

# Combinação das Técnicas de Quadtree e Delaunay para Geração Automática de Malhas de Elementos Finitos

JOAQUIM BENTO CAVALCANTE NETO  
MARCELO TILIO MONTEIRO DE CARVALHO  
LUIZ FERNANDO MARTHA

PUC-Rio – Pontifícia Universidade Católica do Rio de Janeiro  
Departamento de Engenharia Civil e  
ICAD – Laboratório de CAD Inteligente  
Rua Marquês de São Vicente, 225, Gávea  
22453 Rio de Janeiro, RJ, Brasil  
lfm@icad.puc-rio.br

**Abstract.** The paper proposes an algorithm for FEA mesh generation which is a combination of a quadtree technique and a Delaunay triangulation. The mesh is generated in two stages. Initially, the interior of the region is meshed using a quadtree approach. In a second stage, the region between the interior mesh and the boundary of the model is meshed using a Delaunay criterion and also the quadtree information to avoid extensive searches in the data structure.

## Introdução

Muitos sistemas de elementos finitos adotam uma estratégia de decomposição em subdomínios para geração automática da malha do modelo. Neste esquema, o domínio é subdividido em subdomínios, que são então discretizados em elementos finitos. Em geral, o grau de refinamento da malha é controlado pela discretização, a priori, das curvas que formam os contornos dos subdomínios. Os segmentos gerados ao longo destas curvas correspondem a lados de elementos a serem gerados. As principais vantagens desta estratégia são:

- A possibilidade de combinação de técnicas distintas de geração de malha em um único modelo, por exemplo, técnicas de mapeamento transfinito [Haber (1981)] e triangulação genérica.
- O refinamento a priori dos contornos dos subdomínios garante a compatibilidade da malha através desses subdomínios.
- A discretização do contorno do modelo é controlada explicitamente e fica sempre conformada com as propriedades geométricas de suas curvas. Isto acarreta em um refinamento mais uniforme no contorno.
- Em alguns tipos de simulação, como a de propagação de trincas em meios contínuos, a preservação da discretização original (fornecida) do contorno de um subdomínio é muito importante.

Diversos algoritmos com diferentes estratégias para geração automática de malhas de elementos finitos em regiões bidimensionais têm sido publicados recentemente na literatura. Dentre estes, destacam-

se os algoritmos baseados em técnicas de subdivisão espacial recursiva utilizando *quadtree* [Yerry (1984), Baehmann (1987)] e os baseados nos critérios para triangulação de Delaunay [Joe (1986), Chew (1989), Lo (1989), Vianna (1992), Potyondy (1993)].

A utilização da técnica de *quadtree*, devido às propriedades de uma árvore quaternária, faz com que estes algoritmos sejam muito mais rápidos do que os demais, além de criar uma boa transição entre regiões com diferentes graus de refinamento da malha. Entretanto, esta técnica, na sua forma original, não mantém a compatibilidade com uma discretização fixa, previamente fornecida, para o contorno da região, pois são criados novos vértices durante o processo de geração da malha.

Nos algoritmos baseados na triangulação de Delaunay esta discretização do contorno não só é garantida como faz parte dos dados de entrada para o próprio algoritmo. Exemplos de algoritmos que utilizam esta técnica para a geração de malhas de elementos finitos em regiões com contornos arbitrários podem ser encontrados nos trabalhos de Shaw (1978) e Lo (1989). Embora existam algumas variações entre esses algoritmos, a idéia básica é utilizar um procedimento de contração do contorno juntamente com critérios de Delaunay. O processo de geração da malha é desenvolvido em duas etapas. Primeiro são gerados os pontos no interior da região. Diferentes técnicas são aplicadas para esta etapa. Em seguida são gerados os triângulos da malha utilizando-se a idéia da contração do contorno. Este procedimento é baseado em uma lista de arestas (inicialmente esta

