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 *Z*2-*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.

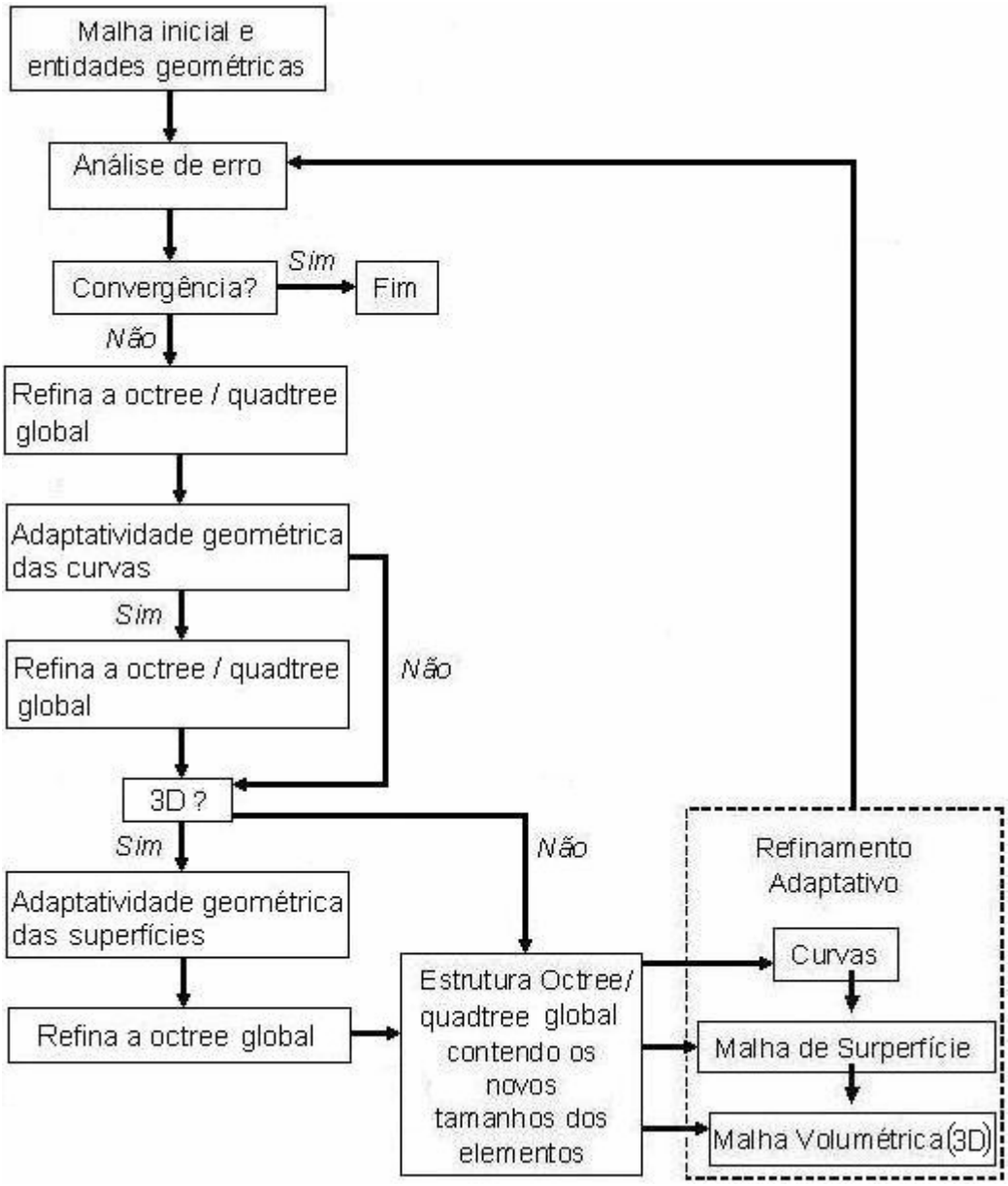


Figura 11 Fluxograma do novo processo adaptativo (2D e 3D).

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 incializado 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 (eqüilátero) 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 / eqüiláteros 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 eqüilátero e o seu lado.

$$L = \sqrt{\frac{4A_{REAL}}{\sqrt{3}}} \tag{33}$$

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.

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

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 testanto 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:

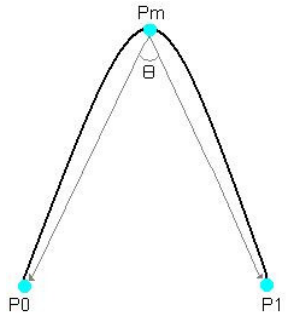


Figura 12 Curva a ser subdividida segundo os critérios geométricos.

