Geração e melhoramento de malhas adaptativas

Alexandre De Lacassa

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Geração e melhoramento de malhas adaptativas

Alexandre De Lacassa

***Orientador:* Prof. Dr. Antonio Castelo Filho**

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para obtenção do título de Doutor em Ciências - Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

USP – São Carlos
Agosto de 2012

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados fornecidos pelo(a) autor(a)

L128g   Lacassa, Alexandre De
              Geração e melhoramento de malhas adaptativas /
              Alexandre De Lacassa; orientador Antonio Castelo
              Filho. -- São Carlos, 2012.
              77 p.

              Tese (Doutorado - Programa de Pós-Graduação em
              Ciências de Computação e Matemática Computacional) --
              Instituto de Ciências Matemáticas e de Computação,
              Universidade de São Paulo, 2012.

              1. geração de malhas. 2. melhoramento de malhas.
              3. poligonalização. I. Castelo Filho, Antonio,
              orient. II. Título.

À Deus e aos meus pais

Agradecimentos

Agradeço à Deus, meu Senhor e Rei, à Jesus meu Salvador e amigo íntimo, razão do meu viver e de toda força que tenho.

Aos meus pais, Miguel e Nilma por simplesmente serem pais, vocês me fortalecem com seu amor.

Às minhas irmãs Juliana, Michelle, Daniele e Gisele, e aos meus sobrinhos, pelo imenso carinho com que sempre me presenteiam quando vou visitá-los.

Aos pastores Lael e Tânia pelas orações e amor que sempre demonstraram e por me acolherem em sua casa.

Ao meu amigo e discipulador Mateus e seu irmão Jônatas, com seus conselhos e apoio vocês me ajudaram a suportar os momentos mais difíceis que tive durante essa jornada.

Aos amigos Mario e sua esposa Juliana pelo carinho e bons conselhos nos agradáveis almoços de domingo.

A todos os irmãos da Comunidade Cristã de São Carlos, irmãos na fé.

Aos amigos do LCAD, principalmente ao amigo Miguel Caro Candezano, pelas palavras de ânimo e pela companhia na torcida pelo timão.

Aos amigos Mario Liziér e João Paulo Gois que me ajudaram com minhas dúvidas e dificuldades de programação.

Ao meu orientador Prof. Castelo, pela compreensão e ajuda no desenvolvimento deste trabalho.

À CAPES pelo suporte financeiro.

Resumo

Técnicas para geração de malhas volumétricas não estruturadas podem ser divididas em três categorias principais: técnicas baseadas em Octree, técnica de avanço de fronteira e técnicas Delaunay. Este trabalho faz uso de uma proposta para geração de malhas volumétricas não estruturadas baseada em uma técnica que utiliza uma triangulação adaptativa que contém similaridades com os métodos de geração de malhas baseados em Octree. Esta triangulação, além de decompor o domínio, oferece suporte para a construção de funções implícitas que aproximam a superfície do objeto de interesse. As funções implícitas são obtidas a partir de nuvens de pontos pelo método de Partição da Unidade Implícita - PUI. Um processo de poligonalização discretiza a isosuperfície fornecida pela função implícita gerando uma malha de triângulos superficial. Uma malha volumétrica é gerada a partir da malha superficial utilizando os tetraedros que decompõem o domínio criados pela triangulação.

Apresentamos nesta tese um método original para a poligonalização de isosuperfícies obtidas através de nuvens de pontos. Apresentamos também uma abordagem para o melhoramento de malhas volumétricas baseadas na aplicação do conceito de molas virtuais. O foco principal é gerar malhas de qualidade para serem usadas em simulações de fluidos em artérias do corpo humano.

Abstract

Techniques for generation of three-dimensional unstructured meshes can be divided into three main categories: Octree-based techniques, advancing front techniques and Delaunay techniques. In the present study it is used a three-dimensional unstructured mesh generation approach based on a technique which uses an adaptive triangulation which contains similarities with Octree-based mesh generation techniques. This triangulation decomposes the domain and supports the construction of implicit functions that approximates the surface of an object of interest. The implicit functions are derived from point clouds by the Partition of Unity Implicits - PUI method. A polygonization process discretizes the isosurface provided by the implicit function generating a triangular mesh surface. A three-dimensional mesh is generated from the surface mesh using the tetrahedra that decomposes the domain created by the triangulation.

In this thesis, we present an original polygonization method of isosurfaces obtained from point clouds. We also present an approach to improve tetrahedral meshes based on the application of the concept of virtual springs. The main focus is to generate quality meshes for fluid flow simulations in human arteries.

Sumário

1	Introdução	1
1.1	Qualidade de malhas	2
1.2	Geradores de malhas na literatura	3
1.3	Principais contribuições	4
1.4	Organização da tese	6
2	Geração de malhas volumétricas	7
2.1	Técnicas de geração de malhas	7
2.1.1	Octree	8
2.1.2	Avanço de fronteira	8
2.1.3	Delaunay	9
2.2	A triangulação J_1^a	12
3	Partição da unidade implícita - PUI	17
3.1	Funções implícitas	17
3.2	Partição da unidade implícita em multinível	18
4	Poligonalização da superfície implícita	23
4.1	Melhoramento de malhas superficiais	24
4.1.1	Melhoramento por heurística	26
4.1.2	Melhoramento por otimização	26
4.2	Algumas considerações	33
5	Melhoramento de malhas tetraedrais	35
5.1	Melhoramento de malhas por suavização	36
5.2	Melhoramento de malhas por operações topológicas	37
5.3	Melhoramento de malhas por inserção de vértices	38
5.4	Suavização de malhas através de molas virtuais	38

5.4.1	Molas longitudinais	40
5.4.2	Molas torcionais	41
5.4.3	Molas semi-torcionais	42
5.4.4	A estratégia de suavização de malhas com molas virtuais	43
5.4.5	Suavização por molas em malhas superficiais	46
5.4.6	Suavização por molas em malhas volumétricas	47
6	Resultados	49
6.1	Resultados do melhoramento de malhas superficiais	49
6.2	Resultados do melhoramento de malhas com molas virtuais	54
7	Considerações Finais	69
	Referências Bibliográficas	72

Lista de Figuras

1.1	Sequência de execução do processo de reconstrução de isosuperfícies por nuvens de pontos desenvolvido por Gois e Polizelli-Junior em [18, 19]	5
2.1	Decomposição de um objeto 2D por uma Quadtree [35].	8
2.2	Exemplo 2D do método de avanço de fronteira onde uma faixa de triângulos foi criada [35].	9
2.3	Exemplo do critério de Delaunay. Em (a) o critério de Delaunay é mantido enquanto em (b), não.	10
2.4	Exemplo bidimensional da triangulação J_1^a (esquerda) e os detalhes do bloco $g = (3, 2)$, $r = 0$ e dois caminhos para representar simplexes (direita). Figura extraída de [10]	13
3.1	Reconstrução da estátua Lucy, de Stanford (16 milhões de pontos). (Figura extraída do trabalho Gois et al. [17, 18])	19
4.1	Os tipos de intersecção de uma isosuperfície com um tetraedro. As arestas vermelhas formam os triângulos gerados pela intersecção. A área pintada em azul indica a representação poligonal da isosuperfície. Os vértices sinalizados com “+” estão do lado de fora, os vértices sinalizados com “-” estão do lado de dentro do modelo, e os vértices sinalizados com “o” indicam que a isosuperfície passa por aquele vértice. . .	25
4.2	As figuras (a) e (b) apresentam o modelo Bunny de Stanford gerado pelo poligonalizador do trabalho de Gois et al. [19] sem o melhoramento proposto originalmente. As figuras (c) e (d) apresentam o mesmo modelo gerado pelo poligonalizador com o melhoramento proposto também em [19]. As figuras (e) e (f) apresentam o mesmo modelo gerado com o melhoramento de malhas superficiais proposto neste capítulo. A escala de cores indica o vermelho como melhor qualidade ($R/2r \approx 1$) e o azul como pior qualidade ($R/2r > 5$).	32

