



Triangulações adaptativas em multirresolução para objetos heterogêneos

Multi-resolution adaptive triangulations for heterogeneous objects

Ricardo Marroquim | Paulo Roma Cavalcanti

resumo

Este artigo apresenta um método para gerar triangulações adaptativas em multirresolução de objetos 3D com várias regiões, geometria complexa e cujas fronteiras não são, necessariamente, variedades. O método pode ser empregado em modelos geológicos os quais contêm várias características que dificultam o uso de algoritmos de triangulação tradicionais, como, por exemplo, falhas, *pinch-outs* e cavidades. O algoritmo tem como objetivo gerar malhas que não contenham elementos malformados e que suportem deformações, sendo desta forma adequados a simulações numéricas. A primeira etapa do estudo é mergulhar o modelo em uma

malha tetraédrica adaptativa semi-regular, conhecida como multitriangulação binária (*binary multiresolution triangulation - BMT*). Os elementos da malha são, então, empurrados em direção à fronteira original por meio de um sistema de compressão baseado em física, no caso deste trabalho, um sistema massa-mola. A triangulação final não contém tetraedros degenerados e oferece uma aproximação da fronteira do modelo baseada em uma escolha de um nível de resolução adequado.

(originais recebidos em 10.10.2006)

Palavras-chave: malhas volumétricas | multirresolução | simulações numéricas

abstract

This paper presents a method for generating multi-resolution triangulations of non-manifold 3D objects composed by several regions with arbitrary geometry. Our target application comes from Geosciences in which geological models for instance are often composed by several regions (forming geological layers), and have an irregular geometry, with surfaces that meet at very small angles (pinch-out). The goal is to generate meshes without bad elements that are able to support deformations and that can be used in numerical simulations. The algorithm has two main steps. In the first step, a grid is adaptively created over the object in order to subdivide the surrounding space. The second step shapes the vertices of the grid to the boundary of the object using a compression scheme, while trying to maintain well shaped elements. Springs are placed on each tetrahedron to prevent them from collapsing and to restrain vertices movements. The final triangulation usually has no degenerated triangles and provides an approximation of the boundary based on a selected resolution.

(expanded abstract available at the end of the paper)

Keywords: volumetric meshes | multi-resolution | numerical simulations

introdução

As simulações numéricas se tornaram ferramentas importantes no desenvolvimento de produtos de engenharia e predição do comportamento de fenômenos físicos, como condições climáticas, geração e migração de óleo, movimento das ondas e terremotos.

Para fazer uso de uma simulação numérica, é necessário que o domínio em questão seja discretizado. Nesta discretização, geralmente conhecida como triangulação, uma série de equações, descrevendo leis físicas, devem ser resolvidas.

Vários algoritmos de triangulação foram propostos nos últimos anos. Dentre eles, aqueles conhecidos como métodos de Delaunay obtiveram um nível alto de aceitação em muitas áreas. Porém, eles não foram tão bem-sucedidos em

três dimensões como o foram em duas (Molino *et al.* 2003), devido ao fato de produzirem tetraedros com volume desprezível, conhecidos como tetraedros “chatos”. Shewchuk (1998) oferece uma visão geral dos métodos de Delaunay e razões pelas quais alguns resultados teóricos podem ser difíceis de serem obtidos na prática. Cheng *et al.* (2000) discutem vários métodos para remover os tetraedros “chatos”, contudo admitem que sua estimativa ainda é muito pessimista. Edelsbrunner e Guoy (2002) mostram que o método de Cheng *et al.* (2000) pode apresentar problemas perto das fronteiras.

A maioria das aplicações numéricas requer triangulações que preservem fielmente a fronteira de um modelo. Em três dimensões, isto significa que a fronteira deve corresponder a faces triangulares de tetraedros. Enquanto o método tradicional de Delaunay triangula apenas o fecho convexo do modelo, várias extensões foram propostas para garantir que fronteiras poligonais arbitrárias estejam presentes na triangulação final. Muitos destes métodos constroem primeiro uma triangulação de Delaunay não restrita, e depois inserem novos vértices para recuperar as faces ou arestas ausentes (Murphy *et al.* 2001; Cohen-Steiner *et al.* 2002; George e Borouchaki, 1998; Weatherhill e Hassan, 1994; Shewchuk, 2002).

Como alternativa para os métodos baseados em Delaunay, existem os métodos de “Avanço de Fronteira” (Mavriplis, 1995; Lohner e Parikh, 1988; Peraire *et al.* 1999; Lohner 1996; Frey *et al.* 1996; Fleischmann e Siegfried, 1997). Estes métodos partem da fronteira (já triangulada) e adicionam novos vértices para criar tetraedros de boa forma, avançando a fronteira para o interior do modelo. A dificuldade está em compatibilizar as diferentes frentes sem gerar inconsistências geométricas. A criação dos novos elementos pode seguir uma variedade de metodologias, como a de empacotamento de esferas (Li *et al.* 1999) ou critérios de Delaunay (Mavriplis, 1995; Fleischmann e Siegfried 1997; Frey *et al.* 1996). Os métodos de avanço de fronteira possuem a característica de produzir malhas que se adequam perfeitamente à fronteira do polígono, o que é importante para algumas aplicações, como por



exemplo, na geração de malhas para peças mecânicas. A desvantagem principal do método está no fato de que é dependente da qualidade da poligonização da fronteira inicial. Em particular, triângulos malformados na frente inicial provavelmente resultarão em tetraedros malformados (Moller e Hansbo, 1995).

No caso de aplicações cuja fronteira mude com o tempo, como no caso de simulações de corpos deformáveis, é desejável que a topologia da malha mude o menos possível durante os passos de tempo. Molino *et al.* (2003) desenvolveram um método de triangulação baseado em um sistema físico no qual uma grade gerada sobre o modelo é aproximada por meio de um sistema de massa-mola. O método foi criado para gerar simulações de contrações de músculos, onde a malha deve suportar grandes deformações.

É apresentado neste trabalho um método baseado em física que utiliza uma BMT (multitriangulação binária) para ser capaz de produzir uma estrutura adaptativa e malhas em multirresolução (Mello *et al.* 2003). Em outras palavras, a natureza hierárquica da BMT possibilita não só uma adaptação ao domínio (produz uma triangulação mais fina nos locais de maior interesse), mas também um truncamento em qualquer nível inferior, produzindo uma aproximação mais grosseira sem custo adicional. As aplicações-alvo deste método são os modelos geológicos que têm fronteiras que evoluem com o tempo e, em algumas simulações, devem ser conduzidas em níveis diferentes de resolução. Além disso, modelos como estes contêm várias propriedades que dificultam sua discretização. Geralmente, possuem bordas complexas e irregulares, pois devem representar falhas, cavidades e diferentes materiais nas camadas dos terrenos. Os algoritmos tradicionais podem apresentar muita dificuldade para triangular este tipo de modelo, além de gerar elementos de má qualidade, o que certamente acarreta problemas numéricos.

O algoritmo proposto aqui pode ser dividido em dois passos principais. Primeiramente, uma malha adaptativa é gerada sobre o modelo original para subdividir o espaço. Num segundo passo, a malha gerada é comprimida utilizando um sistema físico para adequar-se à fronteira do modelo.

subdivisão espacial

Para gerar os elementos da triangulação, o primeiro passo do algoritmo consiste em subdividir o espaço que envolve o modelo gerando uma malha tetraedral. Os elementos desta subdivisão espacial serão rearranjados de forma a se adequarem à fronteira do modelo. No entanto, nenhum elemento novo é inserido após a fase de subdivisão.

Existem alguns pontos importantes na escolha da estrutura da malha: ela deve ser adaptativa, deve suportar multirresolução e deve gerar elementos com boa forma.

Uma estrutura adaptativa permite que coexistam elementos de tamanhos e formas diferentes na malha. Esta propriedade é necessária para que se obtenham elementos menores somente nos locais onde uma aproximação melhor é importante como, por exemplo, nas imediações da fronteira do modelo. Desta forma, obtém-se uma estrutura menos densa, economizando-se elementos onde for possível.

A segunda propriedade importante é o suporte à multirresolução. Deve ser possível gerar a subdivisão espacial de um mesmo modelo em diferentes níveis de resolução. A escolha do nível implica na quantidade de elementos gerados na malha e, conseqüentemente, na qualidade final da aproximação do modelo. Resoluções altas geram malhas densas e aproximações melhores, enquanto resoluções baixas geram malhas mais leves e aproximações mais grosseiras.

A terceira característica desejada é que a malha não possua elementos de má forma. Quanto menores os ângulos da subdivisão espacial inicial, mais chances de se obter elementos com formas ruins ao final da compressão. Um dos objetivos durante a compressão é manter, o mais que possível, a forma inicial dos elementos. Para tanto, é importante que a malha inicial os forneça numa configuração adequada, isto é, o mais próximo possível de um tetraedro perfeito, com ângulos diédricos entre 45° e 90° .

Nesta implementação é utilizada uma BMT por possuir todas as características desejadas. A estrutura inicial de uma BMT é um cubo dividido em seis tetraedros com ângulos diédricos de 45° . Para subdividir um tetraedro, a distância do seu

centróide à fronteira deve ser menor do que a média dos comprimentos de suas arestas.

Este tipo de estrutura também possui a vantagem de produzir uma transição gradual entre os elementos de níveis de refinamento diferentes. Um exemplo é apresentado na figura 1.

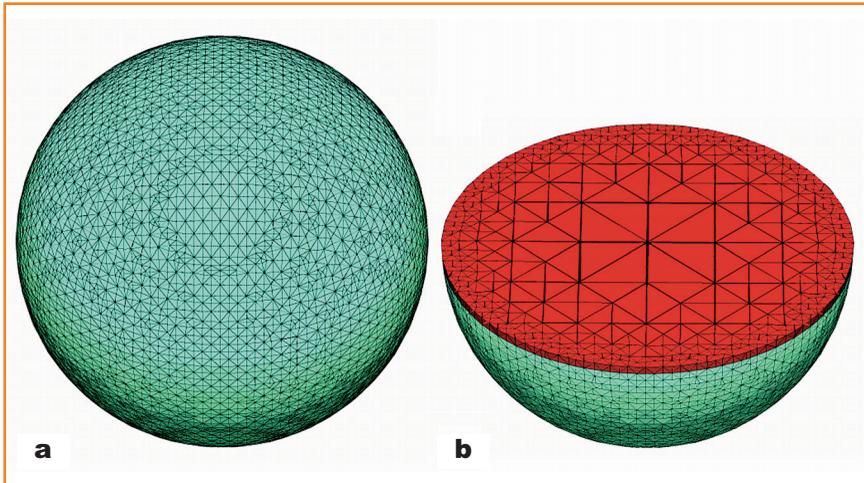


Figura 1

Uma esfera com elementos menores perto da fronteira e maiores no interior. A malha exibida já passou pelo processo de compressão.

