. Nesse contexto, técnicas adaptativas são procedimentos usuais que ajudam a definir estes aspectos.

Algumas estratégias têm sido propostas para determinar estes aspectos de forma eficiente. Em um trabalho apresentado anteriormente por Cavalcante-Neto [1], onde é proposta uma estratégia auto-adaptativa para problemas bidimensionais, envolvendo a geração de malhas e métodos adaptativos. Outros trabalhos têm sido propostos considerando o mesmo problema através de diferentes enfoques, como, por exemplo, o grupo de Mark Shephard [9] da Rensselaer Polytechnic Institute, USA. Entretanto, a análise adaptativa 3D é mais complicada, porque necessita gerar malhas volumétricas através de estimadores de erro que considerem o efeito tridimensional, o que é uma tarefa complicada e aberta à novas pesquisas. Portanto, pesquisas têm sido feitas para desenvolver métodos e algoritmos capazes de resolver o problema para o caso 3D. Cavalcante-Neto propôs em outro trabalho [4] uma técnica de geração de malhas volumétricas compostas por tetraédros para regiões de domínios arbitrários. Combinando esta técnica com a implementação de estimadores de erro em 3D foi desenvolvido um protótipo para a geração adaptativa de malhas tridimensionais. Mas nesse trabalho o processo adaptativo não está sendo feito de forma automática, pois as malhas eram geradas independentemente do estimador de erro, necessitando dessa forma, gerar as novas malhas através da intervenção do usuário. Além disso, não se tratava problemas com multiregiões de forma eficiente e superfícies paramétricas não eram consideradas (o que é uma tarefa essencial em diversos tipos de simulações).

Mackerle [6] divulgou uma exaustiva coleta de artigos publicados sobre estimativa de erro e adaptatividade em elementos finitos, desde 1990 até 2000. Nesse 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 referente ao processo adaptativo.

O presente trabalho tem o propósito de apresentar um novo processo adaptativo para a geração auto-adaptativa de malhas de elementos finitos em 2D e 3D, usando um modelador geométrico considerando multi-regiões e superfícies paramétricas. O processo adaptativo é baseado no refinamento independente de curvas, superfícies e sólidos. Entende-se por processo adaptativo todos os itens necessários para se refinar uma malha de elementos finitos, conforme a estimativa de erro adotada com a menor interferência possível do usuário. Dentro desse contexto estão a técnica de estimativa de erro adotada (resíduos, normas de energia, etc), a estratégia adaptativa (h, r, p, h-p, etc) e técnicas de geração de malhas (decomposição espacial recursiva, avanço de fronteira, etc).

Inicialmente, todas as curvas do modelo são refinadas no seu espaço paramétrico, através de considerações geométricas e de erro numérico, usando uma técnica de

decomposição binária de curva, com base em uma estrutura de dados do tipo *binary-tree*. A discretização das curvas é usada como parâmetro de entrada para o refinamento das superfícies adjacentes. A discretização das superfícies é também construída no espaço paramétrico e aplica o refinamento de uma *quadtree* auxiliar, que é usada para guardar as distorções métricas entre os espaços paramétrico e cartesiano. Essa *quadtree* é utilizada pela técnica de avanço de fronteira para a geração de malhas triangulares não estruturadas na superficie [7]. Estas malhas de superficies são usadas como dados de entrada para o refinamento dos domínios volumétricos adjacentes. A discretização do volume combina uma técnica de avanço de fronteira ([4]) para gerar malhas de elementos finitos não-estruturadas com elementos tetraédricos.

Em todos os estágios do processo adaptativo proposto, o refinamento das curvas, das malhas de superficie e das malhas sólidas são baseados nos criterios geométricos das curvas e superficies na estimativa de erro numérico da malha existente do passo anterior do processo adaptativo.

Tanto as informações de erro numérico quanto as baseadas em criterios geométricos são guardadas em uma única estrutura de dados que engloba todo o modelo. Estrutura esta chamada de *octree* global, que facilita os procedimentos de busca.

2. PROCESSO ADAPTATIVO PROPOSTO

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"), são o Z2-HC [10], baseado na média nodal, e o SPR [11-12], baseado em recuperação de *patches*. A técnica de geração de malhas é baseada em Decomposição Espacial Recursiva e Avanço de Fronteira, usando os algoritmos desenvolvidos por Cavalcante-Neto [4] e Miranda e Martha [7], 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, estas estruturas agora são responsáveis pelo refinamento das curvas e superfícies, além de controlar o tamanho dos elementos que são gerados no domínio. A figura 1 mostra o fluxograma do processo adaptativo apresentado neste trabalho.