5.1	Exemplo 2D de melhoramento de malha por Suavização: o vértice é deslocado para uma região do polígono determinado pelos triângulos incidentes ao vértice para melhorar a qualidade destes triângulos.	37
5.2	Operações topológicas: flip 2-3 e a operação inversa o flip 3-2, o flip 4-4 e o flip 2-2.	39
5.3	Remoção de arestas e Remoção multi-faces: nesta figura 2 operações topológicas são ilustradas, sendo uma o inverso da outra, a remoção de aresta elimina uma aresta do conjunto de tetraedros e acrescenta novas faces para gerar um novo conjunto de elementos, enquanto que a remoção multifaces, de forma contrária, elimina as faces e acrescenta uma aresta.	40
5.4	Ilustração de uma mola longitudinal.	41
5.5	Ilustração de uma mola torcional [47].	41
5.6	Mola semi-torcional [47].	42
5.7	Mola semi-torcional em tetraedros.	43
5.8	Interpretação física do processo de iteração em um sistema de molas.	45
5.9	(a) e (b) malha superficial de uma elipse e histograma da medida de qualidade; (c) e (d) malha superficial suavizada e seu histograma de qualidade.	47
5.10	Modelo de uma elipse; (a) malha volumétrica inicial; (c) a malha suavizada pelo algoritmo 1.	48
6.1	Malhas superficiais do Modelo <i>abdominal aortic aneurysm</i>	51
6.2	Malhas superficiais do Modelo <i>iliac aneurysm</i>	52
6.3	Malhas superficiais do Modelo <i>basilar artery</i>	53
6.4	Diagrama de execução dos testes feitos para obtenção das malhas volumétricas apresentadas nesta seção.	55
6.5	Malha volumétrica do modelo <i>renal artery</i>	58
6.6	Histogramas de qualidade das malhas do modelo <i>renal artery</i>	59
6.7	Malha volumétrica do modelo <i>abdominal aortic</i>	60
6.8	Histogramas de qualidade das malhas do modelo <i>abdominal aortic</i>	61
6.9	Malha volumétrica do modelo <i>basilar artery</i>	62
6.10	Histogramas de qualidade das malhas do modelo <i>basilar artery</i>	63
6.11	Malha volumétrica do modelo <i>iliac aneurysm</i>	64
6.12	Histogramas de qualidade das malhas do modelo <i>iliac aneurysm</i>	65
6.13	Malhas volumétricas do modelo <i>abdominal aortic aneurysm</i>	66
6.14	Histogramas de qualidade das malhas do modelo <i>abdominal aortic aneurysm</i>	67
6.15	O modelo <i>abdominal aortic aneurysm</i> seccionado para visualização dos elementos internos da malha. A figura (a) é a malha inicial; a figura (b) é a malha inicial melhorada com a ferramenta <i>stellar</i> ; a figura (c) é a malha inicial melhorada pela suavização por molas e a figura (d) é a malha da figura (c) depois do melhoramento feito pela ferramenta <i>stellar</i>	68

Introdução

O uso de métodos de elementos finitos e volumes finitos é comum na solução de problemas aplicados a ciência e engenharia. Tais problemas frequentemente envolvem a simulação numérica de equações diferenciais parciais que governam fenômenos tais como: deformação mecânica, transferência de calor, dinâmica dos fluidos, propagação de ondas eletromagnéticas, mecânica quântica, entre outros. Uma característica comum em alguns destes problemas é que a maioria dos objetos de interesse, que caracterizam o domínio do problema, contém formas complexas. Para que seja possível a simulação sobre estes objetos é necessária a sua decomposição em formas geométricas simples chamadas *elementos*, que são comumente tetraedros ou triângulos, mas podem ser também quadriláteros, prismas, pirâmides ou hexaedros. Os elementos e seus vértices discretizam o objeto permitindo a realização dos cálculos do método de elementos/volumes finitos. Ao conjunto de todos os vértices e elementos que representam o objeto denomina-se malha.

As malhas são geralmente categorizadas em estruturadas e não-estruturadas. As malhas estruturadas apresentam uma estrutura topológica uniforme, o que não acontece com malhas não-estruturadas. Nas malhas estruturadas, os índices dos vizinhos de um vértice podem ser obtidos por uma simples soma. Nas malhas não-estruturadas, os vizinhos são obtidos por meio de consultas em estruturas de

dados mais elaboradas. Um tipo particularmente simples de malha estruturada é a malha cartesiana regular, onde os elementos são quadriláteros ou hexaedros idênticos.

Este trabalho aborda a geração de malhas não-estruturadas volumétricas e também a geração de malhas de superfície. As malhas volumétricas em questão são compostas por tetraedros enquanto que as malhas de superfície são compostas por triângulos.

A qualidade das soluções para os problemas envolvendo métodos de elementos finitos está intimamente relacionada com o formato dos elementos da malha. Para malhas de tetraedros, o ângulo diedral caracteriza o efeito que o elemento exerce sobre a solução do problema. Tetraedros com ângulos diedrais grandes (próximos de 180°) [45] causam erros de interpolação e discretização, o que compromete a precisão da simulação numérica. Já ângulos diedrais pequenos (próximos de 0°) causam mal-condicionamento nas matrizes associadas ao método de elementos finitos [45]. Embora malhas anisotrópicas sejam interessantes e necessárias em alguns contextos, como por exemplo em aerodinâmica, neste trabalho abordaremos apenas as malhas isotrópicas, podendo ser usadas preferencialmente em simulações mecânicas, de transferência de calor, e também eletromagnetismo.

1.1 Qualidade de malhas

Podemos então expressar a qualidade de um tetraedro, ou de um triângulo, como sendo um número que represente seu efeito sobre erros de interpolação, erros de discretização, e sobre o condicionamento das matrizes do método de elementos finitos. Por isso, algumas medidas de qualidade de tetraedros podem ser mais eficazes do que outras, dependendo do tipo de aplicação na qual são usadas. Devido as restrições que elementos com ângulos muito grandes ou com ângulos muito pequenos causam em uma simulação, podemos dizer que a qualidade de uma malha está intimamente relacionada com o menor e o maior ângulo entre todos os seus elementos [45]. Portanto, neste trabalho utilizaremos a medida do menor ângulo entre as faces que compõem o tetraedro (ângulo diedral) como uma medida que expresse sua qualidade, sendo que o valor máximo para o menor ângulo encontrado em um tetraedro regular é de $\arccos(1/3) = 70,5^\circ$. Em alguns momentos também citaremos a medida do maior ângulo diedral do tetraedro como uma medida de qualidade. Para malhas superficiais compostas por triângulos, utilizaremos a medida de qualidade chamada “razão dos raios” determinada

pela razão entre os raios dos círculos circunscrito e inscrito de um triângulo. A escolha desta medida de qualidade para malhas superficiais é suficiente para identificar triângulos ruins, com ângulos pequenos (próximos de 0°) ou grandes (próximos de 180°). O valor mínimo para a medida “razão dos raios” é 1, que é a medida aferida do triângulo equilátero.

Seja $q(t)$ uma função que represente a qualidade de um triângulo t . A métrica “razão dos raios” é definida pela seguinte expressão:

$$q(t) = \frac{R}{2r} \quad (1.1)$$

onde R é o raio do círculo circunscrito do triângulo t e r é o raio do círculo inscrito de t .