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.

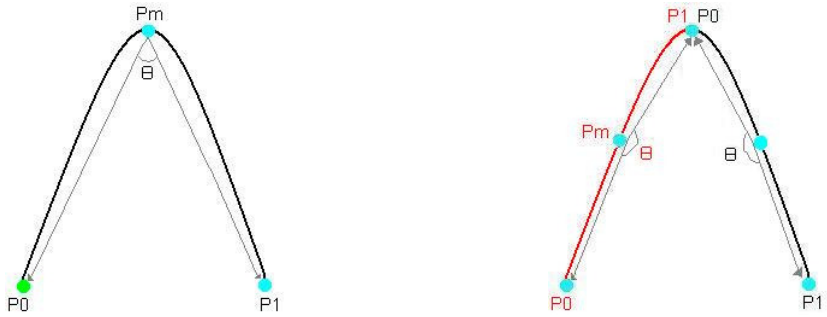


Figura 13 (a) Ponto P0 é inserido (b) Inseri-se uma nova subdivisão, fazendo-se P1 = Pm e Pm sendo a metade do novo trecho.

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 P1 = Pm para o primeiro trecho (em vermelho) e P0 = Pm para o segundo trecho (em preto), e recalculando-se os pontos intermedirários 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 Pm) e cujos vértices finais encontram-se sobre os pontos inicial e final do trecho (P0 e P1). Calcula-se, então, o módulo da diferença entre o valor deste ângulo e 180º. Se esta diferença for menor que a tolerância angular pré-estabelecida, o ponto P0 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 subidivido 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.

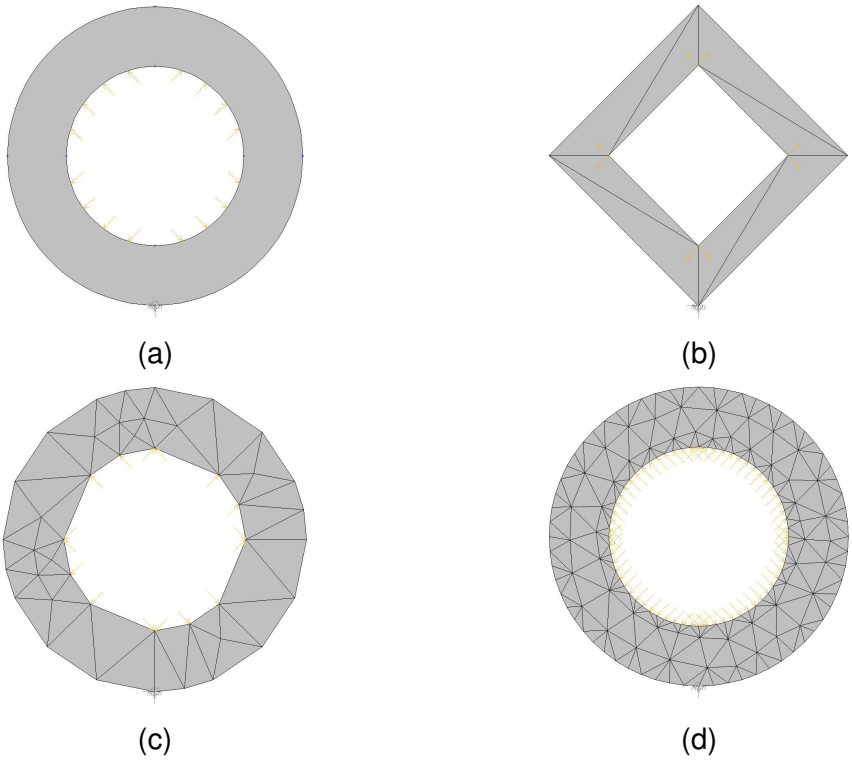


Figura 14 (a) Tubo com pressão interna, (b) Malha grosseira do modelo do tubo, (c) resultado do 1º passo do processo adaptativo usando apenas critérios baseados na estimativa de erro e (d) resultado do 1º passo do processo adaptativo usando os critérios de estimativa de erro e os critérios geométricos.

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).