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.



Figura 1 Fluxograma do novo processo adaptativo (2D e 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 [4] e Miranda [7] 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).

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 para gerar as malhas de superfície no caso 3D e das malhas de domínio no caso 2D. No caso 3D, a geração da malha sólida 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.

Nas seções posteriores será feita referência apenas para o caso 3D, pois em vários passos do processo a implementação em 2D é análoga a de 3D. O caso 2D apenas será referenciado quando for necesario, quando a analogia entre os dois casos não for válida.

2.1. Construção da octree global baseada no erro numérico

O primeiro cubo da *octree* global é incializado com a menor dimensão que é capaz de envolver completamente o modelo

Após se inicializar a *octree* global, calcula-se, de acordo com o volume de cada elemento e com a razão de erro associada a ele, o tamanho característico que será associado ao seu centro geométrico. Este tamanho característico será usado no refinamento da *octree* global.

Isto é feito sudvidindo-se recursivamente a célula que contém o centro geométrico de cada elemento até que seu tamanho seja menor ou igual ao tamanho característico do elemento

2.2. Refinamento da *octree* global baseado na geometria das curvas

Três parâmetros devem ser adotados para se usar a adaptatividade geométrica, de tal forma que os trechos das curvas que apresentarem maior curvatura serão mais refinados. Este fato também se torna muito importante para adaptatividade baseada em erro numérico, pois, em geral, são justamente estes trechos que normalmente apresentam grande gradiente da resposta.

Desses parâmetros, dois são dependentes do comprimento total da curva: o comprimento máximo (d_{MAX}) e o comprimento mínimo (d_{MIN}) de uma subdivisão da

curva. O outro parâmetro é angular. Ele serve para evitar que trechos de curvas que tenham curvaturas acentuadas não sejam discretizadas, pois para o caso em que dois pontos estejam em lados contrários à curvatura o parâmetro d_{MIN} pode falhar, fazendo com que estes trechos não sejam refinados. A figura 2 abaixo ilustra a importancia da adaptatividade geométrica na simulação de problemas de engenharia.



Figura 2. (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.

2.3. Refinamento da *octree* global baseado na adaptatividade geométrica das superficies

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. Desta 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 [7], que foi uma extensão da técnica de geração de malhas proposta por Cavalcante-Neto [2,3].

Nesse algoritmo a malha é gerada no espaço paramétrico da superfície, mas ângulos e distâncias são calculados no espaço Cartesiano 3D. Dessa forma, os elementos finitos podem ser distorcidos no espaço paramétrico. Mas quando são mapeados ao espaço Cartesiano devem apresentar boa razão de aspecto. O algoritmo usa um mapeamento de métrica desenvolvido por Tristano [8] para obter distâncias e ângulos corretos entre os dois espaços.

Este trabalho mantém a idéia de Miranda e Martha [7] 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 esta 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.

A figura 3 ilustra a importância de se considerar a curvatura da superfície na geração de malhas. A figura 3.a mostra uma malha sem considerar as curvaturas das curvas de bordo e da superfície. A figura 3.b mostra a *quadtree* auxiliar construída para realizar a adaptatividade geométrica da superfície e a figura 3.c mostra a malha quando se considera a curvatura.



Figura 3. 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 [5]).

2.4. Refinamento da octree global para forçar mínima disparidade de tamanho

Em seguida a *octree* 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 4 mostra um exemplo bidimensional com este ajuste, onde as células marcadas em "x" serão refinadas conforme este critério.



Figura 4. (a) *quadtree* inicial sem refinamento que força a diferença mínima entre as células, (b) *quadtree* final com refinamento adicional que força a diferença mínima entre as células.

A figura 4, 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.

2.5. 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 da discretização das curvas.



Figura 5. Refinamento de uma curva reta usando a Decomposição Espacial Recursiva binária (binary-tree).

A decomposição recursiva binária é descrita a seguir. Caso o tamanho da curva seja maior que o tamanho da célula da *octree* global 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 5 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.

2.6. 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 do algoritmo de geração de malhas volumétricas que é descrito na próxima seção. Por isto, 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 disto, 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 para a criação da *quadtree* auxiliar é análogo ao que foi descrito para a discretização das curvas: o tamanho das células da *quadtree* auxiliar é ditado pela *octree* global, que já foi refinada considerando critérios de adaptatividade geométrica e numérica.

Uma vez gerada a *quadtree* auxiliar, o algoritmo de geração de malhas de superfície segue o que está descrito no trabalho de Miranda e Martha [7] que, como foi dito, é uma versão simplificada do algoritmo de geração 3D [4].