Da mesma forma para tetraedros a qualidade $q(t)$ aferida pela métrica “razão dos raios” é definida pela expressão:

$$q(t) = \frac{R}{3r} \quad (1.2)$$

onde R é o raio da esfera circunscrita do tetraedro t e r é o raio da esfera inscrita de t .

1.2 Geradores de malhas na literatura

Métodos para geração de malhas procuram sempre gerar elementos com boa qualidade. Uma das técnicas de geração que oferecem garantia de qualidade sobre os elementos da malha é a triangulação de Delaunay [25, 48]. Esta técnica gera triângulos (2D) ou tetraedros (3D) e procura manter uma propriedade sobre cada elemento da triangulação, que é a propriedade do círculo vazio, onde o círculo (esfera, no caso 3D) que passa pelos vértices do triângulo (tetraedro, no caso 3D) não contém nenhum ponto da malha no seu interior.

Outra técnica muito conhecida de geração de malhas é a técnica de avanço de fronteira [31, 26]. Nela, uma fronteira de avanço composta por triângulos é construída inicialmente na borda do domínio, e em cada passo esta fronteira avança com a criação de novos elementos sobre a fronteira do passo anterior. Os elementos criados em cada passo obedecem a um critério de qualidade mínimo. A fronteira deixa de avançar quando o domínio está completamente preenchido. Este tipo de triangulação pode gerar elementos ruins se não houver um cuidado especial quando as fronteiras de avanço se

encontram.

As técnicas de geração de malhas baseadas em octree [51] particionam o domínio com uma estrutura octree e então os elementos (triangulares) da malha são construídos utilizando os vértices da octree e os pontos de interseção entre o domínio e a estrutura octree.

Como estes métodos nem sempre conseguem gerar malhas com bons elementos, estratégias para melhorar a qualidade destes elementos são necessárias. O algoritmo de Refinamento Delaunay busca melhorar a qualidade de uma malha Delaunay através da inserção de pontos em locais estratégicos para eliminar da malha os elementos com qualidade inferior a um certo valor pré-estabelecido.

Também existem técnicas para melhoramento de uma malha, seja ela Delaunay ou não, que não inserem novos pontos na malha, mas movimentam os pontos da malha para novos locais que melhoram os seus elementos. Métodos baseados em malhas dinâmicas buscam inserir uma característica elástica nas arestas da malha utilizando o conceito de molas. Esta técnica desloca os vértices da malha ao buscar o equilíbrio entre as molas inseridas nas arestas dos elementos.

Métodos de melhoramento de malhas baseados em otimização também deslocam os vértices para novas posições. Porém, os deslocamentos são feitos através de processos de maximização ou minimização de funções que fornecem informações sobre a qualidade dos elementos afetados pelos deslocamentos dos vértices.

1.3 Principais contribuições

Neste trabalho de doutorado é apresentado um método novo para a poligonalização de isosuperfícies obtidas através de nuvens de pontos. Este método é baseado em uma estratégia de otimização. Apresentamos também uma abordagem de melhoramento de malhas volumétricas. O foco principal é gerar malhas de qualidade para serem usadas em simulações de artérias do corpo humano. Estas técnicas devem ser integradas a uma ferramenta para simulação computacional do Sistema Cardiovascular Humano [3] desenvolvido no projeto HemoLab do LNCC - Laboratório Nacional de Computação Científica.

O desenvolvimento do gerador de malhas apresentado em [10] é uma extensão do trabalho desenvolvido por Gois e Polizelli-Junior em [18, 19]. O trabalho de Gois e Polizelli aborda apenas a

reconstrução de isosuperfícies por nuvens de pontos. Em [19] a reconstrução de isosuperfícies por nuvem de pontos utiliza um método de decomposição espacial por tetraedros baseado na triangulação J_1^a [7] juntamente com um método de partição da unidade implícita - PUI - originalmente proposto em [18]. Este método fornece uma função implícita para aproximar uma superfície amostrada por uma nuvem de pontos. A reconstrução é completada pelo processo de poligonalização que extrai uma malha superficial através da interseção da triangulação J_1^a com a função implícita de nível 0. A figura 1.1 ilustra o processo de reconstrução de isosuperfícies desenvolvido em [18, 19].

Figura 1.1: Sequência de execução do processo de reconstrução de isosuperfícies por nuvens de pontos desenvolvido por Gois e Polizelli-Junior em [18, 19]

A principal contribuição desta tese foi a criação de uma nova forma de extrair a malha superficial deslocando os vértices dos elementos da decomposição espacial J_1^a . Em [19] é apresentado também um método heurístico para o melhoramento da malha extraída pelo poligonalizador. Entretanto, neste trabalho apresentamos uma estratégia baseada em otimização para o melhoramento da malha superficial gerada pelo poligonalizador.

O gerador apresentado em [10] tem como objetivo estender o trabalho de Gois et al. [19] inserindo elementos internos na malha superficial, obtendo assim, uma malha volumétrica. Para cumprir este objetivo foram utilizadas as ideias apresentadas por Labelle e Shewchuk em [24]. Em [24] os elementos da decomposição espacial são aproveitados como elementos internos a malha da superfície. Em [10] os elementos da malha de decomposição do espaço (neste caso, os tetraedros da triangulação J_1^a) são aproveitados como elementos internos do modelo representado pela malha superficial extraída pelo poligonalizador. Dessa forma é obtida uma malha de tetraedros do modelo representado pela malha superficial. Essas malhas tem como principal característica a adaptatividade, ou seja, um número maior de elementos próximos a superfície, e menos elementos no interior do modelo. No entanto, estas malhas ainda contém um número grande de elementos com uma qualidade razoável que pode ser melhorada.

Neste sentido, desenvolvemos um processo de pós-processamento que busca suavizar a malha

melhorando a qualidade dos seus elementos.

Este pós-processamento consiste em uma estratégia de melhoramento de malhas por suavização através de molas virtuais [47, 2, 14, 52]. Neste processo a suavização é aplicada primeiro nos elementos de superfície e depois nos elementos internos de uma malha volumétrica. As arestas da malha funcionam como molas virtuais entre os vértices, gerando assim um sistema de molas. Neste sistema de molas forças atuam sobre os vértices que precisam se deslocar para manter o equilíbrio estático do sistema. Dessa forma, tanto a malha interna quanto a malha de superfície são suavizadas. A ferramenta *Stellar* criada por Klingner e Shewchuk em [23] é utilizada para melhorar ainda mais a qualidade da malha final.

1.4 Organização da tese

Os capítulos deste trabalho estão estruturados da forma que segue. O capítulo 2 apresenta alguns métodos para geração de malhas não-estruturadas, bem como a triangulação J_1^a , que é a base do gerador utilizado neste trabalho. O capítulo 3 apresenta o Método de Partição da Unidade Implícita que fornecerá uma função implícita que aproxima a superfície amostrada por uma nuvem de pontos. O capítulo 4 apresenta o método baseado em otimização para poligonalização de uma superfície implícita. O capítulo 5 apresenta as principais técnicas encontradas na literatura para melhoramento de malhas. O capítulo 5.4 apresenta o processo de melhoramento de malha por suavização através de molas virtuais. Os capítulos 6 e 7 apresentam os resultados e conclusões.

Geração de malhas volumétricas

Neste capítulo são apresentadas algumas técnicas para geração de malhas volumétricas que são encontradas na literatura segundo uma classificação proposta no trabalho de Owen [35]. É apresentada também a triangulação J_1^a proposta nos trabalhos de Castelo et al. [7, 6], que é utilizada neste trabalho.