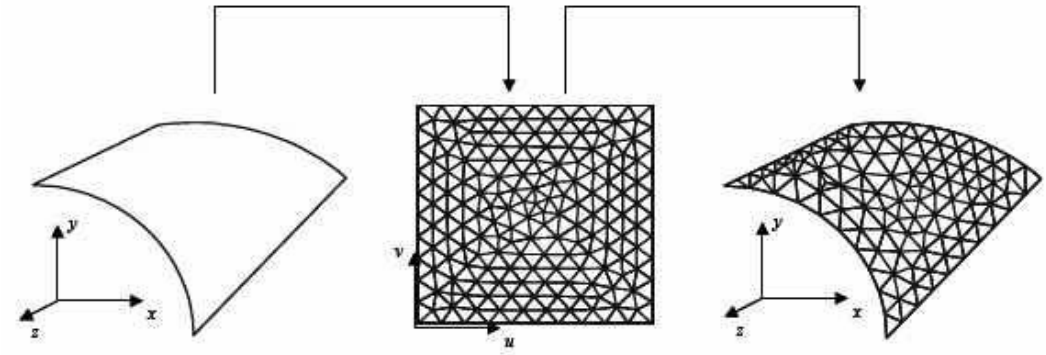


Figura 15 Processo de geração de malhas superficiais paramétricas em 3D (Lira [37]).

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.

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

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

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

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 30º.

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

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.

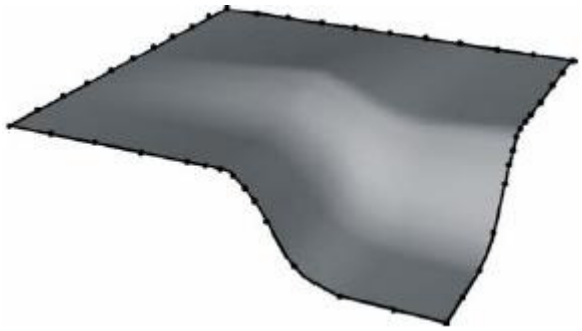


Figura 16 Superfície com as suas curvas de bordo considerando os parâmetros da adaptatividade geométrica das curvas.

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.

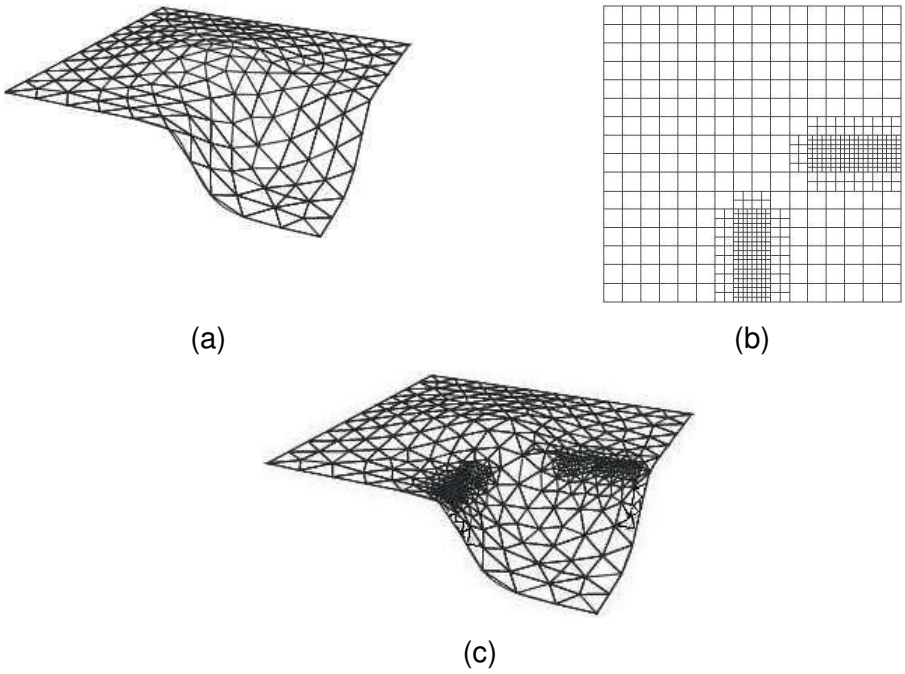
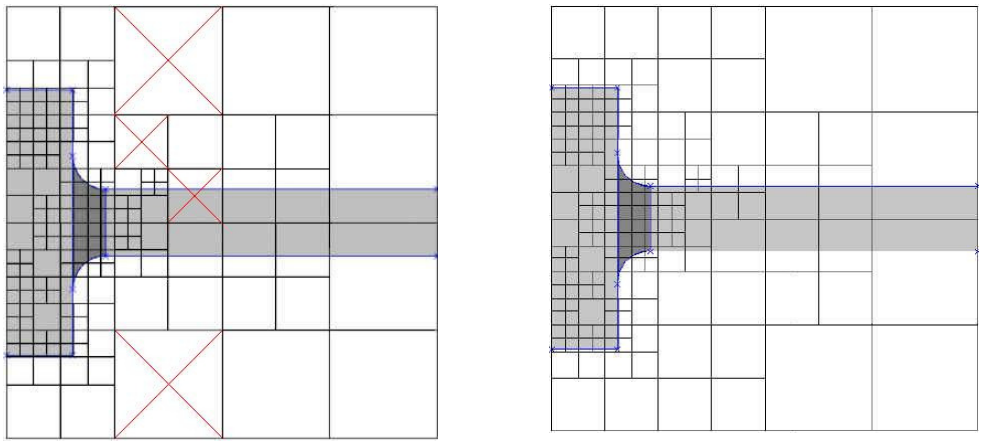