2.7. Algoritmo de geração de malhas volumétricas

O algoritmo utilizado está descrito em detalhes no artigo de Cavalcante-Neto *et al* [4] e tem três requisítos básicos.

Primeiramente, o algoritmo deve produzir elementos de boa forma, evitando elementos com apectos 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.

2.8. Considerações relevantes do processo adaptativo

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 e Martha [7]. 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.

3. EXEMPLOS

Para validar o processo adaptativo proposto são apresentados três exemplos. Um bastante conhecido e com solução analítica conhecida, outro analisado por A. Merrouche [1] e o terceiro foi idealizado pelos autores deste trabalho. No primeiro e no terceiro exemplo os

erros correntes de cada malha (η^*), que serve como critério de parada, foram adotados iguais 5%. No segundo exemplo usou-se, conforme A. Merrouche $\eta^* = 8\%$. Só foram utilizados elementos quadráticos. O processo adaptativo é interrompido quando o erro relativo corrente for menor ou igual que o erro relativo máximo ($\overline{\eta} \le \eta^*$).

É importante enfatizar que as comparações feitas nesse trabalho 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. 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 inciais tenham sido compatíveis.

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.

3.1. Viga curta em balanço

O primeiro exemplo é uma viga curta em balanço, fixa em uma extremidade e submetida a uma pressão tangente distribuída vertical na sua extremidade livre. Este problema também foi analisado por Cavalcante-Neto [4]. Adotou-se o valor de $\eta^* = 5\%$, considerando para a solução deste problema módulo de elasticidade E = 1 e coeficiente de Poisson v = 0.3. A figura 6 mostra as características do problema.



Figura 6. Viga curta em balanço.

A figura 7 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 7. Exemplo 1 com TETRA10: malha inicial (a), malhas seguintes (b,c).

A tabela 1 compara os resultados obtidos para o caso quadrático deste trabalho com os de Cavalcante-Neto.

rabela 1. comparação dos resultados do exemplo da viga curta em balanço 5D.			
	Cavalcante-Neto	Processo adaptativo proposto	
1°	Elemento quadrático (TETRA10)		
NN	386	599	
NE	177	336	
GDL	807	1604	
$\overline{\eta}$	7.90 %	7.48 %	
2°	Elemento quadrático (TETRA10)		
NN	1122	3921	
NE	609	2240	
GDL	2625	10304	
$\overline{\eta}$	4.20 %	3.18 %	

Tabela 1. Comparação dos resultados do exemplo da viga curta em balanço 3D.

Na tabela 1 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 $\overline{\eta}$ obtidos a cada passo são menores do que os obtidos por Cavalcante-Neto [4]. 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.

3.2. Sólido com cavidade cilíndrica

O segundo 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 v = 0.3. A figura 8 mostra as características do problema.



Figura 8. Sólido com cavidade cilíndrica.

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

A figura 9 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 9. Exemplo 2 com TETRA10: malha inicial (a), malhas seguintes (b,c).

Este exemplo mostra bem que, apesar da segunda malha do processo adaptativo (figura 9.b) já atender o critério de convergência adotado, a malha seguinte (figura 9.c) é bem mais refinada.

A tabela 2 compara os resultados obtidos para o caso quadrático deste trabalho com os de A Merrouche.

Tuberu 2. Computição dos resultados do exemplo do sondo com una cuvidade enhanea.		
	A Merrouche	Processo adaptativo proposto
1°	Elemento quadrático (TETRA10)	
NN	_	864
NE	140	527
GDL	—	2466
$\overline{\eta}$	23.50 %	12.62 %
2°	Elemento quadrático (TETRA10)	
NN	_	5302
NE	_	3230
GDL	_	15207
$\overline{\eta}$	_	6.14 %

Tabela 2. Comparação dos resultados do exemplo do sólido com uma cavidade cilíndrica.

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

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

3.3. Pórtico misto 3D

O terceiro 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 paras 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 $v_1 = 0.2$, $v_2 = 0.3$ e $v_3 = 0.4$. As dimensões, as condições de contorno e atributos do problema são mostrados na figura 10. Adotou-se o valor de $\eta^* = 5\%$.



Figura 10. Pórtico misto. 3D

Para este exemplo foram necessários três passos do processo adaptativo.

A figura 11 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 11. Exemplo 3 com TETRA10: malha inicial (a), malhas seguintes (b,c,d).

4. CONCLUSÕES

Neste trabalho utilizou-se as malhas não-estruturadas, pois elas se adequam melhor a domínios arbitrários.