Figure 1

A sphere with smaller elements near the boundary and larger elements inside. This mesh has already been exposed to the compression process.

Modelos geológicos geralmente possuem características que só podem ser representadas por estruturas de dados complexas. Estes modelos podem conter várias regiões, falhas, cavidades e normalmente não são variedades (*non-manifold*). Além disso, modelos que representam bacias sedimentares podem se estender por vários quilômetros, mas ter uma espessura relativamente pequena. Um exemplo de um modelo geológico pode ser visto na figura 2. Um corte bidimensional deste modelo é mostrado na figura 3.

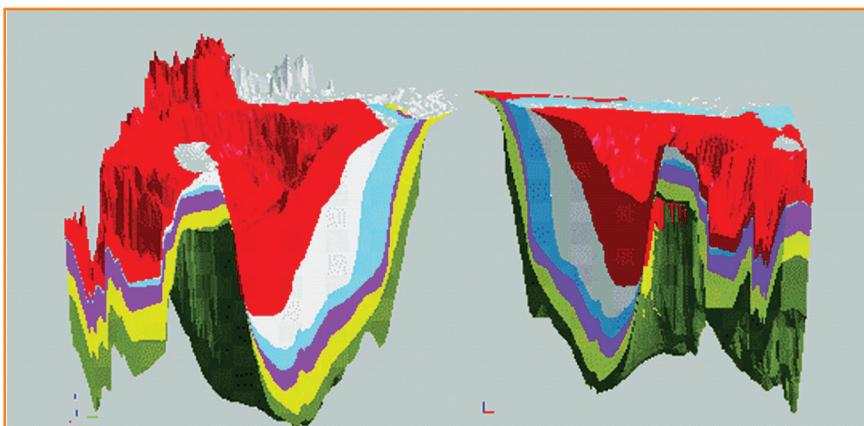


Figura 2 – Modelo em três dimensões da Bacia do Golfo do México.

Figure 2 – 3D Model of the basin of the Gulf of Mexico.

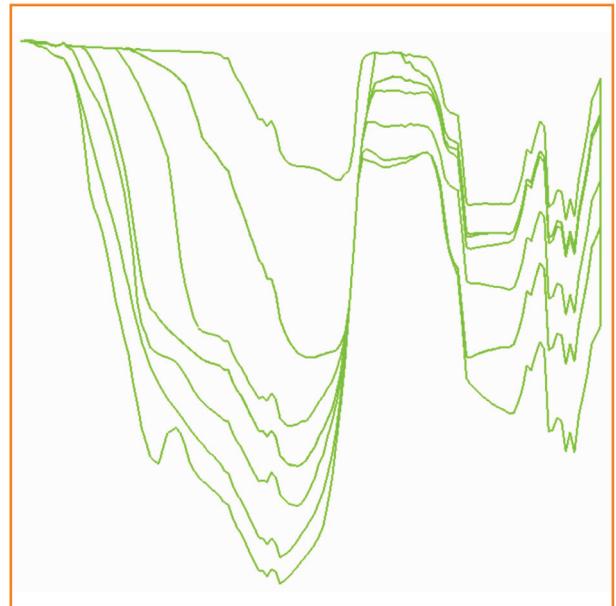


Figura 3 – Um corte bidimensional do modelo do Golfo do México.

Figure 3 – 2D cut view of the model of the Gulf of Mexico.

Para representar adequadamente modelos tridimensionais, é utilizada a estrutura de dados *radial-edge* (RED), que permite obter qualquer relação de adjacência em tempo constante (Weiler, 1988). As estruturas de dados dos modelos são criadas utilizando a biblioteca do sistema *complete geometric complexes* (CGC) (Cavalcanti *et al.* 1997).

função distância

Durante toda execução do algoritmo, a distância de um ponto no espaço à fronteira do modelo é computada diversas vezes. Portanto, uma função distância rápida se faz necessária e uma *octree* é utilizada para melhorar o desempenho.

O nó raiz da *octree* é definido pelas coordenadas da caixa envolvente do modelo. Para criar uma estrutura compacta, as coordenadas dos nós internos e folhas não são armazenadas, mas computadas em tempo real ao atravessar a *octree*. Os únicos atributos mantidos em um nó interno são ponteiros para seus oito filhos, enquanto um nó folha armazena somente uma lista de objetos intersectados pela sua caixa.

Especificamente, a lista de itens contém todas as faces do modelo que intersectam ou estão con-



tidas na caixa do nó folha. Quando o número de elementos desta lista ultrapassar um limite máximo pré-determinado, o nó é subdividido. No entanto, para que um nó não seja repartido indefinidamente, a altura da árvore também é limitada. Note que se existisse um vértice com mais faces incidentes do que o tamanho limite da lista, o nó contendo este vértice seria subdividido eternamente.

Para encontrar a menor distância de um ponto p à superfície do modelo, a árvore é atravessada a partir do nó raiz. Durante a execução do algoritmo, uma estimativa do melhor resultado é armazenada. Quando um nó folha é visitado, a distância para a face mais próxima é retornada. Quando um nó interno é visitado, os seus filhos são ordenados em ordem ascendente da distância de p aos seus octantes. Cada octante filho é visitado recursivamente, mas somente se a sua distância correspondente for menor do que a melhor estimativa até o momento. Se algum filho retornar um valor menor do que a melhor estimativa, esta é atualizada. Na figura 4 há o pseudo-código deste algoritmo.

```

global float melhor ← ∞
procedimento distância (ponto p, nó Octree T)
  se T é um nó folha então
    retorna a distância entre p e a face mais perto em T
  se não
    ordena os filhos de T em ordem ascendente de distância à p
    para cada filho S (ordenados) faça
      seja d a distância entre p e a caixa de S
      se d < melhor então
        D ← distância (p, S)
        se D < melhor então melhor ← D
    retorna melhor
    
```

modelo. O propósito é selecionar um subconjunto de elementos da malha, chamado de malha-d ou malha de deslocamento, e comprimi-lo para que se adeque à fronteira do modelo. Os vértices na fronteira da malha-d são marcados para serem projetados na direção da fronteira. Ao movimentar estes vértices marcados, as forças geradas são propagadas para os outros vértices, rearranjando os elementos interiores. Na figura 5 são mostradas a malha-d (a, b, c) e a triangulação final (d, e, f) de uma esfera em três resoluções diferentes.

Alguns vértices da malha devem ser selecionados ou marcados para serem empurrados na direção da fronteira do modelo. Idealmente, a malha-d deve estar o mais próxima possível da fronteira. Seguindo o esquema de Molino, a malha de

Figura 4

Pseudo-código para a função distância da octree.

Figure 4

Pseudo-code for the octree distance function.

malha de deslocamento

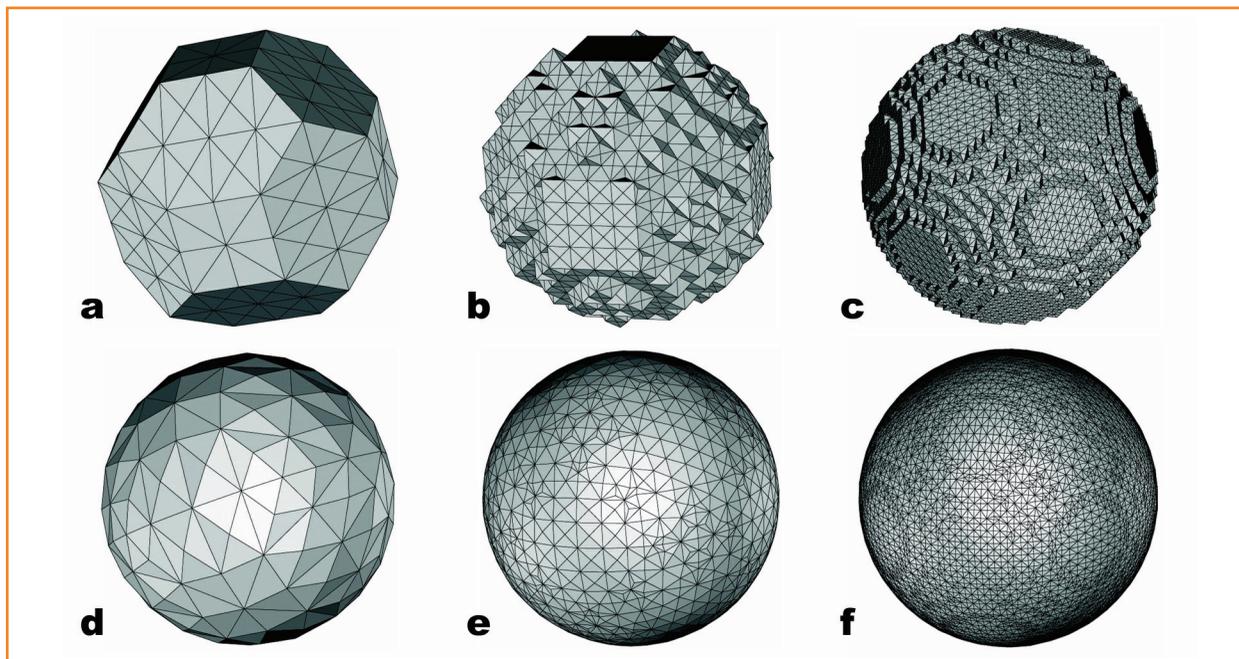
Um sistema físico de massa-mola é utilizado para adaptar a malha refinada à fronteira do

Figura 5

A malha-d de uma esfera antes (a, b, c) e depois (d, e, f) da compressão em três níveis de resolução diferentes.

Figure 5

Mesh-d of a sphere before (a, b, c) and after (d, e, f) compression at three different resolution levels.



deslocamento é composta por todos os tetraedros incidentes em um conjunto de vértices suficientemente interiores ao modelo, chamado de envelope interior (Molino *et al.* 2003). Este conjunto contém todos os vértices em que, pelo menos 25% das arestas incidentes estão no interior do modelo. Esta condição implica em que todos os tetraedros da malha-d possuam no mínimo um vértice no interior da fronteira do modelo.

