10ª Jornada Científica e Tecnológica
7º Simpósio da Pós-Graduação
INSTITUTO FEDERAL Sul de Minas Gerais
ISSN 2319-0124

ANÁLISE DE CUSTO COMPUTACIONAL DE ALGORITMOS INCREMENTAIS IMPLEMENTADOS NA CGAL PARA A TRIANGULAÇÃO DE DELAUNAY

Alan da Silva ROMUALDO [1]; Jéssica Renata NOGUEIRA [2]

RESUMO

A modelagem e a resolução de problemas físicos e da Engenharia utilizando procedimentos computacionais é usual. Uma das maneiras de se realizar essa modelagem é através da malha de Delaunay. Para um conjunto bidimensional contendo *n* pontos, essa é a melhor malha a ser gerada, por conter triângulos equiláteros ou o mais próximo a equiláteros. Para a construção dessa estrutura, a ordem em que os pontos são inseridos no domínio interfere no tempo dos algoritmos que a geram. Assim, neste trabalho, é avaliado o tempo que diferentes formas de inserção de pontos levam para gerar a malha de Delaunay. Os testes foram realizados na biblioteca CGAL, com até 10 milhões de pontos, e utilizaram-se as ordens de inserção de pontos aleatória, a ordem da curva de Hilbert e a ordem da curva de Hilbert combinada com *buckets* dessa biblioteca. Verificou-se que, dentre as ordens de inserção testadas, a curva de Hilbert com *buckets* apresentou os melhores custos computacionais, seguida da curva de Hilbert tradicional.

Palavras-chave:
Geração de malhas; Eficiência computacional; Análise de complexidade; Algoritmos Incrementais

1. INTRODUÇÃO

Neste trabalho, é realizada uma análise de custo computacional de algoritmos incrementais para a geração da malha de Delaunay bidimensional. Em linhas gerais, malhas são subdivisões de um domínio computacional em elementos geométricos. Nesta proposta, os elementos geométricos utilizados são triângulos. O uso de malhas surge, em principal, da necessidade de modelar domínios computacionais para a simulação de problemas físicos e de engenharia. Isso porque existem situações em que encontrar uma solução exata para um problema não é possível ou é inviável.

Para abordagens bidimensionais, a triangulação de Delaunay é a melhor malha possível de ser gerada. Isso porque os circuncírculos que passam por seus triângulos não são invadidos por pontos de outros triângulos. Assim, esses triângulos são equiláteros ou o mais próximo a equiláteros. Essa propriedade é especialmente útil em simulações de problemas que empregam o método dos elementos finitos e das diferenças finitas.

O estudo de algoritmos para a geração da triangulação de Delaunay teve início na década de 1970, sendo que diversos algoritmos foram propostos para a criação dessa malha (OLIVEIRA et al.,

1Bolsista de Inovação Tecnológica, IFSULDEMINAS – *Campus* Passos. E-mail: alan.romualdo@alunos.ifsuldeminas.edu.br.
2Orientadora, IFSULDEMINAS – *Campus* Passos. E-mail: jessica.nogueira@ifsuldeminas.edu.br.

2014). Embora o custo da construção dessa estrutura seja, no pior caso e para n pontos, conhecidamente $O(n \lg n)$, Bentley, Weide e Yao (1978) apresentaram o primeiro algoritmo com comportamento esperado linear para a geração do diagrama de Voronoi, que é o dual geométrico dessa malha (OLIVEIRA et al., 2014). Desde então, há um estudo sobre a possibilidade de aumentar a eficiência dos algoritmos para gerar a malha de Delaunay. Atualmente, para melhorar esse tempo, são utilizadas abordagens incrementais, em que é realizada uma ordenação dos pontos que serão inseridos no domínio. O estudo sobre a necessidade dessa ordenação ocorreu no trabalho de Amenta, Choi e Rote (2003), que propuseram que os pontos fossem inseridos de maneira tendenciosamente aleatória. Após a publicação de Amenta, Choi e Rote (2003), outras formas de ordenação dos pontos de entrada foram propostas, na tentativa de melhorar o custo computacional dos algoritmos para a construção da malha de Delaunay. Um exemplo de trabalho em que são avaliadas diferentes ordens de inserção de pontos pode ser encontrado em Oliveira e Nogueira (2016). Pode-se observar, na Figura 1, uma triangulação de Delaunay para 50 pontos.

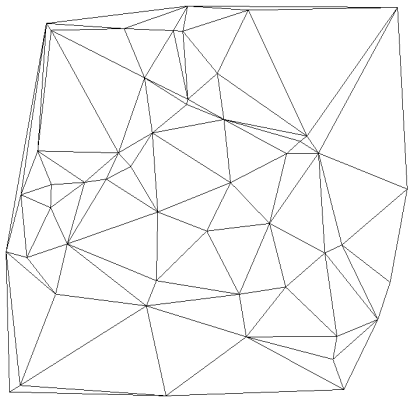

Figura 1: Triangulação de Delaunay para um conjunto de 50 pontos.

2. MATERIAL E MÉTODOS

Para realização dos testes, a seguinte máquina foi utilizada: Computador Acer Aspire E5-573G, Processador Intel Core i5-5200U, CPU 2,2Ghz x 4,8 GB de Memória RAM, Sistema Operacional Ubuntu, 32 *bits*, versão 16.04 LTS, DDR3, com velocidade de *clock* da RAM de 1600 MHz. Utilizou-se também o código disponibilizado na CGAL (2018), na versão 4.11, para a geração da triangulação de Delaunay. As ordens de inserção de pontos utilizadas da CGAL (2018) foram: aleatória, ordenação de pontos utilizando a ordem da curva de Hilbert e a ordenação de pontos realizando a divisão por *buckets* do domínio combinada com a posterior aplicação da curva de Hilbert nos *buckets*. Todas as entradas foram geradas de maneira aleatória no quadrado unitário.

