Geração auto-adaptativa de malhas com sistemas de partículas

LUIZ HENRIQUE DE FIGUEIREDO[1,2] MARCELO TILIO MONTEIRO DE CARVALHO[2]

[1]Computer Systems Group, Department of Computer Science, University of Waterloo
Waterloo, Ontario, Canadá, N2L 3G1 (lhf@csg.uwaterloo.ca)

[2]TeCGraf–Grupo de Tecnologia em Computação Gráfica, Departamento de Informática, PUC-Rio
Rua Marquês de São Vicente 225, 22453-900 Rio de Janeiro, RJ, Brasil (tilio@icad.puc-rio.br)

Abstract. We investigate the use of bubble meshing for automatic mesh generation in polygonal domains in both 2D and 3D. Bubble meshing is a recent technique for mesh generation that uses particle system for finding good interior points and constrained Delaunay triangulations for building meshes. The main goals in our research are faster particle simulation, linear-time triangulation after equilibrium, and controlling particle density using error estimates in the finite-element analysis.

1 Introdução

A maioria dos problemas complexos em engenharia e matemática computacional não têm soluções analíticas, e são resolvidos através de simulações numéricas dos eventos físicos. A técnica mais utilizada para fazer essas simulações numéricas é o método dos elementos finitos [1], devido aos bons resultados obtidos e à sua aplicação em uma grande variedade de problemas.

O método dos elementos finitos requer a discretização do domínio do problema em *malhas*, isto é, uma decomposição do domínio em pequenos elementos de formas simples, como quadriláteros ou tetraedros. Essas malhas de elementos finitos tornam viável a simulação computacional das propriedades físicas de objetos contínuos.

A geração da malha é um dos passos da simulação que consome mais tempo e esforço. Para problemas simples, é possível gerar malhas manualmente. Entretanto, com o aumento da capacidade de processamento dos computadores, os problemas tratáveis atualmente envolvem objetos de geometria complexa, para os quais é tedioso, senão impossível, gerar malhas manualmente. Portanto, é desejável ter métodos automáticos para geração de malhas a partir de uma descrição computacional da geometria de um objeto, tipicamente com polígonos e poliedros.

Durante uma simulação, é frequentemente necessário refinar a malha inicial, para obter resultados mais precisos. Uma maneira natural de fazer isso é levar em conta uma estimativa do erro cometido na análise. Se os resultados da análise não forem satisfatórios, então a malha é refinada, baseada na estimativa de erro, e o ciclo discretização–análise–refinamento é repetido. Neste tipo de *análise adaptativa* é ainda mais importante que a geração e o refinamento da malha sejam automáticos; quando isso ocorre, a análise é chamada *auto-adaptativa* [2].

"Bubble meshing" é uma técnica recente para geração automática de malhas triangulares em domínios de geometria complexa, que usa sistemas de partículas para escolha dos nós da malha [3]. A motivação para esse método é a observação de que esferas ("bolhas") densamente compactadas em uma região formam padrões geométricos regulares, dos quais é possível extrair boas malhas triangulares ligando os centros de esferas vizinhas (Figura 1).

Os resultados obtidos com "bubble meshing" são animadores, mas o método, como descrito originalmente [3], é bastante lento. Estamos investigando mecanismos para acelerar este processo, através de técnicas mais eficientes para a simulação de partículas e para a geração da malhas após o equilíbrio. O objetivo final é ter um método eficiente e robusto para geração auto-adaptativa de malhas em sólidos tri-dimensionais. A investigação está sendo feito inicialmente para malhas bi-dimensionais, por dois motivos: facilidade de visualização, e existência de bons métodos em 2D, que podem servir para comparação.

2 Bubble meshing

A geração de malhas com "bubble meshing" é feita em dois passos: compactação de esferas (bolhas) no domínio, e geração da malha ligando os centros de esferas vizinhas. Ambos os passos são computacionalmente custosos.

A compactação é feita através de uma simulação pseudo-física de um sistema de partículas. As partículas são os centros das bolhas, que se atraem e se repelem. Embora as forças de atração e repulsão tenham raio de ação limitado, em princípio cada partícula tem que interagir com todas as outras, em cada passo da simulação. Assim, se há n partículas, é necessário calcular $\Theta(n^2)$ interações, o que torna a simulação lenta, não só porque n varia durante a simulação (partículas são criadas para preencher os vazios da região), mas principalmente porque