Todos os vértices da malha-d que não pertencem ao envelope interior são marcados para serem empurrados em direção à fronteira. Este procedimento garante que nenhum tetraedro terá seus quatro vértices marcados. O conjunto de vértices marcados é uma aproximação grosseira da fronteira da malha. Todos os tetraedros que não pertencem à malha de deslocamento são descartados.

Alguns vértices marcados são exteriores à fronteira, e outros interiores, mas todos estão suficientemente próximos, como pode ser visto na figura 6. Quanto maior o deslocamento de um vértice para alcançar a fronteira, maior serão as forças criadas na malha. Forças de compressão muito altas causam perturbações na malha, podendo inclusive colapsar alguns elementos.

Para garantir uma compressão suave, dois casos particulares devem ser tratados. Primeiro, arestas internas da malha-d com os dois vértices incidentes marcados podem ser achatadas durante a compressão. Uma aresta interna, como mostrado na figura 7, é identificada quando todos os tetraedros do seu operador Estrela, *i.e.* os tetraedros incidentes na aresta, pertencem à malha-d. O segundo caso refere-se às arestas na fronteira da

malha-d, com mais de duas faces na fronteira incidente, ou seja, arestas que não são variedades.

Em um primeiro passo, todas as arestas que se enquadram em um desses casos são subdivididas, e a malha-d é computada novamente. Quando uma aresta interna é subdividida, o vértice criado é automaticamente inserido no envelope interior. Como a aresta está no interior da malha, é certo que o novo vértice também estará.

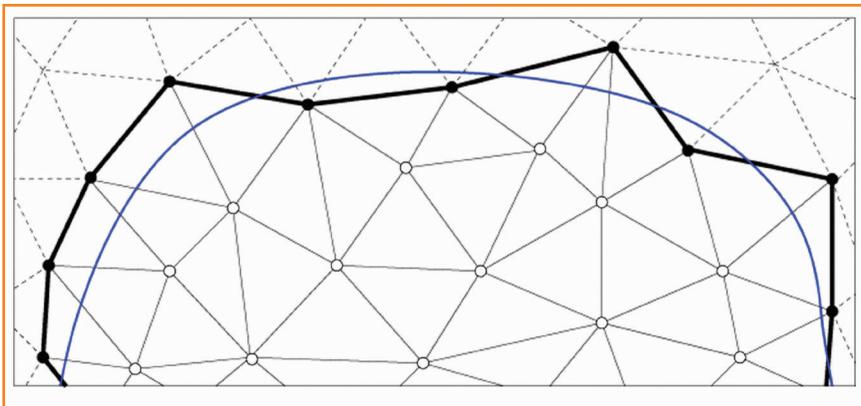
Em passadas subsequentes, a malha-d é novamente verificada. Cada vez que um dos dois casos é identificado, o vértice da aresta em questão com menor distância à fronteira da malha é inserido no envelope interior. Como a distância é negativa dentro da fronteira, se os dois vértices forem interiores, o mais interno é escolhido. Após cada passada, a malha-d é recomputada, e o processo repetido até que nenhum outro caso seja identificado.

Figura 6

Uma representação em duas dimensões da malha de deslocamento. Os vértices do envelope interior estão desenhados em branco. Em preto, os vértices da fronteira da malha de deslocamento ligados pela linha preta grossa. A curva representa a fronteira original do modelo. As linhas pontilhadas são as arestas descartadas da malha inicial.

Figure 6

A 2D representation of the displacement mesh. The inside envelope vertices have been drawn in white. The vertices of the boundary of the displacement mesh are in black and connected by the thick black line. The curve represents the original boundary of the model. The dotted lines represent the edges that have been eliminated from the initial mesh.



Interior Edges

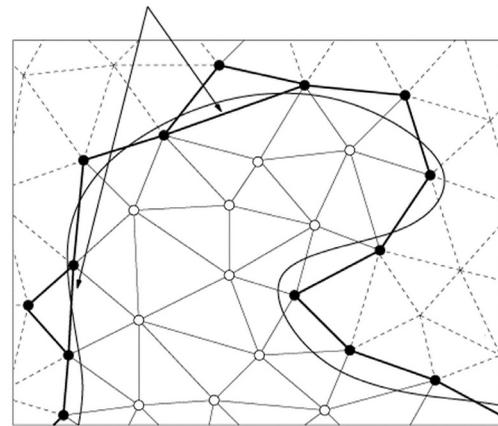


Figura 7 – As arestas apontadas representam as arestas interiores da malha-d conectando dois vértices marcados. A fronteira da malha-d está desenhada com segmentos grossos e a linha curva representa a fronteira do modelo. Para facilitar a visualização, a malha-d está representada em duas dimensões, mas o caso para três dimensões é análogo.

Figure 7 – The highlighted edges represent the interior edges of mesh-d connecting two marked vertices. Mesh-d boundary has been drawn with thick segments and the curve line represents the model boundary. Mesh-d has been represented in 2D to make it easier to distinguish it, but the 3D case is analogue to this one.

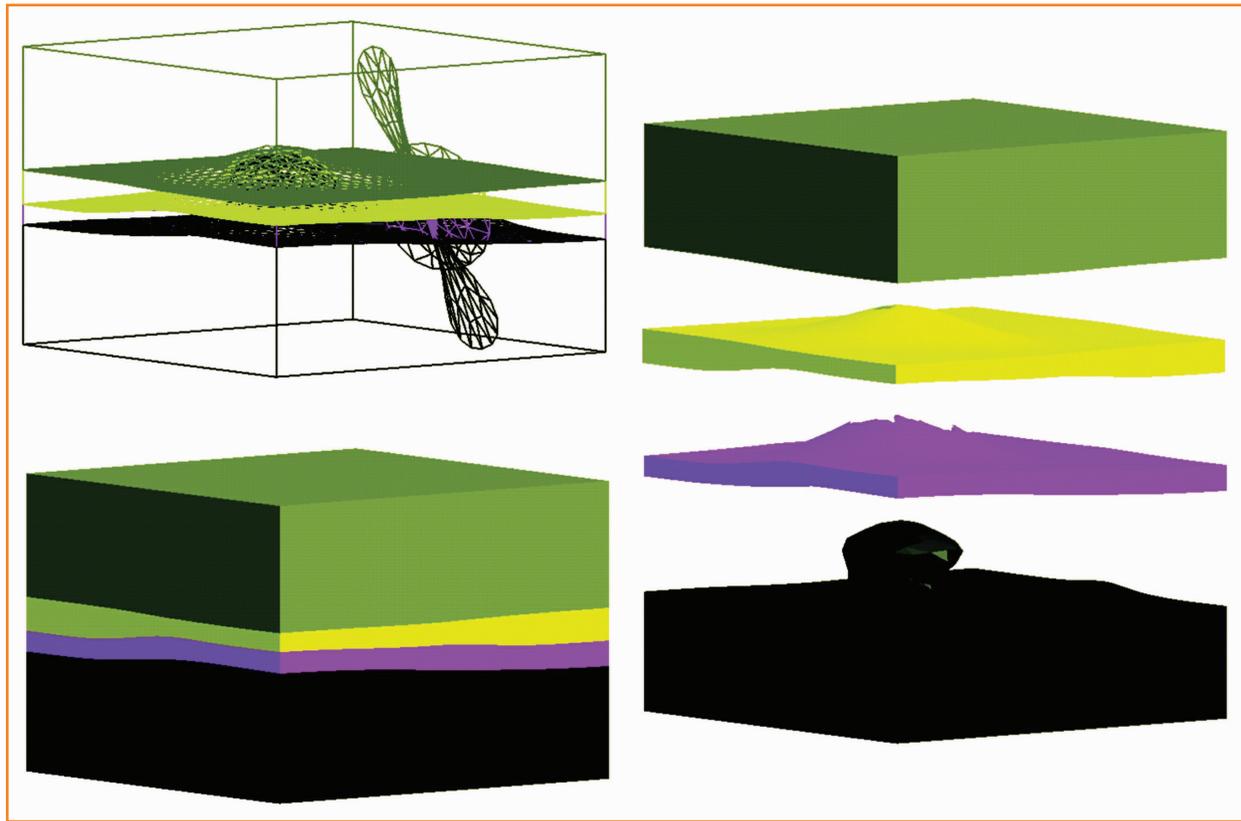


Figura 8

À esquerda, uma visão do modelo completo de um domo de sal. À direita, as regiões separadas.

Figure 8

To the left, a view of the complete model of a salt dome. To the right, separate regions.

modelos com multirregião

Ao se trabalhar com modelos com multirregião, envelopes interiores são criados separadamente para cada uma. Na figura 8 há exemplo de um modelo contendo mais de uma região. Como os envelopes interiores possuem apenas vértices internos das regiões, é garantido que não há intersecção entre envelopes diferentes. Porém, um tetraedro intersectando uma fronteira do modelo pode ser incidente em mais de um envelope interior e, ao se computar a malha-d, pode ter os quatro vértices marcados.

Para solucionar este problema, é preciso criar uma superfície delimitadora entre regiões adjacentes. Tal superfície deve conter os vértices marcados e, após a compressão, se constituirá na aproximação da fronteira entre as regiões.

Ao computar os envelopes interiores de cada região, as mesmas regras são aplicadas: um vértice deve ter todas as arestas incidentes, pelo menos, 25% no interior da fronteira para pertencer ao envelope. Porém, para encontrar a superfície delimitadora, a malha-d é computada somente para algumas regiões. Para isso, é utilizado um

critério simples baseado na identificação da região. Ao selecionar os vértices marcados na fronteira entre duas regiões, somente a região com menor número de identificação é considerada. Ao computar a superfície delimitadora da malha-d para uma região, ela é automaticamente definida para a região adjacente, como mostrado na figura 9.