2.1 Técnicas de geração de malhas

Malhas de triângulos ou de tetraedros são os tipos de malhas mais comuns encontradas nos geradores de malhas não-estruturadas. Em [35] é apresentada uma diferenciação das técnicas mais usadas atualmente, que podem ser agrupadas em três categorias principais:

- Octree
- Avanço de fronteira
- Delaunay

Embora exista uma considerável diferença em complexidade quando se passa do caso 2D para o caso 3D, os algoritmos apresentados abaixo são, em sua maior parte, aplicáveis tanto em geração de malhas de triângulos quanto em geração de malhas tetraedrais.

2.1.1 Octree

A técnica Octree para geração de malhas foi inicialmente desenvolvida por um grupo liderado por Mark Shephard [50, 51, 40, 41]. Neste método cubos contendo o modelo geométrico são subdivididos recursivamente até que se atinja uma resolução desejada. A Figura 2.1 mostra a decomposição 2D de um modelo por Quadtree, o equivalente bi-dimensional da Octree. Células irregulares são criadas onde os cubos intersectam a superfície, o que exige uma quantidade considerável de cálculos de intersecção da superfície. Os tetraedros são gerados a partir das células irregulares que intersectam a superfície e pelas células regulares internas do modelo. A técnica Octree, diferentemente das técnicas Delaunay e avanço de fronteira, não necessita de uma malha da superfície do modelo para ser preenchida. As faces de superfície são obtidas conforme a estrutura interna da Octree intersecta a superfície. Para garantir que não aconteça uma mudança brusca no tamanho dos elementos, a diferença de nível entre cubos adjacentes pode ser limitada a um nível de subdivisão da Octree. Operações de suavização e simplificação também podem ser aplicadas em um pós-processamento para melhorar o aspecto dos elementos, segundo algum critério de qualidade.

Figura 2.1: Decomposição de um objeto 2D por uma Quadtree [35].

2.1.2 Avanço de fronteira

A técnica de Avanço de Fronteira (ou método de fronteira móvel) é outra técnica muito popular entre os geradores de malhas. Dois dos principais contribuidores deste método são Rainald Löhner [31, 29, 30, 28, 46] e S. H. Lo [27, 26]. Neste método, os tetraedros são construídos de forma progressiva, da superfície para a parte interna do objeto. Uma fronteira “ativa” é mantida, formada inicialmente pelos triângulos que compõem a superfície. A partir da fronteira ativa novos tetraedros são formados. A Figura 2.2 mostra um exemplo bidimensional do método Avanço de Fronteira, onde os

triângulos estão sendo formados na fronteira. Assim que o algoritmo progride, a fronteira irá avançar para preencher a área restante com triângulos. No caso tridimensional, para cada face triangular na fronteira ativa, é calculada a localização ideal para um novo vértice, que formará um tetraedro da malha. Também é analisado qualquer outro vértice existente na fronteira ativa que pode formar, com esta face, um tetraedro com boa qualidade. O algoritmo então faz uma escolha entre o tetraedro resultante da inserção do novo vértice e o tetraedro resultante da escolha do vértice existente na fronteira ativa. A opção que gerar o tetraedro com melhor qualidade é escolhida. O algoritmo também requer uma checagem de intersecção para garantir que não aconteça sobreposição de elementos conforme as fronteiras ativas opostas avançam uma sobre a outra. Pode ser utilizada também uma função para controlar o tamanho dos elementos da malha.

Figura 2.2: Exemplo 2D do método de avanço de fronteira onde uma faixa de triângulos foi criada [35].

2.1.3 Delaunay

As técnicas mais populares de geração de malhas de triângulos e de tetraedros são aquelas que utilizam o critério de Delaunay [12]. O critério de Delaunay, chamado às vezes de propriedade da “esfera vazia” (ou “círculo vazio”, no caso bidimensional), determina que nenhum vértice deve estar contido dentro da circunsfera de qualquer tetraedro da malha. A circunsfera é definida pela esfera que passa pelos quatro pontos que definem um tetraedro. A Figura 2.3 ilustra o critério de Delaunay para o caso bidimensional. Na figura, a propriedade do círculo vazio é mantida desde que os circuncírculos dos triângulos em 2.3(a) não contenham nenhum outro ponto da triangulação. Embora o critério de Delaunay seja conhecido há muito tempo, somente nos trabalhos de Charles Lawson [25] e Dave

Figura 2.3: Exemplo do critério de Delaunay. Em (a) o critério de Delaunay é mantido enquanto em (b), não.

Watson [48] foi utilizado no desenvolvimento de algoritmos para triangular um conjunto de vértices. Os algoritmos baseados no critério de Delaunay geram uma triangulação que é única para o conjunto de vértices da malha, desde que os vértices estejam em posição geral, ou seja, não mais que quatro vértices pertencentes a uma mesma esfera.

O critério de Delaunay não é, por si só, um algoritmo para gerar malhas. Ele simplesmente provê um critério para conectar um conjunto de pontos existentes no espaço. Portanto, é necessário fornecer um método para gerar pontos que pertençam a uma dada geometria. Uma abordagem comum é primeiro gerar uma malha a partir dos pontos pertencentes ao bordo da geometria, fornecendo assim uma malha inicial. Os pontos de bordo são então triangulados de acordo com o critério de Delaunay. Novos pontos são então adicionados na malha existente de forma incremental, redefinindo os triângulos ou tetraedros localmente para satisfazer o critério de Delaunay.

Depois da inserção de um novo ponto, os dois algoritmos mais comuns para manter o critério de Delaunay em uma malha são: o algoritmo de “flipping incremental” [21, 25] e o algoritmo de Bowyer-Watson [5, 48].

O método escolhido para posicionar o novo ponto a ser inserido é o que diferencia um algoritmo de Delaunay de outro. Um dos algoritmos mais simples baseado na abordagem de inserção de pontos define a localização dos novos pontos a partir de uma grade regular de pontos que cobre o domínio com uma densidade específica. Para que os elementos variem em tamanho, uma função de tamanho especificada pelo usuário pode ser definida e então os novos pontos serão adicionados até que seja

atingido um valor mínimo para a função de tamanho escolhida. Outra abordagem é inserir recursivamente os pontos nos centróides dos triângulos ou tetraedros. Weatherill e Hassan [49] propõem um gerador de malhas de tetraedros onde os novos pontos são inseridos nos centróides dos tetraedros até atingir um valor mínimo para uma função de tamanho dos tetraedros.

Uma abordagem diferente é proposta por Chew [9] e Ruppert [39], que propõem a inserção dos novos pontos nos centros das circunsferas/circuncírculos dos tetraedros/triângulos. Nesta técnica, quando se adota uma sequência específica na inserção dos novos pontos, pode-se dizer que há uma garantia de qualidade pois os tetraedros/triângulos podem ser criados respeitando um limite mínimo sobre os ângulos presentes nos elementos da malha. Estas técnicas de inserção de pontos sobre uma malha Delaunay são conhecidas também como técnicas de refinamento Delaunay.

Baseado na técnica de refinamento Delaunay proposto por Ruppert, o trabalho desenvolvido por Jonatan R. Shewchuk [43] é um dos mais conhecidos atualmente. Shewchuk desenvolveu um arcabouço numérico de aritmética exata para calcular com exatidão os testes que verificam e mantêm o critério de Delaunay na malha [42], o que tornou seu algoritmo um dos mais robustos para geração de malhas Delaunay no espaço tridimensional. Sem este arcabouço numérico, a triangulação de Delaunay pode apresentar mais que uma única triangulação. No caso tridimensional, esta necessidade é mais evidente, devido ao número de operações aritméticas ser maior do que no caso bidimensional, o que aumenta a probabilidade de erro no cálculo do critério de Delaunay.