lista corresponde às arestas referentes ao contorno da região), onde cada aresta é considerada como base para a formação de um triângulo. Deve-se, então, procurar em uma lista de pontos, formada pelos pontos gerados na primeira etapa e os pontos devido à discretização do contorno, qual o ponto que forma com a aresta base o melhor triângulo. A definição de melhor triângulo é baseada em uma propriedade da triangulação de Delaunay. Após a formação de um novo triângulo, a lista de arestas é atualizada, com a aresta base do triângulo sendo retirada da lista e as novas arestas criadas sendo potencialmente acrescentadas. O algoritmo termina quando não houver mais arestas na lista. Estes algoritmos, entretanto, apresentam um desempenho não muito eficiente, pois para cada aresta deve-se procurar entre todos os pontos qual é o melhor candidato, o que representa uma complexidade de ordem quadrática ( $O(n^2)$ ) na análise do pior caso.

Deve-se observar que existem algoritmos para triangulação por Delaunay de regiões com restrições (regiões não convexas e com furos) com desempenho  $O(n \log n)$  [Chew (1989), De Florani (1992)]. Estes algoritmos, entretanto, são de implementação complexa e normalmente não são utilizados para elementos finitos.

Este trabalho descreve um algoritmo para geração automática de malhas de elementos finitos triangulares ou quadrilaterais, no contexto da estratégia de decomposição de um domínio em subdomínios. A idéia do algoritmo consiste na combinação das técnicas de *quadtree*, para geração dos elementos no interior da região, e triangulação de Delaunay, para geração dos elementos entre o contorno da região e a malha gerada por *quadtree*. Um aspecto fundamental deste algoritmo, que resulta em uma melhor eficiência computacional, consiste na utilização das informações de adjacência da estrutura *quadtree* também para o processo de triangulação de Delaunay.

### Subdivisão espacial recursiva e quadtree

A *quadtree* é uma estrutura de dados que pode ser usada para implementar um algoritmo de subdivisão espacial recursiva [Samet (1984)]. Algoritmos de subdivisão espacial recursiva subdividem uma região particular no espaço (bidimensional ou tridimensional) em regiões de forma similar. O processo é então repetido um número arbitrário de vezes para as regiões menores. A figura 1 mostra dois exemplos de procedimentos de subdivisão espacial recursiva: no primeiro, uma região triangular é subdividida em quatro regiões similares triangulares e, no segundo, o mesmo é feito para uma região quadrangular.

Os algoritmos de subdivisão espacial recursiva somente trabalham com um número pequeno de formas de regiões, no caso triângulos, quadrados ou hexágonos para regiões bidimensionais, e tetraedros, cubos ou hexaedros para regiões tridimensionais. Em regiões bidimensionais a forma mais usada é o quadrado. Neste tipo, cada região ou é subdividida em quatro subregiões similares, ou não é subdividida, dependendo do caso. Essa informação pode ser convenientemente armazenada em uma estrutura de dados do tipo árvore, onde cada nó ou tem quatro ou nenhum filho (*quadtree*), no caso bidimensional, ou tem oito ou nenhum filho (*octree*), no caso tridimensional.

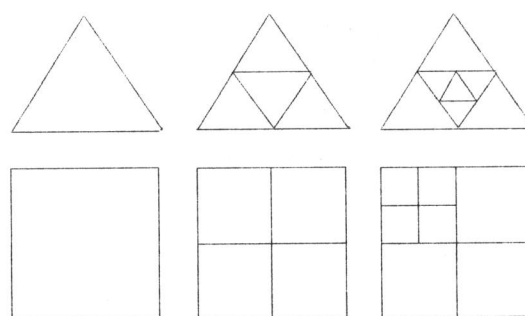


Figura 1: Procedimento de subdivisão recursiva triangular e quadrangular.

A figura 2 mostra um exemplo simples de uma região subdividida e sua *quadtree*. As regiões não divididas, que correspondem a nós folhas na árvore, são chamadas *células*. O tamanho de uma célula pode ser facilmente determinado por sua profundidade na árvore.