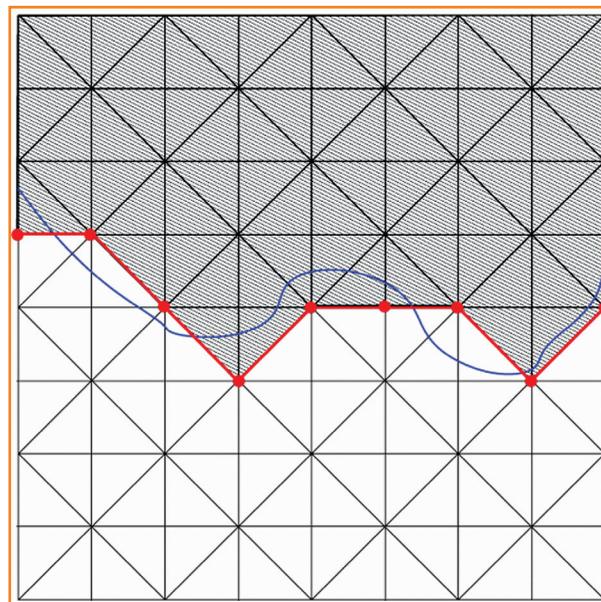


Figura 9

Uma visão em duas dimensões da superfície delimitadora (linha vermelha) entre as regiões branca e cinza. Os vértices na superfície delimitadora são os vértices marcados para serem empurrados em direção à fronteira, representada pela curva.

Figure 9

A 2D view of the boundary surface (red line) between the white and the gray region. The vertices found in the boundary surface have been marked to be pushed towards the boundary, which has been represented by the curve.

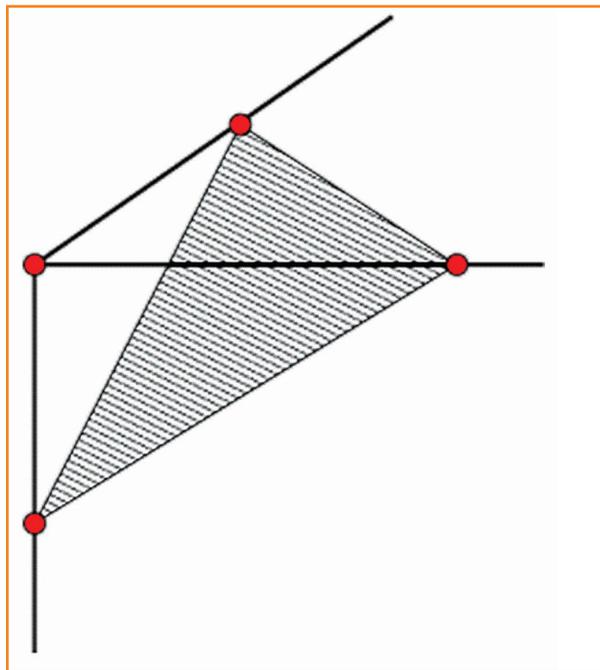
Em modelos com multirregiões, é importante identificar as faces na fronteira da malha de deslocamento para que as arestas internas sejam removidas. Estas arestas são facilmente identificadas como aquelas que possuem os dois vértices incidentes marcados e sem incidência de faces na fronteira. No entanto, não é trivial identificar uma face na fronteira da malha-d, ou seja, na região delimitadora entre duas regiões. Para uma face estar na fronteira, algumas condições devem ser atendidas. A primeira condição estabelece que todos os seus três vértices estejam também na fronteira da malha-d e, portanto, marcados para serem empurrados para a fronteira do modelo. Contudo, esta condição não é suficiente para garantir que a face esteja na fronteira, como pode ser visto na figura 10. Para garantir que uma face esteja realmente na fronteira, é preciso analisar a classificação dos dois tetraedros incidentes a ela.

Figura 10

Um canto de um cubo com uma face interior com os três vértices na fronteira.

Figure 10

A corner of a cube with one inside face with the three boundary vertices.



A classificação dos tetraedros é realizada de acordo com as regiões que contêm os seus vértices. O principal fato que permite classificar os tetraedros de acordo com a classificação de seus vértices é que, exceto aqueles marcados (na fronteira da malha-d), todos os remanescentes devem pertencer a uma mesma região. Na figura 9 nota-se que ao remover os vértices marcados, nenhum triângulo cinza possui vértices na

região branca e vice-versa. Esta mesma propriedade é válida para modelos em três dimensões. Portanto, um tetraedro pode ser classificado como pertencente à mesma região de qualquer um de seus vértices não marcados.

Após classificar os tetraedros, uma face de fronteira pode ser facilmente identificada quando seus dois tetraedros incidentes estão em regiões diferentes.

O esquema de classificação de tetraedros pode falhar para aqueles no exterior do modelo. Estes tetraedros podem ter todos os quatro vértices marcados e, portanto, nenhum vértice livre para associar a classificação. Embora seja uma condição que deva ser evitada, estes tetraedros não fazem parte da malha de deslocamento e, portanto, não são considerados na triangulação final. Na figura 11 há um exemplo de um tetraedro exterior com todos os vértices marcados. Para se evitar falhas no teste, sempre que este caso for encontrado, o tetraedro é classificado como exterior e, assim, não pertencente a região alguma.

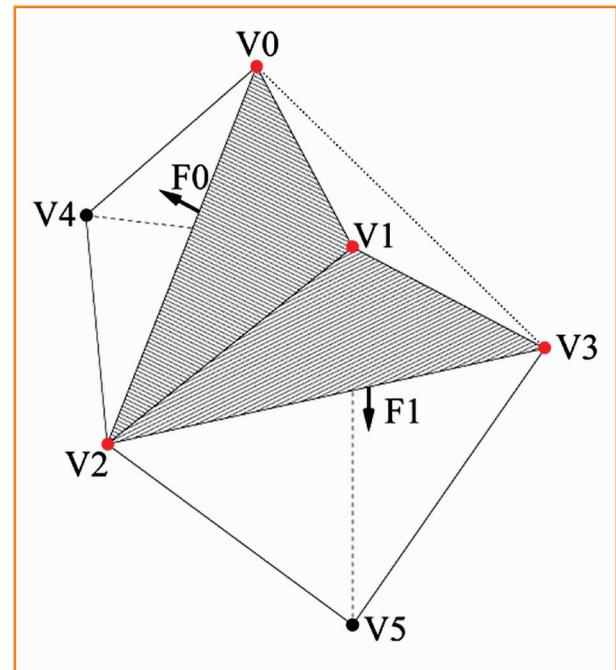


Figura 11 – Tetraedro exterior com os quatro vértices marcados: V_0 , V_1 , V_2 , e V_3 . As faces de fronteira são as hachuradas (F_0 e F_1) e as setas apontam para o interior do modelo.

Figure 11 – The outside tetrahedron with the four marked vertices: V_0 , V_1 , V_2 , and V_3 . The faces of the boundary have been shaded (F_0 and F_1) and the arrows point to the inside of the model.



sistema físico

O sistema físico escolhido para comprimir a malha é baseado em um esquema de massas e molas. Esta compressão realiza a aproximação da malha-d em relação à fronteira do modelo. Os elementos da malha-d são utilizados para representar as molas e partículas do sistema.

configuração das molas

Uma questão muito importante é como configurar as molas na malha tetraedral para se realizar a compressão. Molas mal posicionadas podem ser incapazes de manter a forma dos elementos ou pior: não impedir que sejam colapsados durante a compressão. Diversos esquemas diferentes são possíveis para configurar as molas (fig. 12).

É fácil perceber que uma configuração com molas, por exemplo, somente nas arestas dos triângulos não é suficiente para manter a forma dos elementos. Um vértice pode se deslocar, sem muita resistência, para o centro da sua aresta oposta colapsando o triângulo.

A configuração que apresenta melhores resultados consiste em uma mola por aresta, mais três molas de projeção por triângulo, como ilustrado na figura 12f. Cada mola de projeção é posicionada entre um vértice da malha e a sua projeção ortogonal na aresta oposta do triângulo. Para isto, é necessário criar alguns vértices auxiliares que não fazem parte da malha-d, desenhados em branco na figura 12. Esta configuração não só apresenta melhores resultados na prática, mas também possui prova teórica de que evita os colapsos com maior eficiência, como demonstrado por Cooper e Maddock (1997).

Em três dimensões, é utilizada uma configuração análoga àquela ilustrada na figura 12f. As arestas da malha referenciam suas molas correspondentes enquanto cada face referencia suas duas molas de projeção, como mostrado na figura 13.

Para computar as forças nas molas, duas equações distintas são utilizadas: uma para as molas de aresta, outra para as molas de pro-

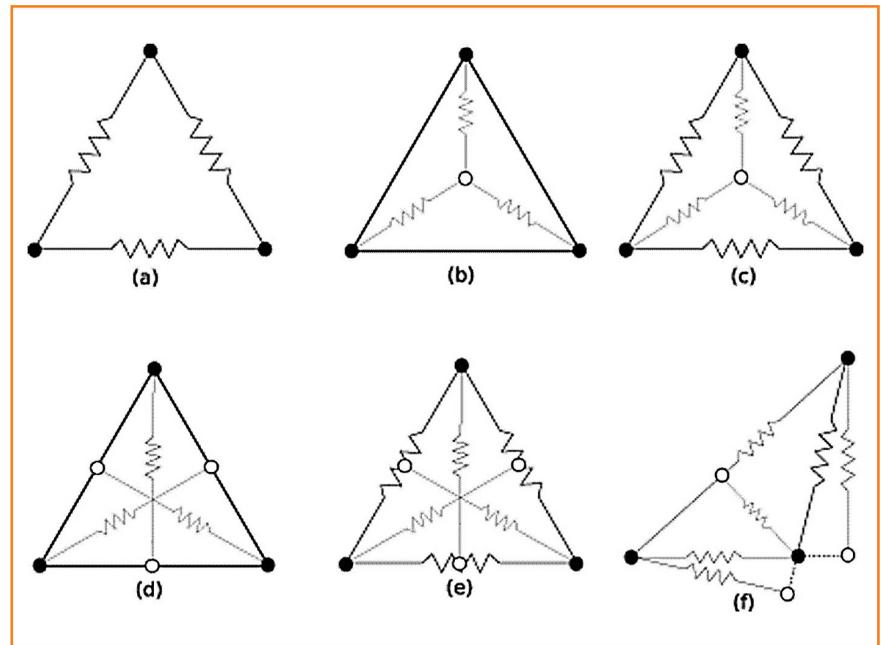


Figura 12 – Seis esquemas diferentes para configuração do sistema de massa-mola na malha:

- (a) somente molas nas arestas da malha;
- (b) somente molas dos vértices ao centróide do triângulo;
- (c) molas nas arestas e nos centróides;
- (d) somente molas nas medianas do triângulo;
- (e) molas nas arestas e nas medianas;
- (f) molas nas arestas e nas projeções dos vértices às arestas opostas.