Embora esse critério seja uma boa alternativa para uma algoritmo de geração de malhas volumétricas, um tratamento especial é necessário para eliminar elementos ruins como os tetraedros “slivers”. Estes tipos de tetraedros são ruins para fins de simulação numérica pois contém ângulos diedrais próximos de 0° e de 180° . A medida de qualidade conhecida como razão “circunraio-menor aresta” não é suficiente para detectar um tetraedro “sliver”. Isso foi observado nos trabalhos de Shewchuk, que utiliza a medida do menor ângulo diedral do tetraedro para identificar tais tetraedros e eliminá-los. O trabalho apresentado em [8] fornece uma técnica para identificar e eliminar estes tipos de tetraedros em uma malha Delaunay.

2.2 A triangulação J_1^a

Proposta por Castelo et al. [7, 6], a triangulação J_1^a é uma estrutura definida algebricamente que pode ser construída em qualquer dimensão de forma eficiente. Para tratar detalhes locais esta estrutura é naturalmente capaz de realizar refinamentos, o que a classifica como um método de triangulação adaptativo, sendo uma das suas principais características. Outras características importantes são a existência de um mecanismo para representação de simplexes e a existência de regras algébricas para percorrer os elementos da triangulação. Estas regras proporcionam um armazenamento eficiente e econômico da estrutura, pois não é necessário armazenar informações sobre a conectividade dos seus elementos.

A triangulação J_1^a em 3D usada para gerar malhas volumétricas pode ser classificada como um método baseado em Octree, uma vez que a construção do particionamento do domínio através da triangulação J_1^a é semelhante a construção feita pelos métodos baseados em Octree. No entanto, a característica algébrica da triangulação J_1^a evita o uso de estruturas de dados geralmente empregadas nos métodos baseados em Octree. Outra diferença com relação a tais métodos, é que os elementos tetraedrais que farão parte da malha final já estão presentes na descrição da triangulação J_1^a .

A triangulação J_1^a em $\mathbb{R}^n$ consiste de uma grade formada por hipercubos n -dimensionais. Estes hipercubos são chamados de blocos. Cada bloco é dividido em $2^n n!$ n -simplexos (8 triângulos em 2D, ou 48 tetraedros em 3D) que podem ser descritos algebricamente usando apenas seis valores:

$$S = (g, r, \pi, s, t, h) \quad (2.1)$$

Os dois primeiros elementos de S definem em qual bloco o simplexo está contido, onde g é um vetor de dimensão n que indica as coordenadas de um bloco em um determinado nível de refinamento r da grade. A Figura 2.4 ilustra, à esquerda, uma triangulação J_1^a em $\mathbb{R}^2$, e à direita, um bloco destacado com nível de refinamento $r = 0$ (0-bloco) e $g = (3, 2)$. Nesta mesma figura pode-se notar que os blocos azul-escuros apresentam um nível de refinamento $r = 1$ – chamados de 1-blocos – e, por isso, fazem parte de uma grade de resolução maior.

Figura 2.4: Exemplo bidimensional da triangulação J_1^a (esquerda) e os detalhes do bloco $g = (3, 2)$, $r = 0$ e dois caminhos para representar simplexos (direita). Figura extraída de [10]

Para tratar diferentes níveis de refinamento, a triangulação J_1^a divide um bloco (que é escolhido para refinamento) em 2^n blocos, aplicando mudanças nos seus blocos vizinhos para acomodar os blocos menores que foram criados nesta divisão. Assim, um tipo diferente de bloco é criado, o qual é chamado de bloco de transição. A fim de manter um padrão nos blocos de transição, a triangulação J_1^a proíbe que haja uma diferença maior que um entre os níveis de refinamento de blocos vizinhos. O nível de refinamento de um bloco de transição é definido como o menor nível dos seus blocos vizinhos.

O bloco de transição tem algumas das suas faces k -dimensionais (onde $0 < k < n$) refinadas para poder acomodar seus blocos vizinhos refinados. Tal fato está ilustrado na Figura 2.4, onde o bloco de transição destacado tem apenas sua aresta esquerda refinada. Os blocos que não são de transição (blocos em azul claro e azul escuro na Figura 2.4), são chamados de blocos básicos.

Para representar cada n -simplexo de um bloco, nota-se primeiro que todos os simplexos compartilham pelo menos um vértice: o vértice central, marcado como v_0 . A partir do vértice central do hipercubo n -dimensional, o próximo passo é dado na direção positiva ou negativa de um eixo coordenado escolhido. Este passo encontrará o vértice v_1 , que se encontra no centro de uma face $n - 1$ -dimensional. O processo continua para os vértices $v_2 \dots, v_n$, sempre partindo do vértice anterior em uma direção positiva ou negativa em um dos eixos, até o centro de uma face $n - 2, \dots, 0$ -dimensional. Dessa forma pode-se representar cada simplexo pelo caminho que se faz de v_0 até v_n . Os vetores π

e s são responsáveis por codificar este caminho, sendo que π armazena a permutação de n inteiros de 1 até n que representam os eixos coordenados, e s representa a direção (por valores negativos ou positivos) a ser tomada em cada eixo. Na Figura 2.4, o simplexo 1 no bloco destacado é representado por $\pi = (2, 1)$ e $s = (1, -1)$, o que significa que o primeiro caminho a ser tomado será pelo eixo y ($\pi_1 = 2$) na direção negativa ($s_{\pi_1} = -1$), e depois pelo eixo x ($\pi_2 = 1$) na direção positiva ($s_{\pi_2} = 1$).

Os vetores π e s contém informações suficientes para representar os simplexos de um bloco básico e os simplexos de um bloco de transição, mas não contém informações suficientes para indicar uma face refinada. Para os casos restantes, nota-se que quando uma face k -dimensional é atingida, existem mais que um centro. Como esta informação deve ser obrigatoriamente armazenada, pois é necessária para representar estes simplexos, o escalar h é usado para informar quantos passos são dados antes de encontrar uma face refinada, e o vetor t define os sinais extras para os eixos $\pi_{h+1} \dots \pi_n$, que são usados para selecionar um centro de face dentre todas as possibilidades em uma face refinada. Observando novamente a Figura 2.4, o simplexo 2 é representado por $\pi = (1, 2)$, $s = (-1, 1)$, $h = 1$ e $t = (0, 1)$; h é igual a 1 pois apenas um único passo é dado antes de atingir uma aresta refinada e a escolha do centro para posicionar o vértice v_1 está na direção positiva de π_{h+1} .

Uma maneira de descrever formalmente a representação de um simplexo dentro de um hipercubo é dada pela seguinte expressão:

$$\begin{cases} v_0 &= (0, \dots, 0); \\ v_i &= v_{i-1} + e_{\pi_i} s_{\pi_i}, \text{ para } 1 \leq i < h; \\ v_h &= v_{h-1} + e_{\pi_h} s_{\pi_h} + \frac{1}{2} \sum_{k=h+1}^n e_{\pi_k} t_{\pi_k}; \\ v_i &= v_{i-1} + \frac{1}{2} e_{\pi_i} s_{\pi_i}, \text{ para } h < i \leq n, \end{cases}$$

onde e_i é um vetor com valor 1 na posição i e 0 no restante das posições.

Outro importante mecanismo da triangulação J_1^a são as regras de pivotamento para percorrer a triangulação sem usar uma estrutura topológica auxiliar. Na Figura 2.4 duas operações de pivotamento são ilustradas, onde o simplexo 1 é pivotado em relação ao vértice v_1 resultando no simplexo 3, e o simplexo 2 é pivotado em relação ao vértice v_2 resultando no simplexo 4.

Todas as regras de pivotamento da triangulação J_1^a podem ser encontrados nos trabalhos de Castelo

et al. [7, 6].