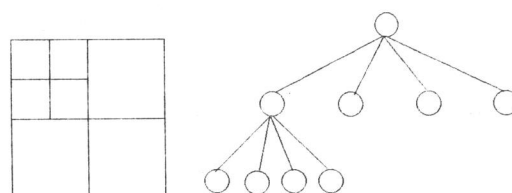


Figura 2: Uma região decomposta e sua árvore quaternária (*quadtree*).

### Geração da malha no interior do domínio

Esta seção define como a malha é criada no interior do domínio através do uso da *quadtree*, deixando uma região entre esta malha interna e o contorno do domínio para ser gerada usando uma técnica de triangulação de Delaunay. A geração da malha no interior do modelo pode ser dividida em algumas fases. Estas fases são descritas nas sub-seções a seguir, adotando-se um exemplo hipotético simples, mostrado na figura 3, que tem como único objetivo uma melhor definição das fases.

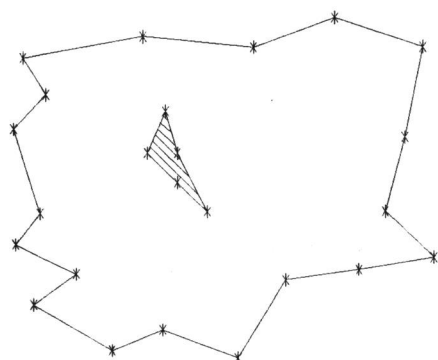


Figura 3: Exemplo e sua discretização.

#### Criação da árvore inicial

O processo é iniciado pela criação da árvore *quadtree* baseada na discretização do contorno (externo e furos) que é fornecida. Primeiro, as coordenadas máximas e mínimas do contorno são usadas para gerar uma caixa envolvente do modelo que representa o nó raiz da *quadtree*. Percorre-se, então, todas as arestas do contorno, uma por vez, para criar a árvore usando-se o seguinte procedimento: o comprimento e o ponto médio da aresta são calculados, e uma busca é feita no estado atual da árvore para saber em que célula este ponto está; verifica-se então se o tamanho desta célula é maior que uma percentagem do comprimento da aresta e neste caso subdivide-se esta célula, continuando o processo até o tamanho da célula ser menor que esta percentagem. Passa-se então a uma nova aresta e repete-se o procedimento. Com relação a esta percentagem, é recomendado um fator entre 0.7 e 1.4 [Potyondy (1993)] e neste algoritmo usou-se 1.0 (100 %), pois testes com vários tipos de contorno foram feitos e de um modo geral o uso deste valor foi o que gerou melhores resultados.

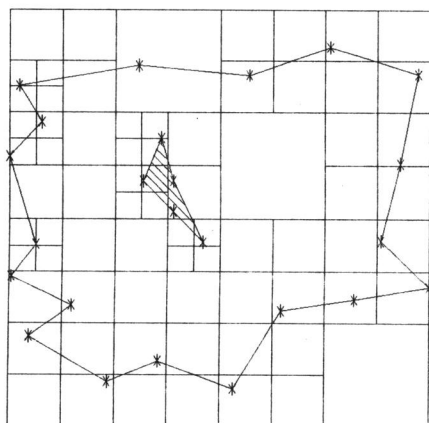


Figura 4: Criação da árvore inicial.

A figura 4 mostra a árvore gerada pelo processo acima descrito. As células da árvore podem ser de quatro tipos: *células exteriores*, que são células completamente fora do domínio; *células vértices*, que são células que contém no seu interior algum vértice do contorno do domínio; *células do contorno*, que são células atravessadas por alguma aresta desse contorno; e *células interiores*, que estão completamente dentro do domínio. Essa classificação será importante nas próximas fases.

#### Ajustes para garantir somente um nível de diferença entre células adjacentes

Criada a árvore inicial, ajustes devem ser feitos para garantir só um nível de diferença de profundidade na árvore entre células adjacentes. Isto auxilia na etapa de geração dos elementos finitos em células interiores, onde são utilizados padrões previamente definidos. Estes padrões dependem da configuração da *quadtree* no que diz respeito às células adjacentes da célula na qual são gerados os elementos finitos. Isto será abordado com mais detalhes na seqüência. Além disto, um único nível de diferença assegura uma boa qualidade de transição entre os elementos na malha a ser gerada. A figura 5 mostra a árvore depois de assegurado um único nível de diferença entre células adjacentes. Neste exemplo, esta fase só afetou uma única célula, na parte superior da árvore, que foi então dividida. Isto pode ser notado na figura 5, quando comparada com a figura 4 da árvore inicial.