Em preto, os vértices reais da malha, em branco os vértices virtuais.

Figure 12 – Six different schemes to setup the physical mass spring system:

- (a) only springs on the mesh edges;
- (b) only springs from the vertices to the centroid of the triangle;
- (c) springs on the edges and centroids;
- (d) only springs on the medians of the triangle;
- (e) springs on the edges and medians;
- (f) springs on the edges and on the projections of the vertices to the opposed edges.

Actual vertices of the mesh in black; virtual vertices in white.

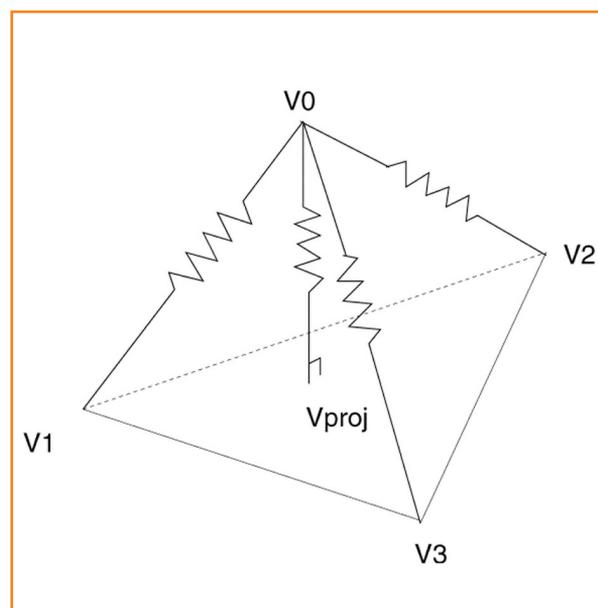


Figura 13

Molas incidentes no vértice v_0 : três molas de aresta e uma mola de projeção. As outras molas do tetraedro (três de projeção e três de aresta) não estão ilustradas nessa figura.

Figure 13

Springs on vertex v_0 : three edge springs and one projection spring. The other springs of the tetrahedron (three projection springs and three edge springs) have not been included in this figure.

jeção. A equação 1 se refere às molas de aresta e segue a lei de Hooke:

$$F = \left[k_s * (|\Delta x| - x_0) + kd * \left(\frac{\Delta v \cdot \Delta x}{|\Delta x|} \right) \right] * \frac{\Delta x}{|\Delta x|} \quad (1)$$

onde k_s e kd são os coeficientes de elasticidade e amortecimento, x_0 é o comprimento de descanso da mola e Δx e Δv são as diferenças de posição e velocidade dos vértices.

Para as molas de projeção é aplicada uma equação não linear, como descrito em Cooper e Maddock (1997) e mostrado na equação 2. Esta mola não linear exerce uma força infinita nos vértices quando seu comprimento tende a zero.

$$F = k_s * \left(|\Delta x| - \frac{x_0^2}{|\Delta x|} \right) * \frac{\Delta x}{|\Delta x|} \quad (2)$$

sistema de partículas

Para simular o movimento oscilatório do sistema massa-mola, a malha é tratada como um sistema de partículas (Witkin, 1994). Cada vértice representa uma partícula no sistema, e partículas são associadas às extremidades das molas. Algumas partículas temporárias podem ser introduzidas para representar os vértices virtuais. Tais partículas existem apenas ao computar as forças associadas às suas molas de projeção e, portanto, influenciam apenas localmente o sistema.

Cada partícula possui os seguintes atributos: posição, velocidade, força e massa. O sistema itera em passos de tempo, atualizando, a cada ciclo, a posição e a velocidade de cada partícula. A cada passo de tempo são computadas as forças nas partículas. A força em cada vértice é a soma de todas as forças geradas por suas molas incidentes.

O valor da massa é calculado diferentemente para cada partícula. Aquelas nas proximidades da fronteira do modelo são mais pesadas do que aquelas mais afastadas. Isto implica serem os tetraedros menores, perto da fronteira, sejam mais rígidos e menos suscetíveis a perder a forma original. Por outro lado, os tetraedros maiores, no interior do modelo, são mais leves e podem ser

rearranjados com mais facilidade. Desta forma, quando a resistência dos tetraedros interiores é muito grande, evita-se que aqueles na fronteira fiquem muito deformados, ou que seus vértices marcados não alcancem a fronteira. A massa de um tetraedro é computada como a razão entre seu nível de refinamento e o nível máximo de refinamento da subdivisão espacial. A massa de cada partícula é calculada como a média de todas as massas dos tetraedros incidentes.

A duração de um passo de tempo é definida pelo usuário. Valores pequenos são melhores, pois criam menos oscilações nas molas. Porém, valores pequenos também resultam em mais iterações e, conseqüentemente, em tempos de simulação maiores.

Cada iteração pode ser dividida em três fases distintas:

1. cálculo da força acumulada em cada vértice;
2. cálculo dos valores derivados;
3. cálculo das novas posições e velocidades.

acumuladores de forças

A força acumulada em cada vértice é a soma das contribuições de todas as molas incidentes. Vértices virtuais não possuem acumuladores, mas contribuem para a força acumulada no vértice real na outra extremidade da mola.

O primeiro passo em cada iteração de tempo é inicializar as forças de todas as partículas com valores nulos. Em seguida, todas as arestas e faces são iteradas para calcular os valores correspondentes nas suas partículas utilizando as equações 1 e 2, respectivamente. Para cada mola, uma das equações é computada e seu valor é somado ao acumulador de força de um vértice. Ao acumulador de força do vértice na outra extremidade, caso seja um vértice real, é somado o mesmo valor com sinal contrário. No caso de um vértice virtual, não é necessário acumular sua força incidente, já que sua posição é computada dinamicamente de acordo com a aresta de projeção.

valores derivados

Neste passo, as derivadas dos valores da posição e a velocidade de cada partícula são calcu-



ladas, ou seja, sua velocidade e aceleração, respectivamente.

Um vetor auxiliar é utilizado para armazenar tais valores. O tamanho do vetor é $6n$, onde n é o número de partículas no sistema. O vetor de derivadas é escalado para o tempo de uma iteração.

atualização das coordenadas e velocidades

Outro vetor é criado para armazenar os valores correntes das coordenadas e a velocidade de cada partícula. O tamanho do vetor é o mesmo do vetor de derivadas. Para calcular os valores das novas coordenadas e velocidades das partículas, os dois vetores descritos são somados.

critérios de compressão

Os comprimentos de repouso das molas são definidos como seus comprimentos iniciais, ou seja, inicialmente o sistema está completamente estável. Portanto, se nenhuma perturbação for introduzida no sistema, o estado não muda. Para iniciar a compressão, os vértices marcados são empurrados em direção à fronteira.

Para que a compressão comece, as partículas da fronteira da malha-d são projetadas na direção definida pela média das normais das faces de fronteira incidentes. Se o vértice estiver no exterior da fronteira do modelo, as normais apontando para dentro da fronteira são utilizadas. Caso contrário, se o vértice estiver no interior, as normais contrárias são utilizadas.

Para os vértices marcados, considera-se apenas a componente da força perpendicular à velocidade. Após cada passo de tempo, a velocidade é recomputada utilizando as normais das faces incidentes. Assim, os vértices marcados caminham em direção à fronteira enquanto se rearranjam no plano perpendicular à sua trajetória. Esta direção impede que vértices vizinhos cheguem muito próximos à fronteira. Se estes vértices interagissem livremente, as forças internas da malha, eventualmente, os retornaria a seus estados iniciais.

limitando a compressão

Para melhor controlar a malha-d durante a compressão, algumas restrições são impostas ao movimento das partículas. Tais restrições são motivadas pelas técnicas utilizadas em simulação de tecidos (Bridson *et al.* 2002; Baraff e Witkin, 1998; Provot, 1995). Estas técnicas limitam a compressão máxima das molas e a taxa de compressão por passo de tempo. Primeiro, nenhuma mola pode sofrer uma compressão maior do que 60% do seu comprimento de repouso. Segundo, durante um passo de tempo, nenhuma partícula pode se movimentar de tal sorte que alguma de suas molas incidentes sofra uma compressão maior do que 10% da sua altura corrente. Quando um vértice infringe uma dessas condições, ele é congelado durante o passo de tempo atual. Todos os vértices que não foram parados têm suas posições e velocidades recomputadas iterativamente como em um passo de tempo comum. Tal procedimento é repetido até que nenhuma partícula precise ser congelada. Estes limites artificiais geram uma robustez maior durante a compressão da malha e ajudam a obter uma triangulação com maior qualidade.

Os vértices marcados, que estão sendo empurrados em direção à fronteira, também podem ser congelados quando infringem uma das condições descritas. Em alguns casos, podem ser continuamente parados de forma que nunca cheguem até a fronteira. Isto acontece mais comumente em malhas de baixa resolução, onde os vértices devem caminhar distâncias maiores e a variação de tamanho – entre os elementos perto da fronteira e aqueles no interior – é pequena. Este procedimento é válido somente onde uma justaposição da fronteira do modelo não é necessária. A simulação termina quando todos os vértices marcados chegam na fronteira ou estão congelados.