Partição da unidade implícita - PUI

Neste capítulo são apresentados alguns conceitos presentes em trabalhos referentes a funções implícitas e ao método de Partição da Unidade Implícita em Multinível para representação de superfícies por nuvens de pontos, proposto por Gois et al. [18, 19] (Ver Figura 3.1). Tal método foi denominado pelo autor em sua tese [17] como “PUI em duas vias”, por apresentar características adaptativas de duas formas distintas. Neste trabalho utilizamos este método para gerar a função implícita a partir de um conjunto de pontos no espaço (nuvem de pontos).

3.1 Funções implícitas

Em aplicações de simulação e visualização é frequente o uso de superfícies representadas tanto na forma parametrizada como na forma implícita. Para propósitos computacionais a forma parametrizada é representada por uma superfície linear por partes (PLC - Piecewise Linear Complex) – uma malha poligonal – enquanto a forma implícita é representada geralmente por um conjunto de dados volumétricos (escalares), e uma coleção de pontos em $\mathbb{R}^3$, sendo que cada ponto tem um valor de um campo escalar f associado ao conjunto de dados. A passagem de uma forma para outra é necessária em alguns casos: a forma implícita é compacta e apropriada em algumas situações, mas a forma parametrizada

permite cálculos computacionais tais como análise estática e o cálculo de métodos de elementos finitos de forma eficiente.

A transformação de uma forma implícita para uma forma paramétrica é popularmente conhecida como “problema de extração de isosuperfície”.

Seja $f: \mathbb{R}^4 \rightarrow \mathbb{R}$ uma função suave de classe C^2 e $P \subset \mathbb{R}^3$ um conjunto de pontos. Definimos o conjunto de dados volumétricos como um subconjunto do gráfico de f , ou seja, $\hat{f} = \{(p, f(p)) \mid p \in P\}$. Assume-se que $S = f^{-1}(\kappa)$ é uma superfície compacta. O conjunto S é a *isosuperfície* de valor κ para f e κ é o *isovalor* sobre o qual S é definido. O problema de extração de isosuperfície procura determinar uma malha poligonal para S – uma representação paramétrica da superfície implícita definida pelo conjunto $\{x \in \mathbb{R}^3 \mid f(x) = \kappa\}$. Note que não é possível representar S apenas com os dados de $\hat{f}$. Por isso, é necessário interpolar ou aproximar uma função g a partir de $\hat{f}$ e assumir que $S = f^{-1}(\kappa) = g^{-1}(\kappa)$.

3.2 Partição da unidade implícita em multinível

O método de Partição da Unidade Implícita em Multinível (MPUI - Multi-level Partition of Unity Implicitity), originalmente proposto por Ohtake et al. [34], é um método de representação de superfícies por nuvens de pontos. O método gera uma função implícita global composta por funções locais que aproximam a superfície amostrada pelo conjunto de pontos P . A superfície será então aproximada pela isosuperfície S de isovalor $\kappa = 0$. Vale ressaltar que este método requer a existência de vetores normais orientados para cada ponto da amostra.

Assim como outros métodos de aproximação por funções implícitas, MPUI é definido como o conjunto de nível zero de uma função F . Esta função (global) F é definida em um domínio finito Ω e é obtida de uma soma linear de aproximações locais com pesos. Por isto, é definido um conjunto de funções-peso não-negativas com suporte compacto:

$$\Phi = \{\phi_1, \dots, \phi_n\}, \quad (3.1)$$

satisfazendo a propriedade

Figura 3.1: Reconstrução da estátua Lucy, de Stanford (16 milhões de pontos). (Figura extraída do trabalho Gois et al. [17, 18])

$$\sum_{i=0}^n \phi_i(x) \equiv 1, x \in \Omega, \quad (3.2)$$

que é denominada por partição da unidade, e também é definido um conjunto

$$\mathbb{F} = \{f_1, \dots, f_n\} \quad (3.3)$$

de funções (com sinal) de distância locais definida em Ω . Dado o conjunto $\mathbb{F}$ e Φ , a função global $F : \mathbb{R}^3 \rightarrow \mathbb{R}$ é definida como:

$$F(x) \equiv \sum_{i=0}^n f_i(x) \phi_i(x), x \in \Omega. \quad (3.4)$$

A partição da unidade pode ser definida a partir de um conjunto de funções não-negativas com suporte compacto θ e expressa como:

$$\phi_i(x) = \frac{\theta_i(x)}{\sum_{k=1}^n \theta_k(x)}, \quad (3.5)$$

onde $\theta_i(x)$ é uma função peso com suporte compacto.

Dessa forma, a partir da definição da função F tem-se que a superfície reconstruída é definida implicitamente pelo conjunto $S = \{x \in \mathbb{R} \mid F(x) = 0\}$.

A função implícita aproximada F avaliará um ponto da grade e dará sua posição relativa na superfície. Para um dado ponto da grade p , se $F(p)$ é positivo então p está fora da superfície, se é negativo, p está dentro da superfície e se $F(p) = 0$ então p está na superfície.

Diferentemente de [34], onde o domínio é subdividido usando uma octree, no trabalho de Gois et al. [18] o domínio é subdividido usando a grade da triangulação J_1^a (apresentado na seção 2.2) e um suporte esférico é definido para cada bloco.

Neste método, as funções locais $f_i(x)$ são funções polinomiais calculados por meio de mínimos-quadrados. É definido um sistema de coordenadas locais e uma base de polinômios ortogonais segundo o produto interno induzido pelas equações normais, e então, a aproximação local (função polinomial) é calculada através do método de mínimos-quadrados.

O uso da triangulação J_1^a no trabalho proposto em [18] permite a adaptatividade do método, que está relacionada com o erro nas aproximações locais. Se em um bloco o erro nas aproximações locais é menor que um valor predefinido então o bloco é refinado, configurando assim a adaptatividade. Para cada novo bloco criado as aproximações locais devem obedecer ao critério de erro, caso contrário, o refinamento é feito novamente.

Por outro lado, o uso de bases de polinômios ortogonais para as aproximações locais permite uma adaptatividade também para as funções locais em cada bloco. Quando um bloco é refinado, a solução é refinada localmente de forma recursiva, ou seja, a aproximação obtida antes de refinar o bloco é

aproveitada para gerar a nova aproximação, sem adicionar um custo computacional elevado. Estas características adaptativas do método foram originalmente denominadas pelos autores do trabalho em [18] como “adaptatividade em duas vias”.

Poligonalização da superfície implícita

O poligonalizador desenvolvido por Gois e Polizelli em [19, 18, 17, 37] extrai uma malha superficial a partir da triangulação J_1^a e da isosuperfície obtida através da função implícita gerada pelo método de Partição da Unidade Implícita - PUI (Capítulo 3). Tal poligonalizador, semelhantemente aos métodos baseados em Marching Cubes [32, 20, 33, 13], obtém a representação paramétrica do objeto amostrado por uma nuvem de pontos através da intersecção dos elementos (neste caso, tetraedros) da decomposição espacial com a isosuperfície. Os elementos da malha superficial gerada por esta intersecção são triângulos que, em muitos casos, não tem uma qualidade boa devido a região em que a isosuperfície intersecta um tetraedro. Gois e Polizelli em seu trabalho [19] desenvolveram um processo baseado em heurística para deslocar os vértices da triangulação J_1^a na tentativa de melhorar a malha extraída pelo poligonalizador.

A ideia desenvolvida neste trabalho, por sua vez, também busca melhorar a malha superficial deslocando os vértices dos tetraedros da triangulação J_1^a , porém, o faz através de um processo de minimização de uma função energia. Este processo é chamado na literatura de “smoothing” ou suavização.

4.1 Melhoramento de malhas superficiais

Melhorar a qualidade de uma malha de superfície é um aspecto importante em muitas simulações numéricas tridimensionais. Um exemplo de aplicação é o problema de fronteira imersa onde a superfície se desenvolve e se transforma com o tempo devendo ser adaptada para manter a estabilidade numérica, a precisão e a eficiência enquanto preserva a geometria.