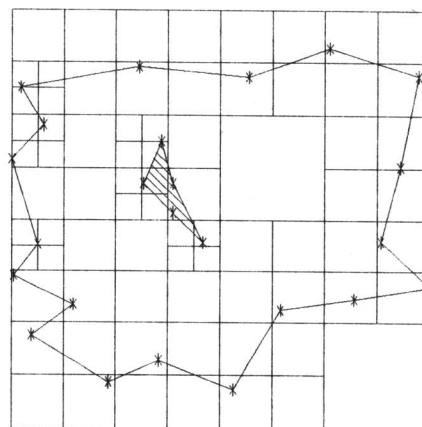


Figura 5: Árvore após fase de um único nível de diferença.

#### Eliminação de células perto do contorno

No método proposto, todas as células da árvore que forem interiores ao domínio irão gerar elementos finitos através de padrões. Na região entre as

células interiores e o contorno externo os elementos finitos serão gerados utilizando-se a triangulação de Delaunay e uma técnica de contração do contorno, conforme mencionado anteriormente. Precisa-se então classificar bem o que são células interiores, pois células que estiverem muito perto do contorno poderão ocasionar elementos finitos de forma ruim (ângulos muito agudos, por exemplo) ao se processar a triangulação de Delaunay. A solução para isso é não considerar como células interiores as células que estejam a uma distância de uma aresta do contorno menor que uma percentagem do seu comprimento. No caso desse algoritmo adotou-se um fator de 0.2 (20%).

*Ajustes para forçar o máximo tamanho de célula no contorno*

Pode-se notar nas figuras 4 e 5 que existem células muito grandes no interior da *quadtrees*. Desta forma vão ser gerados elementos finitos também muito grandes nestas regiões, o que pode não ser desejável. De modo a evitar isto, cada célula interior é subdividida até pelo menos o mesmo nível da maior célula do contorno. Se o tamanho da célula já for igual ou menor do que a maior célula do contorno, nada é feito. A figura 6 mostra a árvore após estes ajustes. Nota-se que esta fase, neste exemplo, só atingiu uma única célula.

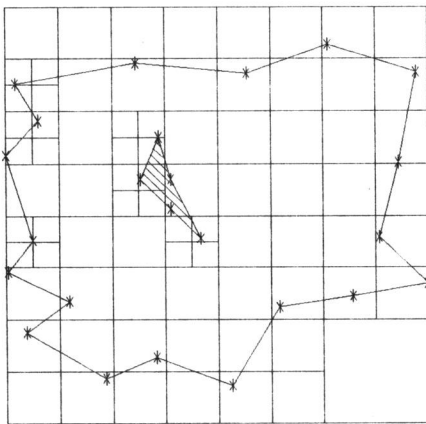


Figura 6: Árvore após fase para forçar o máximo tamanho de célula no contorno.

*Geração de malha em células interiores por padrões*

Nesta fase são gerados os elementos finitos em cada célula interior, seguindo os padrões mostrados na figura 7 [Baehmann (1987)]. Estes padrões são definidos de acordo com o número de células adjacentes de cada célula. É considerado que existe somente um nível de diferença entre células adjacentes.

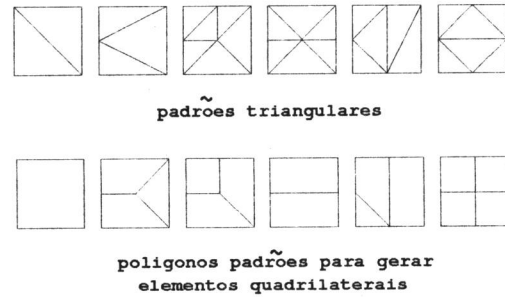


Figura 7: Padrões para elementos triangulares e quadrilaterais.