3. RESULTADOS E DISCUSSÕES

Nesta seção, são apresentadas as médias e desvios padrões de tempo de execução dos

algoritmos incrementais. Para os testes, foram utilizados conjuntos contendo entre 500 mil e 10 milhões de pontos. Para cada conjunto, foram realizadas três execuções, foi calculada a média de tempo, o desvio padrão do tempo e os resultados são apresentados na Tabela 1. Na Figura 2, pode-se observar a média de tempo de execução para todas as entradas, utilizando a ordem da curva de Hilbert e a ordem da curva de Hilbert combinada ao uso de *buckets*. Testes cujo tempo de execução excedeu 10 minutos foram abortados.

n	Ordem de inserção dos pontos					
	Aleatória		Curva de Hilbert		Hilbert + *bucket*	
	Média	Desvio Padrão	Média	Desvio Padrão	Média	Desvio Padrão
500 mil	31,52	0,10	2,73	0,01	2,58	0,01
1 milhão	96,04	2,50	5,51	0,02	5,19	0,01
2.5 milhões	402,57	1,32	13,85	0,03	13,07	0,02
5 milhões	-	-	28,01	0,36	26,15	0,03
7.5 milhões	-	-	41,08	0,01	39,37	0,09
10 milhões	-	-	55,67	0,18	52,53	0,15

Tabela 1: Média e desvio padrão do custo computacional dos algoritmos incrementais

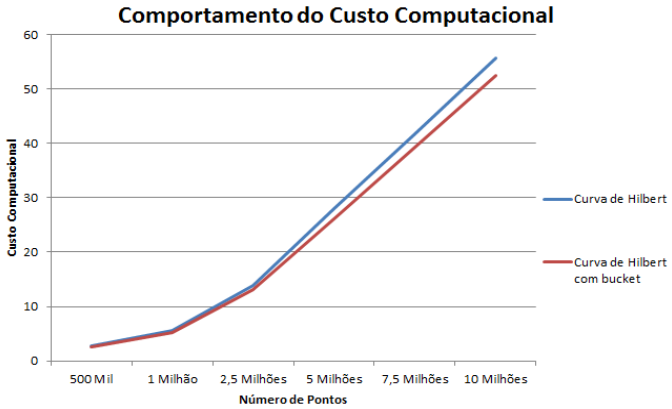

Figura 2: custo computacional dos algoritmos incrementais testados

Para as distribuições utilizando a ordem da curva de Hilbert e a ordem da curva de Hilbert com *bucket*, foi possível observar que, embora o tempo de execução esperado para a construção da malha de Delaunay bidimensional seja, no pior caso seja $O(n \lg n)$, um comportamento linear foi obtido em testes. Ainda, a subdivisão inicial do domínio por meio de *buckets* permitiu uma melhora no tempo de construção da triangulação de Delaunay, se comparado ao tempo de execução dos algoritmos que empregam apenas a ordem da curva de Hilbert. Conforme esperado, a não ordenação dos pontos faz com que o tempo de execução dos algoritmos para a construção da triangulação de Delaunay seja alto. Esse fato pode ser observado nos resultados obtidos na

distribuição aleatória, uma vez que foi observado um comportamento quadrático para essas execuções. Portanto, conforme esperado, o processo de ordenação dos pontos reduz o tempo de execução dos algoritmos para a geração da triangulação de Delaunay.

4. CONCLUSÕES

Este trabalho apresentou uma comparação do custo computacional de algoritmos para a criação da triangulação de Delaunay. Foi verificada a necessidade de ordenar os pontos de entrada, pois a inserção aleatória de pontos no domínio apresenta um custo computacional alto, se comparado ao custo de algoritmos cujos pontos são ordenados. Dentre as entradas testadas, a que mostrou melhor desempenho computacional foi a que utiliza a ordem da curva de Hilbert combinada ao uso de *buckets*, seguida da ordenação dos pontos através da ordem da curva de Hilbert pura. Próximos passos referem-se à implementação de outras formas de ordenação dos pontos de entrada, para comparação com outras ordenações de pontos propostas na literatura.

AGRADECIMENTOS

Este trabalho foi desenvolvido com o apoio financeiro do Programa Institucional de Incentivo ao Desenvolvimento de Inovação Tecnológica, no Edital 12/2016 do IFSULDEMINAS.

REFERÊNCIAS

AMENTA, N.; CHOI, S.; ROTE, G. (2003): Incremental constructions con BRIO. In Proc. 90th Annual Symposium on Computational Geometry, pp. 211–219, USA. ACM.

BENTLEY, J. L.; WEIDE, B. W.; YAO, A. C. (1978): Optimal expected-time algorithms forclosest point problems. In 16th Proceedings-Annual Allerton Conference on Communication, Control, and Computing, p. 843–851, Monticello, IL, USA.

CGAL – The Computational Geometry Algorithms Library (2018). Disponível em https://www.cgal.org/. Acesso em 26 de julho de 2018.

OLIVEIRA, S. L. G. de; NOGUEIRA, J. R.; TAVARES, J. M. R. S (2014): A systematic review of algorithms with linear-time behaviour to generate Delaunay and Voronoi tessellations. Computer Modeling in Engineering and Sciences, Duluth, v. 100, n. 1, p. 31-57.

OLIVEIRA, S. L. G. de; NOGUEIRA, J. R. (2016): An evaluation of point-insertion sequences for incremental. Computational & Applied Mathematics, v. 37, n. 1, p. 641- 674.