resultados

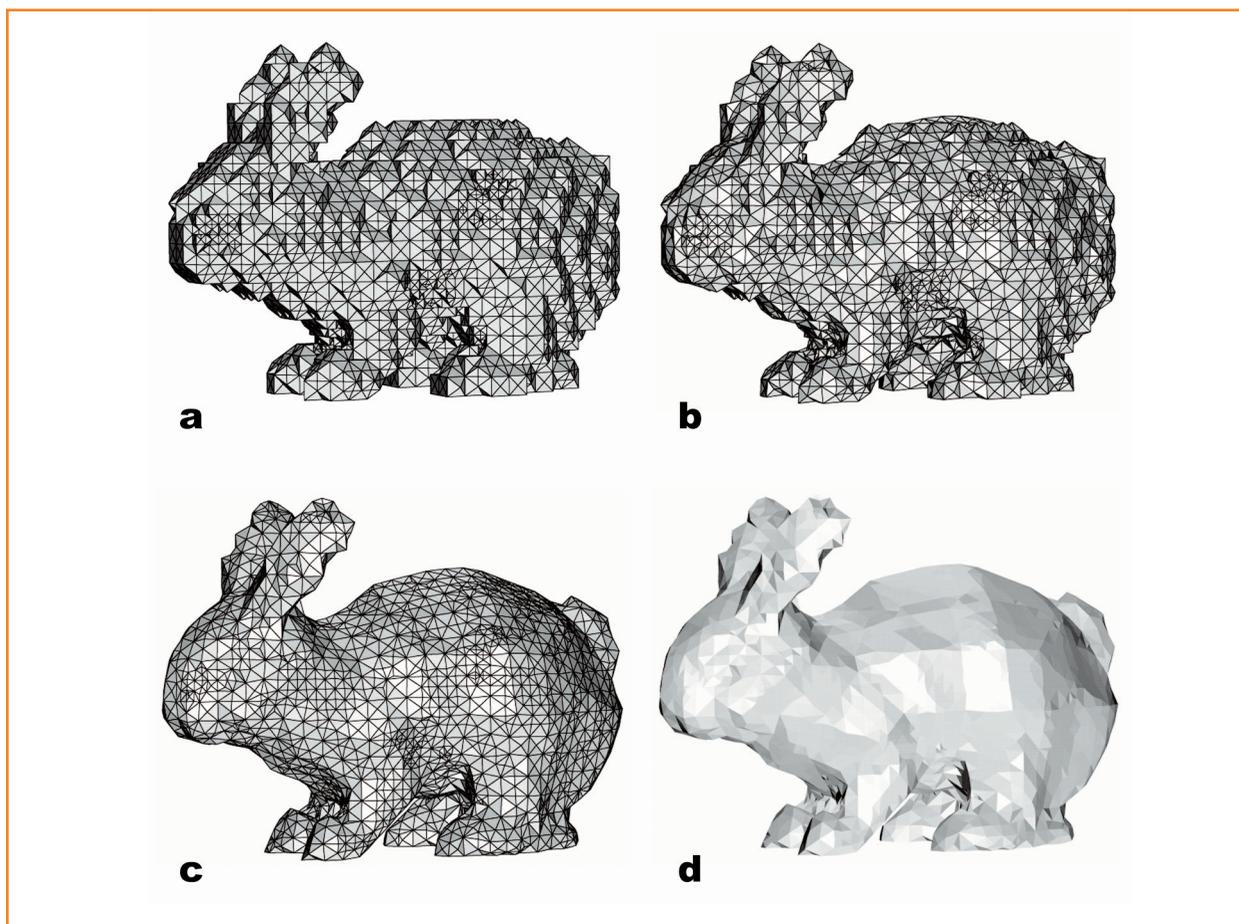
Na figura 14 é exposta uma aproximação grosseira do modelo do coelho de Stanford. Esta malha contém 30 K tetraedros, com ângulo diédrico mínimo de aproximadamente $3,5^\circ$, onde 99,4% dos ângulos estão acima de 18° .

Figura 14

Aproximação grosseira do coelho de Stanford com 30 K tetraedros. Na seqüência, a malha-d antes (a), durante (b) e ao final com (c) e sem arestas (d) da compressão.

Figure 14

Gross approximation of Stanford Rabbit with 30 K tetrahedrons. The sequence of mesh-d before (a), during (b) and after c(c) ompression (with and without edges)



A triangulação do tigre, na figura 15, foi gerada com 160 K tetraedros e ângulo diédrico mínimo de $4,6^\circ$. Na figura 16 é exposto o interior do tigre. Na figura 17 são mostrados 21 níveis de resolução diferentes para o modelo.

Figura 15

Triangulação do tigre com aproximadamente 160 K tetraedros.

Figure 15

Tiger triangulation with approximately 160 K tetrahedrons.

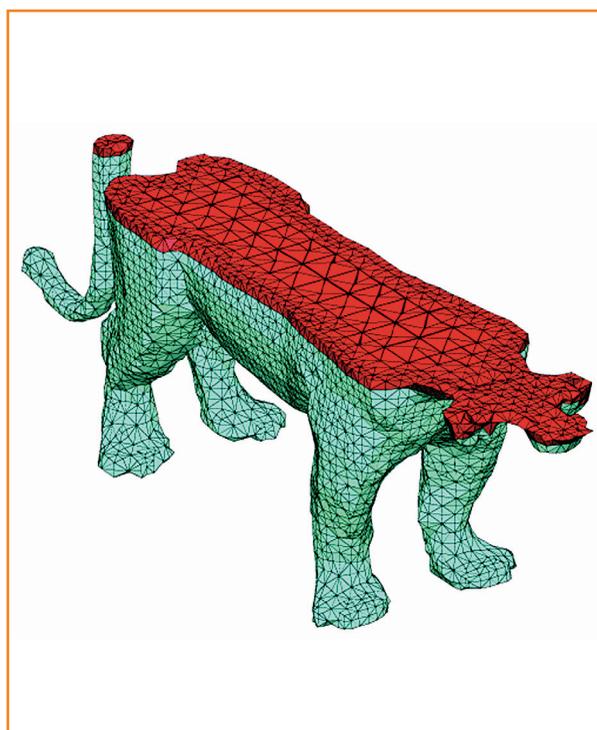


Figura 16 – Um corte do modelo do tigre.

Figure 16 – A cut view of the tiger model.

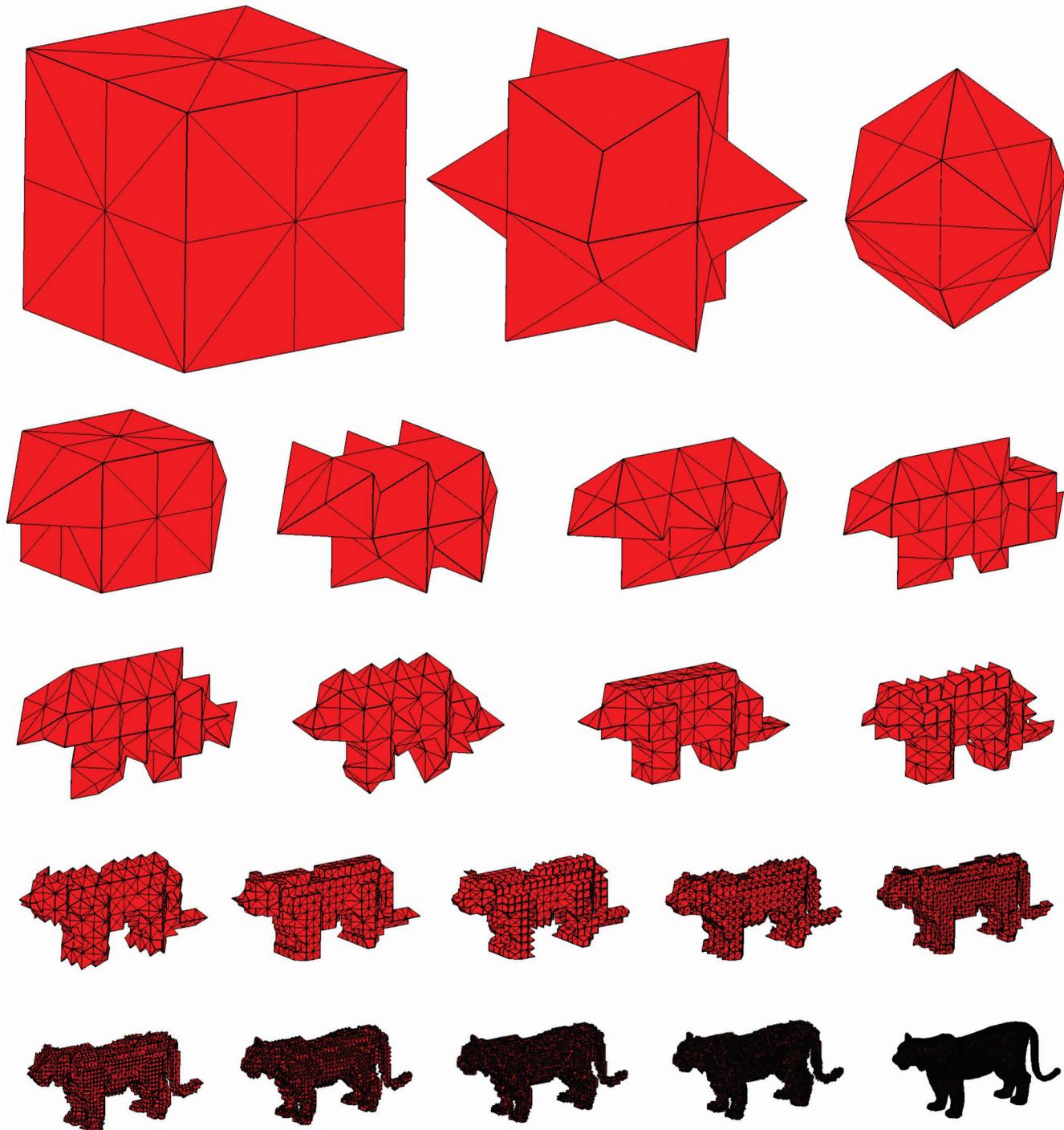


Figura 17 – Modelo do tigre exposto em diversas resoluções consecutivas.

Figure 17 – Tiger model exposed in different consecutive resolutions.

Na figura 18, três visões (duas com arestas e uma sem) de um modelo de uma cabeça são mostradas, e também sua respectiva malhas-d. Esta malha contém 242 K tetraedros, seu ângulo mínimo é de $3,1^\circ$ e 99% dos ângulos estão acima dos 25° . O tempo de execução foi de aproximadamente uma hora.

Dois cortes da malha do domo de sal são mostrados na figura 19. A triangulação final contém aproximadamente 900 K tetraedros, ângulo mínimo de $4,9^\circ$ e com 99,7% dos ângulos acima de 18° .

Figura 18

Três visões da malha final (a, b, c) e uma da malha-d (d).

Figure 18

Three views of the final mesh and one view of mesh-d.