Para a malha com elementos triangulares são utilizados os padrões mostrados na parte superior da figura 7, para subdividir a célula em triângulos, que correspondem aos próprios elementos finitos. No caso da malha com elementos quadrilaterais, a célula é subdividida em polígonos (padrões da parte inferior da figura 7). Estes polígonos serão, então, subdivididos em um passo posterior, em elementos finitos quadrilaterais conforme o seguinte procedimento: cria-se um ponto na metade de cada lado de cada polígono, insere-se um ponto no centróide do polígono e liga-se este ponto a cada ponto médio dos lados do polígono [Baehmann (1987)]. Nota-se a importância de garantir apenas um único nível de diferença entre células adjacentes na árvore, como foi descrito em fase anterior, uma vez que estes padrões consideram, no máximo, um ponto intermediário em cada lado da célula.

Com o uso destes padrões, gera-se então a primeira parte da malha, parte esta que ocupa normalmente a maior área do domínio. Na figura 8 é mostrado o estágio do algoritmo após a etapa de geração da malha interna utilizando a *quadtrees*.

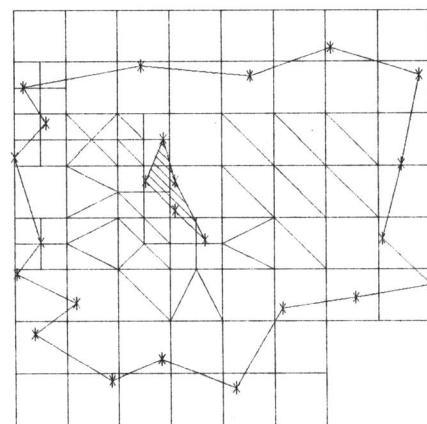


Figura 8: Geração de malha no interior do modelo.

Pode-se notar, também na figura 8, um exemplo de uma célula (a célula à esquerda do vértice reentrante no lado direito do contorno) que seria inicialmente classificada como interior, mas é reclassificada de acordo com o critério explicado anteriormente como célula de contorno.

### Geração da malha no contorno

Esta seção define como a malha é criada na região entre a *quadtree* interna e o contorno. Isto é feito através da técnica de triangulação de Delaunay por uma contração de contorno. Isto completa o método de geração de malhas de elementos finitos proposto.

Também aqui pode-se dividir a geração em algumas fases, mostradas nas sub-seções a seguir.

#### Inicialização da lista de arestas ativas do contorno

O processo se baseia em uma lista de arestas que representa o contorno ativo ou corrente da região. Esta lista contém, inicialmente, todas as arestas referentes à discretização das fronteiras da região. Cada aresta desta lista será considerada como aresta base para a formação dos triângulos. O processo é iniciado por uma aresta qualquer desta lista.

#### Escolha de um vértice interior para formação de um elemento finito

Para cada aresta base deve-se encontrar um vértice que forme com esta aresta o melhor triângulo. Este triângulo é definido, baseado na propriedade da triangulação de Delaunay, como sendo o triângulo com o maior ângulo formado pelo vértice candidato e os vértices da aresta base, como se pode ver pela figura 9.

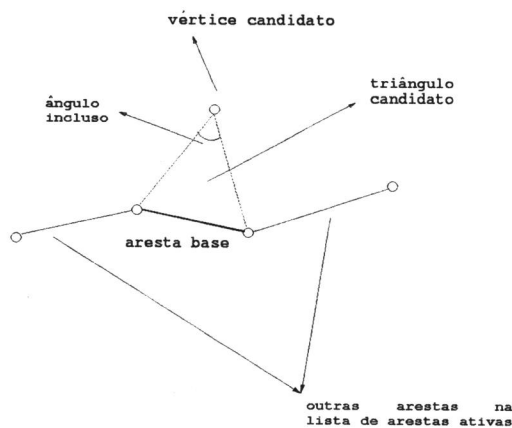


Figura 9: Definição do melhor triângulo pela técnica de Delaunay.