Apresentamos neste capítulo uma estratégia nova para a melhoria da malha superficial obtida pelo poligonalizador em [37, 19]. Diferentemente da abordagem comum dos métodos de melhoramento de malhas, a estratégia apresentada tem como objetivo melhorar a condição dos tetraedros que são intersectados pela isosuperfície em uma etapa anterior a de geração da malha superficial pelo poligonalizador. A ideia é movimentar os vértices da malha que decompõe o espaço antes da extração da malha superficial para que a intersecção com a isosuperfície gere elementos superficiais com boa qualidade. Para isto, a intersecção deve acontecer o mais próximo possível do centro das arestas dos elementos da malha de decomposição, desde que os elementos tenham uma boa qualidade.

Supondo que a região interna do modelo ao qual a isosuperfície representa seja aquela em que, dada a função implícita F e um ponto x desta região, temos $F(x) \leq 0$.

Portanto, se o valor da função F avaliado em um ponto é menor que 0, então este ponto está no interior da região definida pela superfície, se é igual a 0, então o ponto pertence a superfície, e se o valor é maior que 0 então o ponto não pertence ao interior do modelo. Podemos dizer então que a intersecção de uma aresta com a isosuperfície acontecerá apenas se os valores de F nos extremos da aresta tiverem sinais diferentes. A Figura 4.1 apresenta os casos de intersecção da isosuperfície com um tetraedro.

No caso de uma malha de decomposição espacial de tetraedros, os elementos resultantes da intersecção da isosuperfície com a malha de decomposição serão definidos pelos pontos de intersecção da isosuperfície com as arestas dos tetraedros que decompõe o espaço. A intersecção de um tetraedro da malha com a isosuperfície pode acontecer de 10 maneiras diferentes como apresentado na Figura 4.1. Os casos que nos interessam são apenas aqueles em que pelo menos uma aresta do tetraedro é “cortada” pela isosuperfície (os tetraedros em destaque na Figura 4.1). Nestes casos, a qualidade

Figura 4.1: Os tipos de intersecção de uma isosuperfície com um tetraedro. As arestas vermelhas formam os triângulos gerados pela intersecção. A área pintada em azul indica a representação poligonal da isosuperfície. Os vértices sinalizados com “+” estão do lado de fora, os vértices sinalizados com “-” estão do lado de dentro do modelo, e os vértices sinalizados com “o” indicam que a isosuperfície passa por aquele vértice.

dos triângulos da superfície dependerá da posição dos pontos de intersecção nas arestas do tetraedro intersectado. Podemos supor então que a melhor posição para a intersecção da isosuperfície com um tetraedro seria exatamente no ponto médio de cada aresta do tetraedro ou nos vértices dos tetraedros. A ideia principal é movimentar os vértices dos tetraedros de forma que a intersecção aconteça o mais próximo possível do ponto médio das arestas do tetraedro ou exatamente nos seus vértices.

Com base nesta ideia é apresentada abaixo a formalização e apresentação de um algoritmo baseado em suavização por otimização para o melhoramento de malhas superficiais extraídas a partir de uma função implícita. Note que o processo de suavização de malhas é aplicado na malha de decomposição espacial (triangulação J_1^a) e não na malha superficial, e os deslocamentos são realizados apenas nos vértices dos tetraedros que são intersectados pela isosuperfície. Tal processo busca melhorar a condição dos elementos intersectados pela isosuperfície, permitindo que o processo de poligonalização gere melhores elementos.

É apresentada também a técnica de deslocamento desenvolvida no trabalho de Gois et al. [19] que é baseada em heurística. Tal técnica também busca uma melhor posição para a intersecção da isosuperfície com as arestas dos elementos da decomposição (no caso, a triangulação J_1^a), porém,

não leva em consideração o fato de acontecer uma melhoria na malha caso a isosuperfície intersecte um elemento exatamente em seus vértices. Isto é devido ao fato de que, no desenvolvimento de tal técnica, os vértices da decomposição espacial são perturbados de modo a evitar a intersecção com a isosuperfície.

4.1.1 Melhoramento por heurística

A ideia da técnica de melhoramento proposto por Gois et al. [19] consiste em “afastar” os vértices da triangulação J_1^a da isosuperfície a uma taxa inversamente proporcional ao valor da função F no vértice, de forma a melhorar a qualidade dos triângulos. A seguinte equação apresenta como um vértice x é redefinido em sua nova posição x_{new} :

$$x_{new} = x + \left[\frac{\text{sign}(F(x))}{\|\nabla F\|} \left(\frac{|F_{max}| - |F(x)|}{|F_{max}|} \right)^2 \frac{e(l)}{4} \alpha \right] \nabla F(x), \quad (4.1)$$

onde $\text{sign}(F(x))$ representa o sinal da função, F_{max} é uma estimativa para o máximo da função no domínio, l é o nível de refinamento da triangulação associado ao vértice, $e(l)$ é o tamanho da aresta do bloco da triangulação J_1^a no nível de refinamento l e $0 < \alpha < 1$ determina a amplitude máxima de movimento permitida sem invalidar a triangulação. A equação (4.1) tem por finalidade movimentar um vértice na direção do gradiente, porém no sentido do sinal da função. O termo $(\frac{|F_{max}| - F(x)}{F_{max}})^2$ normaliza o deslocamento permitido por um vértice, enquanto que o termo $\frac{e(l)}{4} \alpha$ define a amplitude máxima de deslocamento de um vértice da triangulação J_1^a .

O procedimento de deslocamento é aplicado apenas para os vértices pertencentes aos tetraedros que intersectam a isosuperfície durante a poligonalização. Tal fato implica a desnecessidade de memória extra, e o maior esforço computacional adicionado ao poligonalizador se deve ao cálculo de F_{max} e à avaliação do gradiente da função em cada deslocamento de vértice. Um resultado obtido com esta técnica é apresentado na Figura 4.2.

4.1.2 Melhoramento por otimização

Para um melhor entendimento da técnica baseada em otimização apresentada nesta seção é necessário antes a formalização de alguns conceitos.

4.1.2.1 Definições

Seja a função $F : \mathbb{R}^3 \rightarrow \mathbb{R}$ e uma triangulação $\mathcal{T}$ qualquer de decomposição do espaço. Seja $S = \{x \in \mathbb{R}^3 | F(x) = 0\}$ uma superfície compacta que correspondente ao conjunto de nível 0 da função F .

Denotaremos os vértices da triangulação por $\{x_i\}_{i=1}^N \subset \mathbb{R}^3$ e as arestas não orientadas por $\overline{e_{ij}} = \{x_i, x_j\}$. Seja $\mathcal{E}$ o conjunto representando as arestas de $\mathcal{T}$. Então, seja

$$\mathcal{E}(S) = \{\overline{e_{ij}} \in \mathcal{E} \mid F(x_i) \cdot F(x_j) \leq 0\} \quad (4.2)$$

representando as arestas em $\mathcal{T}$ que interceptam S em seu interior, e para um dado vértice x_i seja $\mathcal{N}_i$ o conjunto dos índices de seus vizinhos em $\mathcal{E}(S)$, $\mathcal{N}_i = \{j \mid \overline{e_{ij}} \in \mathcal{E}(S)\}$.

Um tetraedro tem uma boa razão de aspecto se tem formato próximo ao equilátero, ou seja, com ângulos diedrais próximos de 70° .