Figura 17 Exemplo de uma malha em uma superfície 3D: (a) sem considerar informações sobre curvaturas; (b) *quadtree* auxiliar construída para realizar a adaptatividade geométrica da superfície da figura (c) e (c) malha da superfície após se considerar as suas curvaturas (Lira [37]).

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.



(a) (b)

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.

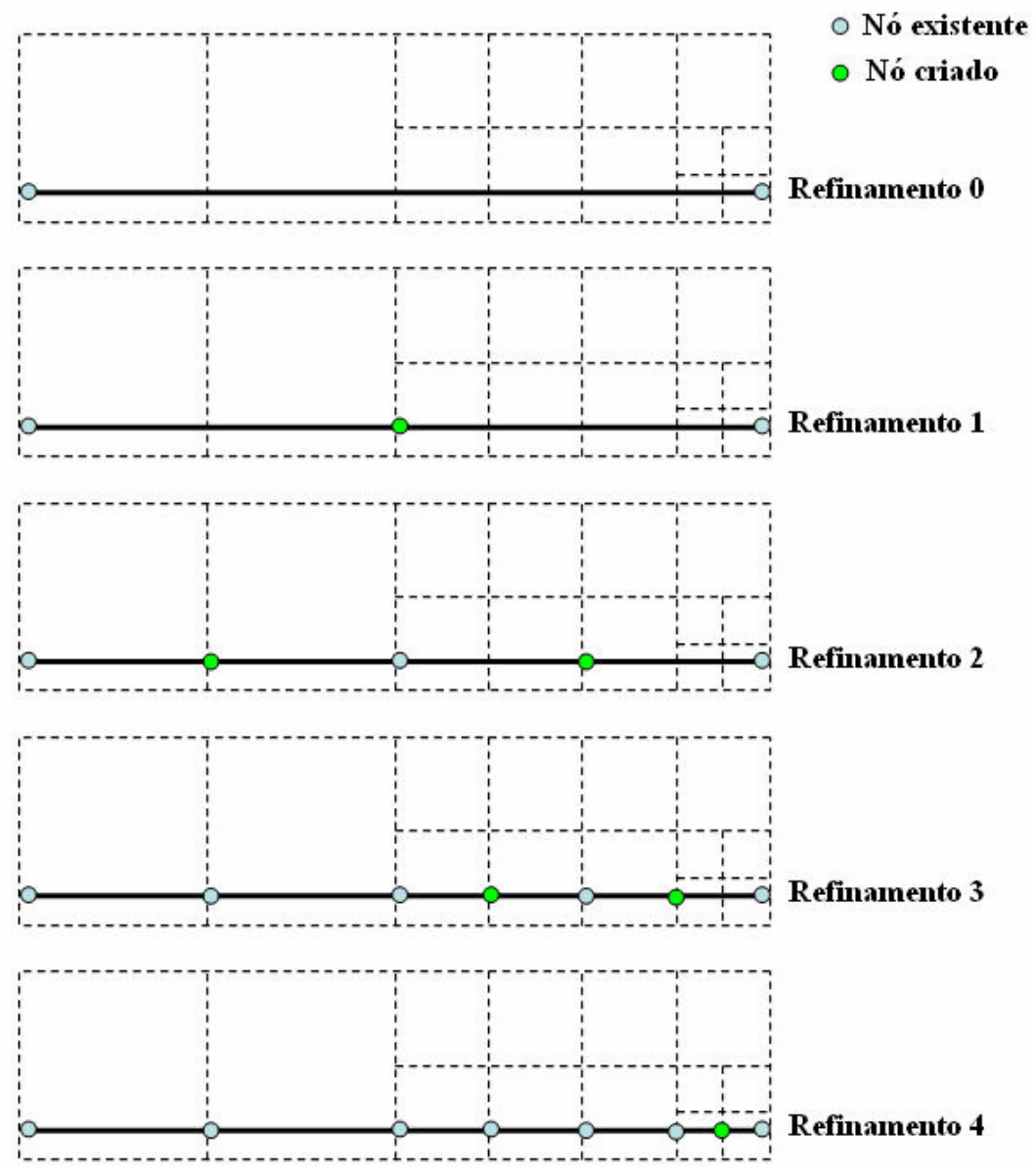


Figura 19 Refinamento de uma curva reta pela técnica de Decomposição Espacial Recursiva binária (*binary-tree*).