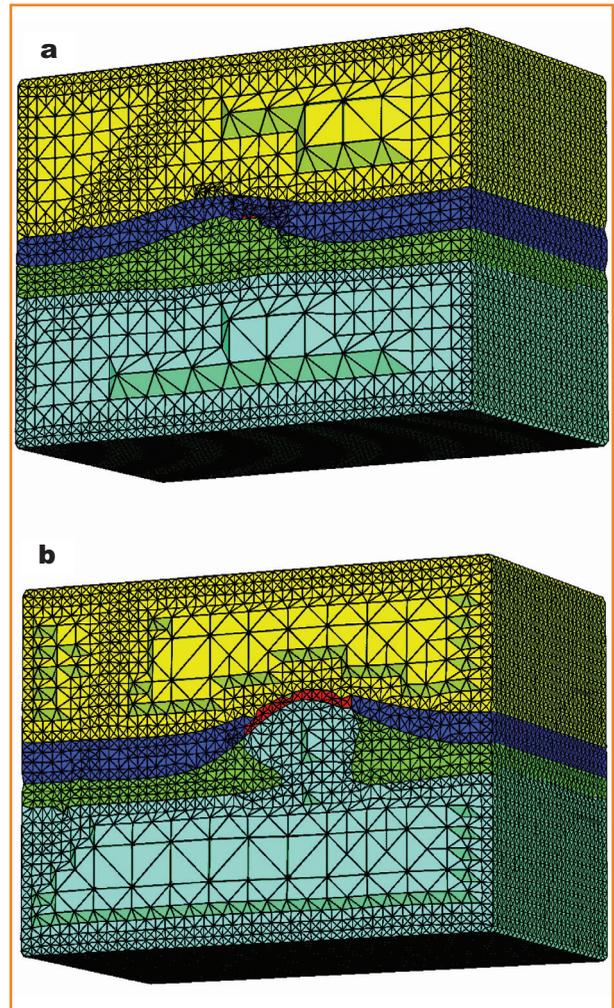
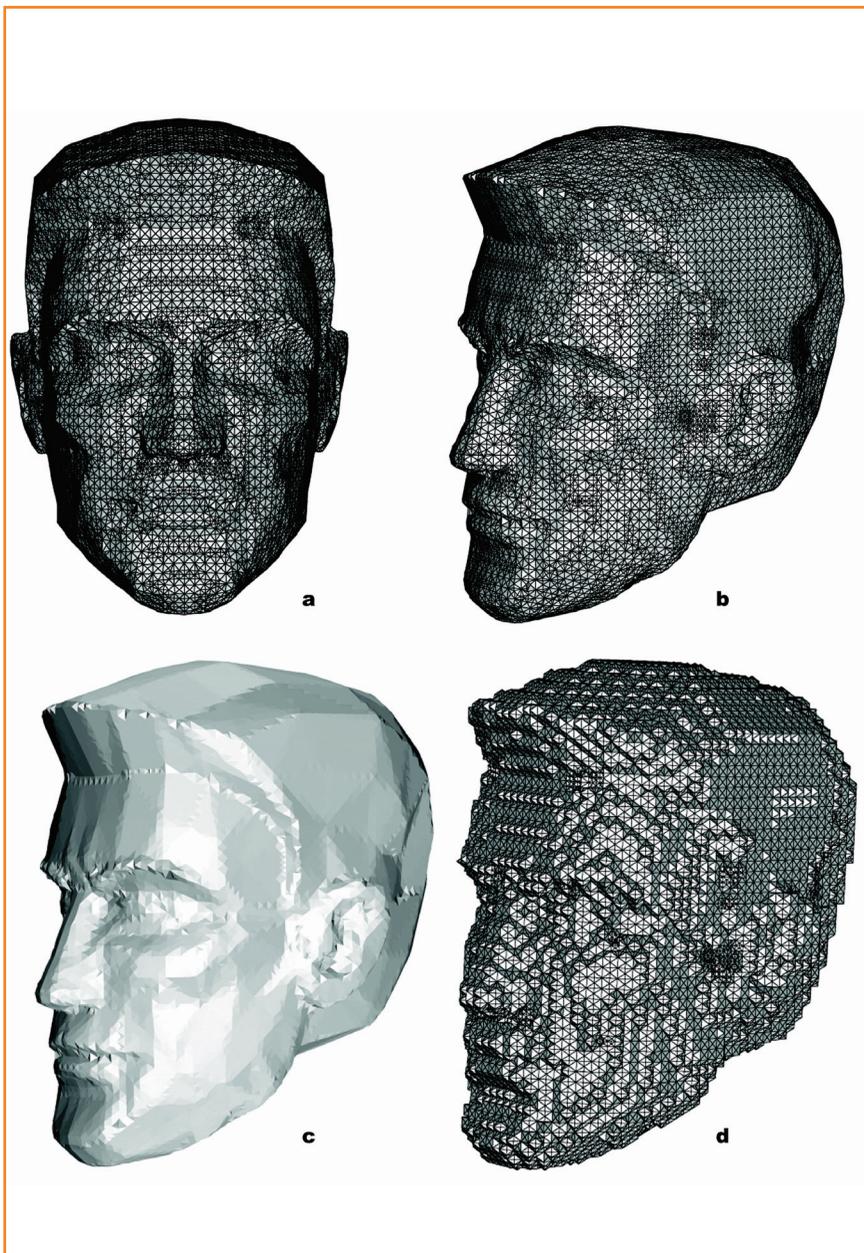


Figura 19 – Dois cortes da malha do domo de sal.

Figure 19 – Two cut views of the salt dome mesh.

conclusões

Foi apresentado aqui um algoritmo para triangulação de modelos em três dimensões com bordas complexas e irregulares. O algoritmo é capaz de gerar a malha em diversas resoluções e não produz elementos de área ou volume zero. Estas são as principais vantagens oferecidas como extensão dos algoritmos de triangulação tradicionais.

Uma das maiores diferenças dos modelos convencionais reside no fato de não haver uma fidelidade quanto à fronteira da malha gerada em relação à fronteira original do modelo. Apesar desta propriedade ser importante em vários casos, existem aplicações onde uma aproximação é acei-



tável, como, por exemplo, em modelos geológicos. Estes modelos provêm de interpretações humanas de dados sísmicos e, por esta razão, estão sujeitos a erros. Assim, pequenos desvios nos detalhes da malha não provocam grandes problemas na sua utilização. Além disso, modelos geológicos são utilizados em simulações numéricas nas quais a qualidade dos elementos da malha é de maior importância do que a fidelidade à fronteira. Elementos achatados e ângulos muito pequenos sempre causam problemas numéricos.

Alguns algoritmos de triangulação tradicionais obtiveram resultados expressivos com provas teóricas de ângulo mínimo e com garantia de término, especialmente em duas dimensões.

O algoritmo proposto também oferece um esquema de multirresolução sem esforço adicional. A malha, após ter sido gerada, pode ser visualizada em qualquer resolução. Isto se deve não só ao fato da estrutura utilizada para gerar a grade já possuir a propriedade de multirresolução como também ao fato da topologia da malha não ser alterada durante todo processo.

trabalhos futuros

Um esquema de subdivisão espacial baseado na curvatura da fronteira, e não na distância da mesma, é de grande interesse. A malha conterà, desta forma, tetraedros menores em locais onde a fronteira for mais detalhada e complexa. Nas regiões onde a fronteira é mais homogênea, é possível diminuir a quantidade de elementos.

Outra proposta é a criação de uma métrica para avaliar a aproximação da malha em relação à fronteira original. Com isto, é possível saber quais malhas apresentaram uma aproximação melhor e onde ocorrem os maiores desvios.

O problema do serrilhamento é outro ponto a ser melhorado. Refinar mais a malha onde ocorre o serrilhamento pode melhorar a aproximação, mas ainda não elimina o problema por completo.

Uma investigação mais detalhada para encontrar valores ideais para as constantes elásticas e de amortecimento das molas pode melhorar os resultados gerados. Os testes mostram que variações nas constantes melhoram os ângulos míni-

mos gerados em alguns modelos. Para um mesmo modelo de uma esfera, por exemplo, foi possível elevar o ângulo mínimo de 19° para 41°, apenas aumentando os valores das constantes, tornando as molas mais flexíveis. Porém, esta relação ainda não está clara.

referências bibliográficas

BARAFF, D.; WITKIN, A. Large steps in cloth simulation. In: ANNUAL CONFERENCE ON COMPUTER GRAPHICS AND INTERACTIVE TECHNIQUES, 25., 1998, Orlando. **Proceedings...** New York: ACM Press, 1998. p. 43-54.

BERN, M.; EPPSTEIN, D. Quadrilateral meshing by circle packing. **International Journal of Computational Geometry and Applications**, Singapore, v. 10, n. 4, p. 347-360, 2000.

BRIDSON, R.; FEDKIW, R.; ANDERSON, J. Robust treatment of collisions, contact and friction for cloth animation. In: ANNUAL CONFERENCE ON COMPUTER GRAPHICS AND INTERACTIVE TECHNIQUES, 29., 2002, San Antonio, Texas. **Proceedings...** New York: ACM Press, 2002. p. 594-603.

CAVALCANTI, P. R.; CARVALHO, P. C. P.; MARTHA, L. F. Nonmanifold modeling: an approach based on spatial subdivisions. **Computer-Aided Design**, Guildford, v. 29, n. 3, p. 209-220, Mar. 1997.

CHENG, S. W.; DEY, T. K.; EDELSBRUNNER, H.; FACELLO, M. A.; TENG, S. H. Silver exudation. In: ANNUAL SYMPOSIUM ON COMPUTATIONAL GEOMETRY, 15., 1999, Miami Beach. **Proceedings...** New York: ACM Press, 1999. p. 1-13.

COHEN-STEINER, D.; VERDIÈRE, E. C. de; YVINEC, M. Conforming delaunay triangulations in 3D. In: SYMPOSIUM ON COMPUTATIONAL GEOMETRY, 18., 2002, Barcelona. **Proceedings...** New York: ACM Press, 2002. p. 199-208.