A escolha do vértice para formar um novo triângulo deve seguir, então, o critério explicado no

parágrafo anterior e que é utilizado em diversos algoritmos existentes. O aspecto fundamental do algoritmo proposto consiste na determinação dos vértices que serão candidatos a formar um novo triângulo e sobre os quais são efetuadas operações geométricas, tais como cálculo de ângulo. Este processo, que utiliza as informações da *quadtree* gerada na primeira etapa, é feito da seguinte maneira: localizam-se as células da *quadtree* que contenham respectivamente os nós inicial e final da aresta base, identificando-se, então, as suas células adjacentes. Células adjacentes são as células que possuem pelo menos um ponto em comum. A determinação das células adjacentes é um processo extremamente rápido, pois consiste em percorrer uma árvore quaternária. Os vértices candidatos a formar o novo triângulo devem pertencer, portanto, a estas células, podendo ser vértices do contorno original da região ou vértices da malha interna gerada anteriormente. Desta forma tem-se um número reduzido de vértices para os quais o ângulo deve ser computado, aumentando a eficiência do algoritmo. Isto torna-se mais significativo se for considerado que são feitos também testes geométricos para a validação do vértice candidato, que deve estar dentro do domínio no qual está sendo gerada a malha e também não pode formar arestas que interceptem alguma aresta existente.

No caso de não existir nenhum vértice candidato após o procedimento descrito acima, deve-se considerar vértices que pertençam a células adjacentes às células adjacentes iniciais. Estes casos ocorrem com pouca frequência e refletem na verdade uma discretização inadequada para o contorno da região.

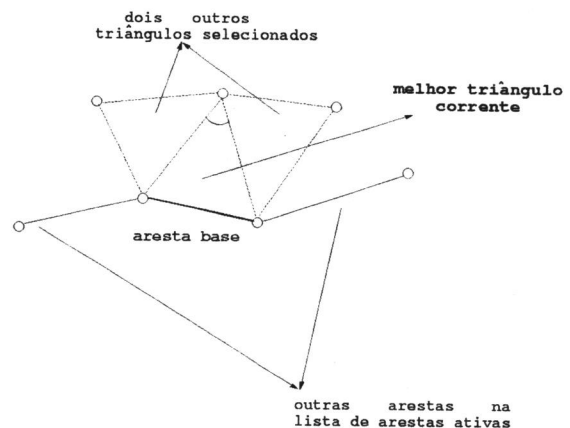


Figura 10: Identificação dos dois melhores triângulos adjacentes.

Caso se deseje gerar uma malha com elementos quadrilaterais ao invés de triangulares o que se faz é, após a identificação do melhor triângulo em



relação a uma determinada aresta base, selecionar um outro triângulo usando as duas arestas deste melhor triângulo como arestas base. Os dois triângulos combinados devem formar um quadrilátero aceitável dentro de certos critérios. Aqui três possibilidades existem (veja figura 10): nenhum triângulo adjacente é selecionado (então só o melhor triângulo é atualizado na estrutura de elementos finitos), um único triângulo adjacente é selecionado, ou dois triângulos adjacentes são selecionados.

Se um ou dois triângulos adjacentes forem selecionados, então tem-se um ou dois quadriláteros possíveis que devem ter sua forma estudada para aceitação ou não. Para isso, uma medida numérica de forma  $S$  é definida, como exemplificada na figura 11 [Potyondy (1993)], representando o desvio de qualquer ângulo interno do elemento finito em relação a 90 graus. O elemento é aceitável se  $S \leq 1.5$ , que corresponde a ângulos internos entre 45 e 135 graus.

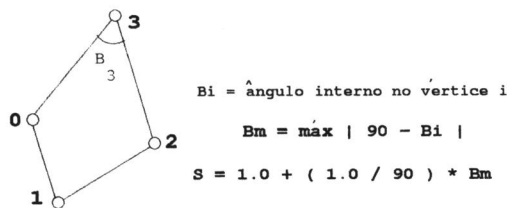


Figura 11: Definição da medida de forma quadrilateral  $S$ .