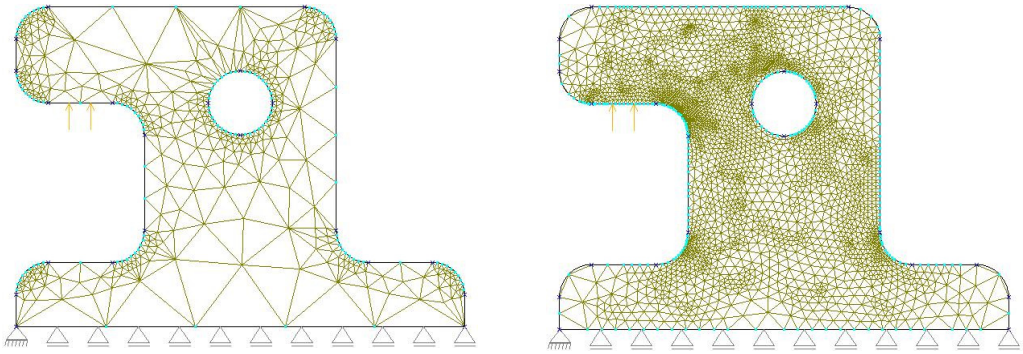
Esse uso da *octree*e / *quadtree* para refinar a curva e para controlar o tamanho dos elementos é inovador, pois na estratégia anterior de Cavalcante-Neto [12,13] para se refinar as curvas, buscava-se a informação sobre o tamanho característico (h) dos elementos adjacentes à curva, de tal forma que

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 curvas estavam mais discretizadas que o necessário (segundo os critérios de erro estabelecidos).



(a) (b)

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 da 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 algoritimo 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 requisítos específicos. 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.

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).

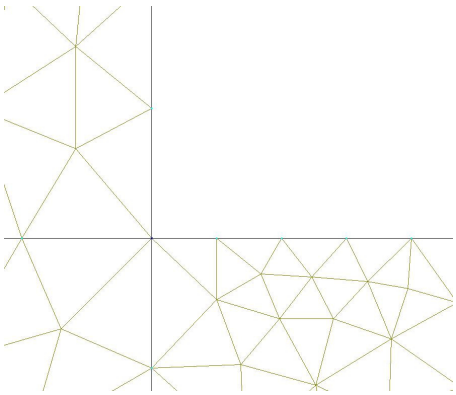
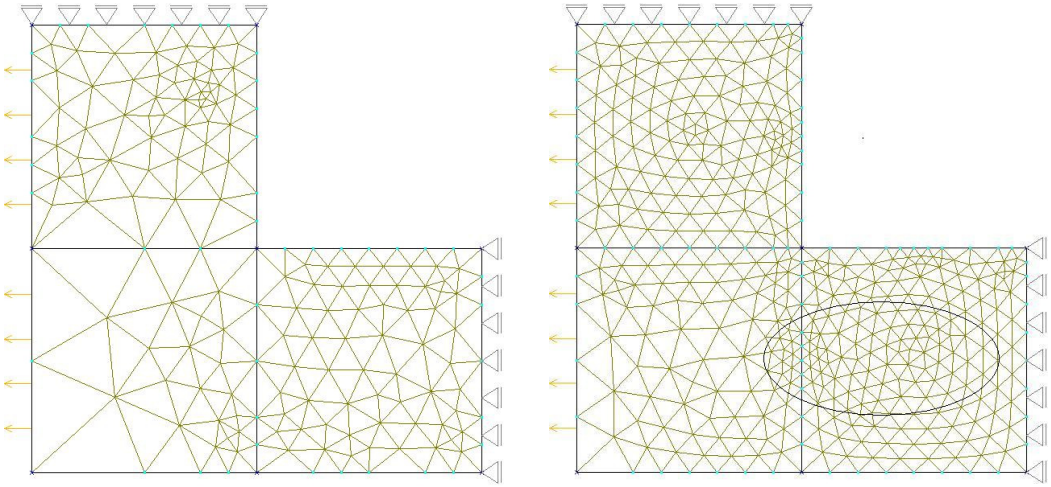


Figura 21 Problema de refinamento em cantos quando o valor de h de cada elemento é associado ao seu centro geométrico.

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.



(a) (b)

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 probelmas. 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 *quadtrees* 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.