O processo adaptativo desse trabalho está fundamentado no refinamento independente de curvas, superfícies e sólidos, usando uma estimava de erro *a-posteriori* baseada em normas de erro numérico, métodos de suavização do tipo Z2-HC [10] e SPR [11-12] e usando, de forma hierárquica, técnicas de geração de malha de Avanço de Fronteira com uma função de espaçamento nodal baseado em Decomposição Espacial Recursiva *binary-tree* (1D), *quadtree* (2D) e *octree* (3D). Foi tomado como referência o trabalho apresentado por Cavalcante-Neto [2-4]

Foi estendida a função da técnica de Decomposição Espacial Recursiva (*quadtree* ou *octree*) para o refinamento do domínio, o que foi uma das principais contribuições deste trabalho. 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. Além disso, diferentemente dos trabalhos anteriores, a *quadtree / octree* é global e única para todo o modelo. Portanto, para refinar as curvas, basta 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.

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

Como resultado final, este trabalho sintetizou vários trabalhos anteriores em um produto final, onde 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 modelo geométrico, seus atributos e as suas respectivas malhas iniciais associadas. Uma vez feita a análise de elementos finitos de cada passo do processo, o algoritmo refina as regiões em que ocorrem altos gradientes de desposta e "desrefina" regiões em que o refinamento existente é desnecessário. Para ilustrar este fato e a eficácia do método, apresentou-se três exemplos, onde foi possível verificar que todos os requisitos necessários para validar tal procedimento adaptativo foram atendidos.

REFERÊNCIAS

- [1] A Merrouche, Selman, A. e Nnopf-Lenoir, C.3D Adapative mesh refinement, Commun. Numer. Meth. Engng, 14, 397-407 (1998).
- [2] Paulino, G. H.; Menezes, I.F.M.; Cavalcante-Neto, J.B.; Martha, L.F.C.R. A Methodology for Adaptive Finite Element Analysis: Towards an Integrated Computational Environment. Computational Mechanic. v. 23, n. 5/6, 361-388 (1999).
- [3] Cavalcante-Neto, J.B.; Wawrzynek, P.A.; Carvalho, M.T.M.; Martha, L.F.; Ingraffea, A.R. *An Algorithm for Three-dimensional Mesh Generation for Arbitrary Regions with Cracks*. Engineering with Computers. vol. 17, no. 1, 75-91 (2001).
- [4] Cavalcante-Neto, J. B.. *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 (1998).
- [5] Lira, W. W. M. *Modelagem geométrica para elementos finitos usando multi-regiões e superfícies paramétricas*. Tese de Doutorado. Departamento de Engenharia Civil. PUC-RIO, Brasil (2002).
- [6] Mackerle, J. *Error estimates and adaptive finite element methods*. Engineering Computations. v. 18, n. 5/6, 802-914 (2001).
- [7] Miranda, A. C. O.; Martha, L. F. C. R. *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 (2002).

- [8] Tristano, J. R.; Owen, S. J.; Canann, S. A. Advancing front surface mesh generation in parametric space using Riemannian Surface Definition. 7th International Meshing Roundtable Proceedings, 429-445 (1998).
- [9] Shephard, M., S.; Yerry, M.A. Automatic Three-Dimensional Mesh Generation by Modified-Octree Technique. Int. J. Numer. Meth. Engng., 20, 1965-1990 (1984).
- [10] Zienkiewicz, O. C.; Zhu, J. Z. A simple error estimator and adaptive procedure for practical engineering analysis. Int. J. Numer. Meth. Engng., 24, 337-357 (1987).
- [11] Zienkiewicz, O. C.; Zhu, J. Z., 1992. *The Superconvergent patch recovery (SPR) and adaptive finite element refinement*. Comput. Methods Appl. Mech. Engrg., 101, 207-224.
- [12] Zienkiewicz, O. C.; Zhu, J. Z.. *The Superconvergent patch recovery (SPR) and a posteriori error estimates. Part. 1: The recovery technique.* Int. J. Numer. Meth. Engng., 33, 1331-1364 (1992).
- [13] Zienkiewicz, O. C.; Zhu, J. Z.. *The Superconvergent patch recovery (SPR) and a posteriori error estimates. Part. 2: Error estimates and adaptivity.* Int. J. Numer. Meth. Engng., 33, 1365-1382 (1992).
- [14] Zienkiewicz, O. C.; Taylor, R. L.. *The finite element method*. Fifth ed. Vols1 e 2, Butterworth_Heinemann (2000).
- [15] Zienkiewicz, O. C. *The background of error estimation and adaptivity in finite element computations*. Comput. Methods Appl. Mech. Engrg. 195, 207-213 (2006).