COOPER, L.; MADDOCK, S. Preventing collapse within mass-spring-damper models of deformable objects. In: INTERNATIONAL CONFERENCE IN CENTRAL EUROPE

- ON COMPUTER GRAPHICS AND VISUALIZATION, 5., 1997, Plzen-Bory, Czech Republic. **Proceedings...** Holoubkov: Vikomt, 1997. p. 70-78.
- EDELSBRUNNER, H.; GUOY, D. An experimental study of sliver exudation. **Engineering with Computers**, New York, v. 18, n. 3, p. 229-240, Oct. 2002.
- FLEISCHMANN, P.; SELBERHERR, S. Three-dimensional delaunay mesh generation using a modified advancing front approach. In: INTERNATIONAL MESHING ROUNDTABLE, 6., 1997, Park City, Utah. **Proceedings...** New Mexico: Sandia National Laboratories, 1997. p. 267-278.
- FREY, P. J.; BOROUCAKI, H.; GEORGE, P. L. Delaunay tetrahedralization using an advancing-front approach. In: INTERNATIONAL MESHING ROUNDTABLE, 5., 1996, Pittsburgh, Pennsylvania. **Proceedings...** New Mexico: Sandia National Laboratories, 1996. p. 21-46.
- GEORGE, P. L.; BOROUCAKI, H. Delaunay triangulation and meshing: application to finite elements. Paris: Hermes, 1998. 413 p.
- LI, X. Y.; TENG, S. H.; ÜNGÖR, A. Biting spheres in 3D. In: INTERNATIONAL MESHING ROUNDTABLE, 8., 1999, South Lake Tahoe, California. **Proceedings...** New Mexico: Sandia National Laboratories, 1999. p. 85-95.
- LÖHNER, R. Progress in grid generation via the advancing front technique. **Engineering with Computers**, New York, v. 12, n. 3/4, p. 186-210, Sep. 1996.
- LÖHNER, R.; PARIKH, P. Three-dimensional grid generation by the advancing front method. **International Journal for Numerical Methods in Fluids**, Chichester, v. 8, p. 1135-1149, 1988.
- MAVRILIS, D. J. An advancing front Delaunay triangulation algorithm designed for robustness. **Journal of Computational Physics**, New York, v. 117, n. 1, p. 90-101, 1995.
- MELLO, V.; VELHO, L.; CAVALCANTI, P. R.; SILVA, C. Bmt: a generic programming approach to multiresolution spatial decompositions. In: HEGE, H.; POLTHIER, K. (Ed.). **Visualization and Mathematics III**. Berlin: Springer-Verlag, 2003. p. 337-360.
- MOLINO, N.; BRIDSON, R.; TERAN, J.; FEDKIW, R. A crystalline, red green strategy for meshing highly deformable objects with tetrahedra. In: INTERNATIONAL MESHING ROUNDTABLE, 12., 2003, Santa Fe, New Mexico. **Proceedings...** New Mexico: Sandia National Laboratories, 2003. p. 103-114.
- MÖLLER, P.; HANSBO, P. On advancing front mesh generation in three dimensions. **International Journal for Numerical Methods in Engineering**, Chichester, v. 38, p. 3551-3569, 1995.
- MURPHY, M.; MOUNT, D. M.; GABLE, C. W. A point-placement strategy for conforming delaunay tetrahedralization. **International Journal of Computational Geometry and Applications**, Singapore, v. 11, n. 6, p. 669-682, 2001.
- PERAIRE, J.; PEIRÓ, J.; MORGAN, K. Advancing front grid generation. In: THOMPSON, J. F.; SONI, B. K.; WEATHERILL, N. P. (Ed.). **Handbook of grid generation**. Boca Raton: CRC Press, 1999. Chapter 17.
- PROVOT, X. Deformation constraints in a mass-spring model to describe rigid cloth behavior. In: GRAPHICS INTERFACE '95, 1995, Quebec. **Proceedings...** Ontario: Canadian Human-Computer Communications Society, 1995. p. 147-154.
- RUPPERT, J. **Results on triangulation and high quality mesh generation**. Berkeley: University of California at Berkeley, 1992. 135 p. Thesis (Ph. D.).
- RUPPERT, J. Delaunay refinement algorithm for quality 2-dimensional mesh generation. **Journal of Algorithms**, San Diego, v. 18, n. 3, p. 548-585, 1995.
- SHEWCHUK, J. R. Tetrahedral mesh generation by delaunay refinement. In: ANNUAL SYMPOSIUM ON COMPUTATIONAL GEOMETRY, 14., 1998, Minneapolis, Minnesota. **Proceedings...** New York: ACM Press, 1998. p. 86-95.
- SHEWCHUK, J. R. Delaunay refinement algorithms for triangular mesh generation. **Computational Geometry**:



theory and application, Amsterdam, v. 22, n. 1/3, p. 21-74, May 2002.

SHEWCHUK, J. R. **Delaunay refinement mesh generation**. Pittsburgh: Carnegie Mellon University. Department of Computer Science, 1997. Thesis (Ph. D.)

WEATHERILL, N. P.; HASSAN, O. Efficient three-dimensional delaunay triangulation with automatic point creation and imposed boundary constraints. **International Journal for Numerical Methods in Engineering**, Chichester, v. 37, p. 2005-2039, 1994.

WEILER, K. The radial-edge structure: a topological representation for nonmanifold geometric boundary representations. In: INTERNATIONAL FEDERATION FOR INFORMATION PROCESSING WORKING GROUP 5.2 WORKING CONFERENCE, 1986, Rensselaerville, New York. **Geometric modeling for CAD applications**: selected and expanded papers. Hardbound: Elsevier, 1988. p. 3-36.

WITKIN, A. **An introduction to physically based modeling**: particle system dynamics. New York: Rochester Institute of Technology, 1994. (Association for Computing Machinery's Special Interest Group on Graphics and Interactive Techniques. Course Notes, 32).

expanded abstract

Numerical simulations have become an essential step in the development of engineering products, or in predicting the behavior of physical phenomena, such as weather conditions, oil generation and migration, tide movement, earthquakes, etc.

In order to be able to carry out any simulation it is necessary to discretize the domain of interest, onto which a set of equations that describe physical laws must be solved. This discretization process is generically referred to as the triangulation of the domain.

Although several triangulation algorithms have been proposed in the past three decades, these algorithms are meant mainly to deal with mechanical parts, produced by computer aided design (CAD) systems. However, there are numerous important applications in which the domain presents no symmetry at all, and boundaries are not well or clearly established.

Geological models for instance are often composed by several regions (and form geological layers), and have an irregular geometry, with surfaces that meet at very small angles, a situation known as a pinch-out. The geometry of the surfaces (horizons and faults) is obtained from seismic data, by a process that is not automatic and requires a geologist's interpretation. As a consequence, the whole model creation process not exact, and tends to be error prone.

Traditional triangulation algorithms can be classified according to the way they work. Delaunay based algorithms recover the boundaries after triangulating the convex hull of the pointset (Cavalcanti et al. 1997). Ruppert and Shewchuck (Ruppert, 1992; Ruppert, 1995; Shewchuck, 1997) insert a vertex at the circumcenter of a bad element to improve the mesh quality.

On the other hand, advancing front algorithms (Mavriplis, 1995) start at the boundaries and proceed toward the center of the model, thus inserting new vertices to create good shaped elements. The trick is to merge the different fronts without generating any inconsistency. The success of the algorithm depends greatly on the simplicity of the boundary geometry. Also, as cited by George and Borouchaki (1999), there is no theoretical guarantee about the convergence of this process.

Packing sphere algorithms (Bern, 2000; Li et al. 1999) generate a vertices distribution by forcing edges and faces to be present in the final mesh. The latter can be then generated by a Delaunay-like algorithm.

Efforts were also made to join some of these techniques. Frey et al. (1996) proposed a method that

joined some advantages of the Delaunay approach with the advancing-front point placement strategy generating good quality meshes.

Although many of these traditional triangulation algorithms have achieved good results in many cases, it is difficult for them to deal with models having complex and irregular boundaries that contain small angles. Moreover, they do not support multi-resolution, i.e. the ability to generate meshes in different levels of detail for the same model. Even when a triangulation algorithm does a good job meshing this kind of model, it may generate small angles in the final mesh, which usually cause problems in running numerical simulations.

Furthermore, Shewchuk (1998) states that for any angle bound larger than zero, there is a PSLG (planar straight line graph) that cannot be triangulated without creating a new angle smaller than the bound, and this holds true for every triangulation algorithm.

This paper presents an adaptive algorithm to triangulate non-manifold, multi-region models, which approximates their irregular boundaries instead of trying to match them exactly. Although the boundary is just an approximation, our main concern is to generate a good mesh, as close as possible to the original one. Some features, like multi-resolution can only be implemented by loosening the boundary preservation. The algorithm is able to produce meshes in different levels of the details for the same model, that is, models that may contain more or less elements. Thus small input angles can be eliminated. The ability to triangulate multi-region models is also important as most target geological models contain more than one region defined by geological layers.

Physically based triangulation algorithms are not new. Molino (2003) has used a similar approach to obtain numerical meshes before. Nonetheless, despite his good results, he considered only single region manifold objects. In this paper we expand Molino's work to be able to deal with non-manifold multi-region objects, which are necessary to address a broader set of applications.

The algorithm has two main steps. In the first step, a grid is adaptively created over the object in order to subdivide the surrounding space. The adaptive property of this mesh can be enhanced based on a given criterion. The one chosen here is the distance from a tetrahedron to a boundary surface, thus producing finer resolution near the boundary, and coarser on the inside. The representation of the input data has been created using the complete geometric complex CGC



system (Cavalcanti et al. 1997), and the target is geological applications, which define a set of horizons and faults. Faults represent discontinuities (cracks) in the subsurface of the earth, and divide the model into a set of regions. A group of regions with similar attributes forms a geological layer. The geometry of geological models is intrinsically complex because surfaces have been folded many times during their formation process. A model representing a sedimentary basin may extend from several hundred kilometers along x and y , to just a few hundred meters in z . A layer near the boundary may be only 30 m deep.

The second step shapes the vertices of the grid to the boundary of the object using a compression scheme, boundary of the object using a compression scheme, while trying to maintain well shaped elements. In order to assure that the enhanced mesh is in conformity with the model boundary, a physical mass spring system is used. The idea is to select a subset of the mesh and compress it to the boundary of the model. Springs are placed on each tetrahedron to preserve their shape during compression. In order to simulate the mass-spring oscillatory movement, the mesh is treated as a particle system. The algorithm has several advantages over traditional triangulation techniques, such as:

- 1) it is easy to implement and offers multi-resolution at the same time;
- 2) it can deal with models that have complex and irregular boundaries; and
- 3) it is not overly affected by numerical issues.

autor principal



Ricardo Marroquim

Universidade Federal do Rio de Janeiro (UFRJ)

Laboratório de Computação Gráfica – COPPE

e-mail: ricardo@lcg.ufrj.br

Ricardo Marroquim concluiu o Mestrado em engenharia de sistemas e computação pela Universidade Federal do Rio de Janeiro (UFRJ) em 2005, onde cursa Doutorado desde então. Recebeu o 2º lugar na categoria Tecnologia de Exploração da 1ª edição do Prêmio Petrobras de Tecnologia em 2005. Na área de computação gráfica, seus interesses são: malhas tetraedrais, multirresolução, sistema baseado em física e visualização volumétrica.