#### Atualização da estrutura de dados

Quando se chega a essa fase, um novo elemento finito triangular ou quadrilateral, dependendo do caso, foi formado. Esse elemento é adicionado na lista de elementos finitos já existentes e é feita uma atualização na lista de arestas ativas para fazer a contração do contorno propriamente dita. A técnica de contração de contorno tradicional é tal que, formado um elemento finito, a aresta base é retirada da lista de arestas ativas e cada uma das arestas formadas (duas na triangulação ou três na quadrangulação) é inserida na lista, se for uma nova aresta, ou retirada dela, se já existir. Isto faz com que o novo contorno corresponda à região na qual ainda não foi gerada a malha. Conforme a malha vai sendo gerada, este contorno vai contraindo até que seja gerada a malha em toda a região inicial.

No algoritmo apresentado a diferença em relação ao processo descrito acima é que o processo de contração do contorno não se aplica a todo o domínio mas somente à região entre as fronteiras da região original e a malha gerada pela *quadtrees*, o que representa uma lista de arestas reduzida em relação ao

procedimento tradicional, aumentando a eficiência do algoritmo.

#### Finalização da contração do contorno

O processo descrito acima continua até que não haja mais nenhuma aresta na lista de arestas ativas. Neste momento tem-se a malha completamente gerada, conforme é mostrado na figura 12.

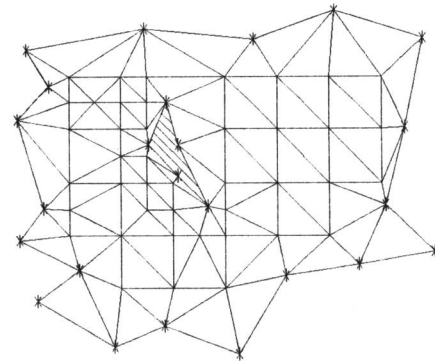


Figura 12: Malha final gerada.

#### Suavização da malha

Finalmente, após ser gerada toda a malha, executa-se um processo de *suavização*. A suavização é simplesmente uma média, para cada vértice, considerando as coordenadas dos vértices pertencentes aos elementos finitos adjacentes ao vértice em questão. Essa média é repetida quatro ou cinco vezes para dar um resultado final satisfatório e afeta logicamente só os vértices interiores, já que não se pode alterar as coordenadas dos vértices originais do contorno. A figura 13 mostra a malha final suavizada e comparando-se com a figura 12 pode-se notar a melhora conseguida.

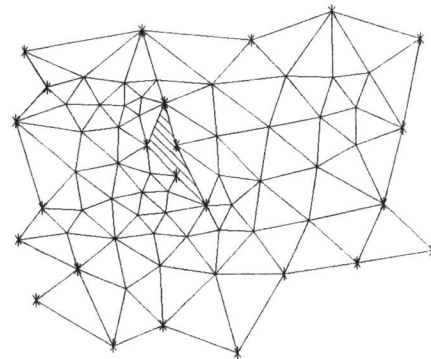


Figura 13: Malha final suavizada.

### Conclusões

O algoritmo descrito neste trabalho pode ser considerado um algoritmo híbrido entre o proposto por Shephard [Baehmann (1987)] e o proposto por Potyondy (1993), tentando aliar o que há de melhor nos dois.

O algoritmo proposto por Shephard [Baehmann (1987)] é todo ele baseado na estrutura *quadtree*, isto é, a malha é totalmente gerada através da manipulação das células da *quadtree* criadas para o modelo em questão. Este algoritmo é robusto, e bastante rápido pois percorre uma árvore quaternária para gerar a malha, mas tem a característica de não manter a discretização original do contorno, criando novos vértices no contorno não definidos originalmente.

O algoritmo de Potyondy (1993), assim como o de Vianna (1992), usa a *quadtree* somente para criar os pontos interiores no domínio, descartando a partir deste momento toda a informação referente à mesma. Estes pontos interiores, então, vão servir para a geração dos elementos finitos seguindo a técnica de contração do contorno. Esse algoritmo mantém a discretização original do usuário, mas não é muito eficiente pois apresenta uma complexidade de ordem quadrática na análise de pior caso.