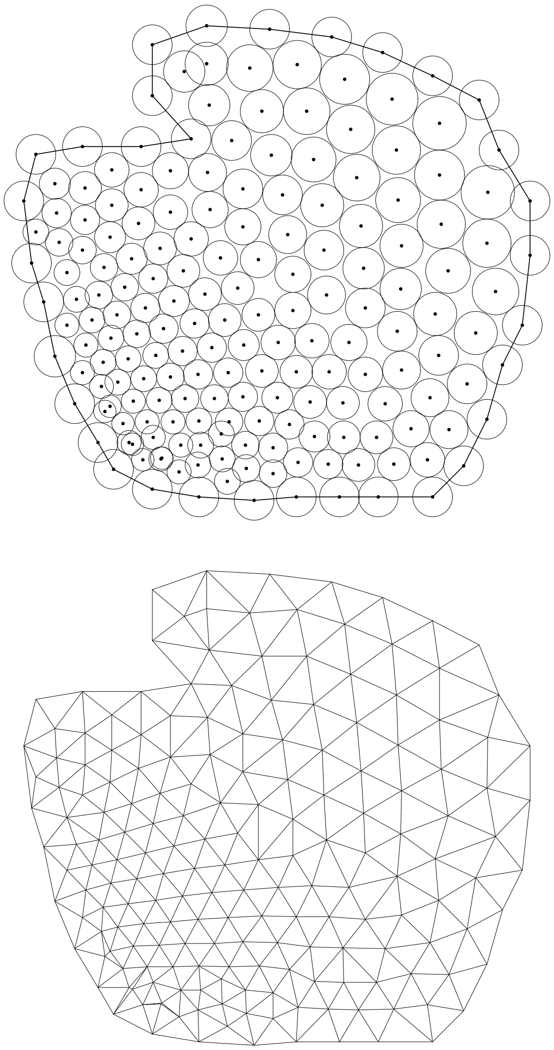

Figura 1: “Bubble meshing”: Posição das bolhas após equilíbrio e malha gerada.

vários passos são necessários até atingir um equilíbrio. Como é possível variar os raios das esferas ao longo da simulação, “bubble meshing” é adequado para métodos auto-adaptativos, embora isso não seja mencionado no artigo original [3].

A geração da malha após o equilíbrio é feita com uma triangulação de Delaunay com restrições. Embora existam algoritmos assintoticamente ótimos, que levam tempo $O(n \log n)$, eles não são simples de implementar e a constante escondida na notação O pode ser grande.

3 Acelerando “bubble meshing”

É possível acelerar “bubble meshing” e ainda obter os mesmos resultados. O passo mais importante é acelerar a interação das partículas. Como essa interação se dá somente entre partículas próximas, não é necessário considerar todos os pares de partículas, desde que saibamos identificar as partículas próximas a uma partícula. Para isso é necessário uma estrutura de dados mais sofisticada do que uma simples lista contendo todas as partículas.

Uma solução é manter todas as partículas ordenadas pela coordenada x do seu centro, e somente considerar a interação com partículas que estão próximas nesta lista. Esta solução foi implementada em 2D e mostrou-se razoavelmente eficiente. Uma outra solução, que ainda pretendemos avaliar, é decompor a região em células retangulares e somente considerar a interação entre partículas numa mesma célula ou nas células imediatamente vizinhas. Com ambas as soluções, reduz-se o número de interações de $\Theta(n^2)$ a $O(n)$, com algum *overhead* para manutenção das estruturas de dados.

O passo de geração da malha após o equilíbrio também pode ser acelerado. A principal observação aqui é que qualquer algoritmo genérico de triangulação tem que refazer uma ordenação espacial dos pontos. Esta é a razão da complexidade do problema de triangulação ser $\Omega(n \log n)$. Entretanto, se temos uma estrutura de dados que representa proximidade, como a necessária para fazer uma simulação eficiente, então essa ordenação espacial está praticamente completa, e uma triangulação pode ser extraída em tempo linear. Um algoritmo para fazer isso ainda está em desenvolvimento.

4 Conclusão

Embora existam métodos automáticos e auto-adaptativos para 2D, ainda não existem métodos completamente automáticos para 3D. Por exemplo, a geração de malhas por contração de fronteira é um método muito bem sucedido em 2D, mas que não funciona quando estendido para 3D. Os resultados preliminares com as acelerações descritas aqui para “bubble meshing” em 2D são animadores e sugerem sucesso também em 3D.

Agradecimentos. Roberto Ierusalimschy sugeriu usar uma lista ordenada para identificar partículas próximas. Os autores são parcialmente financiados por bolsas de pesquisa do CNPq e da FAPERJ.

Referências

[1] O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method. Volume 1: Basic Formulation and Linear Problems*. McGraw-Hill, fourth edition, 1989.

[2] I. Babuska and M. Suri. The P and H-P versions of the finite element method, basic principles and properties. *SIAM Review*, 36(4):578–632, 1994.

[3] K. Shimada and D. Gossard. Bubble mesh: Automated triangular meshing of non-manifold geometry by sphere packing. In C. Hoffman and J. Rossignac, editors, *Solid Modeling ’95*, pages 409–419, May 1995.

[4] F. Bossen. Anisotropic mesh generation with particles. Technical Report CMU-CS-96-134, Carnegie Mellon University, May 1996.