O algoritmo proposto procura aliar as boas características de ambos os métodos citados e criar uma geração eficiente, robusta e rápida. O contexto deste algoritmo mantém a discretização original do contorno, que em um sistema interativo é normalmente definida pelo usuário. Isto é importante para fazer a ligação com outros algoritmos (mapeamentos, por exemplo), para simulação de propagação de trincas, e para o usuário ter um maior controle sobre a malha a ser gerada.

Dentro deste contexto, o algoritmo usa a *quadtree* para gerar a malha somente no interior do modelo, simplificando um pouco o processo proposto por Shephard [Baehmann (1987)], que inclui operações não triviais para tratamento da *quadtree* na região próxima do contorno a fim de evitar elementos finitos ruins nesta área. Na fase da contração do contorno, a substituição da procura de vértices para formação de triângulos, para cada aresta base, em uma lista de vértices completa pela procura em uma lista de vértices que pertencam às células adjacentes à aresta base evita testes exaustivos contra vértices distantes da aresta base que não formariam elementos finitos melhores. Isto torna a performance do algoritmo bem superior pois ele deixa de ter complexidade quadrática ( $O(n^2)$ ), levando a ganhos bem altos em modelos grandes e complexos.

### Agradecimentos

Este trabalho foi desenvolvido no Laboratório de CAD Inteligente – ICAD – da PUC-Rio. Os autores agradecem ao Grupo de Tecnologia em Computação Gráfica – TeCGraf – da PUC-Rio pelo suporte e infra-estrutura fornecidos na manutenção do laboratório ICAD. O grupo TeCGraf é suportado financeiramente principalmente pelo Convênio CENPES/Petrobrás – PUC-Rio.

### Referências

- Baehmann, P.L., Wittchen, S.L., Shephard, M.S., et al., "Robust Geometrically Based, Automatic Two-Dimensional Mesh Generation," *Int. J. Num. Meth. Engng.*, **24**, pp. 1043-1078, 1987.
- Chew, L.P., "Constrained Delaunay Triangulation," *Algorithmica*, **4**, pp. 97-108, 1989.
- De Florani, L. and Puppo E., "An On-line Algorithm for Constrained Delaunay Triangulation," *Computer Vision, Graphics, and Image Processing*, **54**, pp. 290-300, 1992.
- Haber, R., Shephard, M.S., Abel, J.F., Gallagher, R.H., and Greenberg, D.P., "A General Two-Dimensional, Graphical Finite Element Pre-processor Utilizing Discrete Transfinite Mappings," *Int. J. Num. Meth. Engng.*, **17**, pp. 1015-1044, 1981.
- Joe, B., "Delaunay triangular meshes in convex polygons," *SIAM J. Sci. Stat. Comput.*, **7**, pp. 514-539, 1986.
- Lo, S.H., "Delaunay Triangulation of Non-Convex Planar Domains," *Int. J. Num. Meth. Engng.*, **28**, pp. 2695-2707, 1989.
- Potyondy, D.O., "A Software Framework for Simulating Curvilinear Crack Growth in Pressurized Thin Shells," Ph.D. Thesis, School of Civil Engineering, Cornell University, 1993.
- Samet, H., "The Quadtree and Related Hierarchical Data Structures," *ACM Computer Surveys*, **16**, No. 2, 1984.
- Shaw, R.D. and Pitchen, R.G., "Modifications to the Suhara-Fukuda Method of Network Generation," *Int. J. Num. Meth. Engng.*, **12**, pp. 93-99, 1978.
- Vianna, A.C., "Modelagem Geométrica Estendida para Modelos Bidimensionais de Elementos Finitos," Tese de Mestrado, Depart. Eng. Civil, PUC-Rio, 1992.
- Yerry, M.A. and Shephard, M.S., "Automatic Three-Dimensional Mesh Generation by Modified-Octree Technique," *Int. J. Num. Meth. Engng.*, **20**, pp. 1965-1990, 1984.