É fácil perceber que, se os tetraedros em $\mathcal{T}$ tem razão de aspecto boa, então a triangulação da superfície S induzida por $\mathcal{T}$ terá uma boa qualidade se cada um dos pontos $z_{ij} \in \overline{e_{ij}}$ for igual a x_i ou x_j , ou é igual a $(x_i + x_j)/2$. Podemos visualizar essa idéia supondo que o tetraedro na figura 4.1 seja um tetraedro equilátero. Sendo assim, se as intersecções da superfície estão nos pontos extremos ou nos pontos médios das arestas do tetraedro então os elementos gerados pelas intersecções terão boa qualidade (triangulos em azul na figura 4.1).

Hipótese 1. *A triangulação $\mathcal{T}$ é suficientemente refinada de forma a garantir que para cada aresta $\overline{e_{ij}} \in \mathcal{E}(S)$ existe exatamente um ponto $z_{ij} \in \overline{e_{ij}}$ para o qual $F(z_{ij}) = 0$.*

4.1.2.2 Algoritmo do gradiente

Para escrever o algoritmo do gradiente vamos definir uma função “energia” $w : \mathbb{R} \rightarrow [0, \infty)$ tal que

$$w(d) = (1/2) ((d(d - 1/2)(d - 1))^2). \quad (4.3)$$

Esta função satisfaz as seguintes condições

$$w(d) = w(1 - d), \quad w(d) \geq 0, \quad w(0) = w(1/2) = w(1) = 0. \quad (4.4)$$

O objetivo é que a superfície intersecte uma aresta da triangulação ou nas extremidades de uma aresta ou no seu ponto médio.

Seja $X = (x_1, x_2, \dots, x_N) \in \mathbb{R}^{3N}$ os vértices de $\mathcal{T}$, então definimos $W : \mathbb{R}^{3N} \rightarrow [0, \infty)$ tal que

$$W(X) = \sum_{\overline{e_{ij}} \in \mathcal{E}(S)} w\left(\frac{|x_i - z_{ij}|}{|x_i - x_j|}\right) \equiv \sum_{\overline{e_{ij}} \in \mathcal{E}(S)} w(d_{ij}/h_{ij}). \quad (4.5)$$

Note que

$$d_{ij} = |x_i - z_{ij}| = |x_i - x_j| - |x_j - z_{ij}| = h_{ij} - d_{ji},$$

então

$$w(d_{ij}/h_{ij}) = w(1 - d_{ji}/h_{ij}) = w(d_{ji}/h_{ij}),$$

isto é independente da representação de $\overline{e_{ij}} \in \mathcal{E}(S)$.

Se x_i é um vértice em $\mathcal{E}(S)$, o gradiente de W com respeito a x_i pode ser calculado da seguinte forma,

$$\nabla_{x_i} W(X) = \sum_{k \in \mathcal{N}_i} w'(d_{ik}/h_{ik}) \nabla_{x_i} (d_{ik}/h_{ik}) \quad (4.6)$$

$$= \sum_{k \in \mathcal{N}_i} w'(d_{ik}/h_{ik}) \left(-\text{sign} \left(1 - \frac{F(x_k)}{F(x_k) - F(x_i)} \right) \left(\frac{F(x_k) [\nabla F(x_i)]}{(F(x_k) - F(x_i))^2} \right) \right). \quad (4.7)$$

Descrevemos a seguir como é feito o cálculo de $\nabla_{x_i} (d_{ik}/h_{ik})$.

A distância $d_{ik} = |x_i - z_{ik}|$ depende dos valores x_i e x_k , pois o ponto z_{ik} pertence a aresta formada por x_i e x_k . Reescrevendo z_{ik} em função de x_i e x_k temos:

$$z_{ik} = ax_i + (1 - a)x_k, \quad (4.8)$$

sendo a um parâmetro a ser encontrado. Então, considere que

$$\frac{d_{ik}}{h_{ik}} = \frac{|x_i - z_{ik}|}{|x_i - x_k|} = \frac{|x_i - ax_i + (1 - a)x_k|}{|x_i - x_k|} = \frac{|(1 - a)(x_i - x_k)|}{|x_i - x_k|} = |1 - a|. \quad (4.9)$$

Supondo que

$$F(z_{ik}) = \lambda_i F(x_i) + \lambda_k F(x_k) = 0$$

e

$$\lambda_i + \lambda_k = 1,$$

i.e.,

$$\begin{pmatrix} 1 & 1 \\ F(x_i) & F(x_k) \end{pmatrix} \begin{pmatrix} \lambda_i \\ \lambda_k \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

temos então um sistema linear. Aplicando a regra de Cramer para resolvê-lo, temos

$$\lambda_i = \frac{F(x_k)}{F(x_k) - F(x_i)}$$

e

$$\lambda_k = \frac{-F(x_i)}{F(x_k) - F(x_i)}.$$

Logo, podemos dizer que

$$F(z_{ik}) = \lambda_i F(x_i) + \lambda_k F(x_k) = 0 \Rightarrow \quad (4.10)$$

$$\Rightarrow z_{ik} = \lambda_i x_i + \lambda_k x_k \quad (4.11)$$

e assim de (4.8) e (4.11) segue que

$$a = \lambda_i = \frac{F(x_k)}{F(x_k) - F(x_i)}. \quad (4.12)$$

Portanto, conhecendo o valor de a e substituindo em (4.9) temos

$$|1 - a| = \left| 1 - \frac{F(x_k)}{F(x_k) - F(x_i)} \right|.$$

Agora, reescrevendo $\nabla_{x_i}(d_{ik}/h_{ik})$, segue:

$$\nabla_{x_i}(d_{ik}/h_{ik}) = |1 - a|' = \left| 1 - \frac{F(x_k)}{F(x_k) - F(x_i)} \right|' \quad (4.13)$$

$$= -\text{sign} \left(1 - \frac{F(x_k)}{F(x_k) - F(x_i)} \right) \left(\frac{F(x_k)[\nabla F(x_i)]}{(F(x_k) - F(x_i))^2} \right). \quad (4.14)$$

O algoritmo descrito abaixo minimiza W – ou, pelo menos, encontra um ponto estacionário – pelo método dos gradientes, conhecido também como método da máxima descida.

Repetir os passos abaixo ate atingir a convergência:

Passo 1: Seja x_i o vértice em $\mathcal{E}(S)$ para o qual o valor $\sum_{k \in \mathcal{N}_i} w(d_{ik}/h_{ik})$ é grande.

Passo 2: Calcule $g = \nabla_{x_i} W(X) / |\nabla_{x_i} W(X)|$.

Passo 3: Determine a distância máxima r na qual o vértice x_i pode mover-se de forma que a triangulação $\mathcal{T}$ seja válida.

Passo 4: Faça $x_i \leftarrow x_i - \alpha g$ onde $0 \leq \alpha \leq r$ é escolhido de forma a minimizar a função

$$\hat{W}(\alpha) = \sum_{k \in \mathcal{N}_i} w \left(\frac{|x_i - \alpha g - z_{ik}|}{|x_i - \alpha g - x_k|} \right). \quad (4.15)$$

Passo 5: Atualize $\mathcal{E}(S)$ (e consequentemente os valores z_{ik} , d_{ij} , h_{ij} etc.) quando necessário.

Consideramos uma triangulação válida aquela em que todos os seus elementos tem volume positivo, utilizando a forma orientada para o cálculo do volume.

No Passo 3 acima, a distância máxima r é a menor das distâncias entre o vértice x_i e a face oposta a x_i dentre todos os tetraedros incidentes a ele. Dessa forma é garantida a validade da triangulação independentemente da direção de deslocamento g .

O algoritmo usado para encontrar o valor α que minimiza a função $\hat{W}$ (no Passo 4) é o algoritmo de Brent (tirado do livro “*Numerical Recipes*” [38]), usado para encontrar o ponto de mínimo local de uma função.

Notemos que este método não utiliza nenhum cálculo para obter a qualidade do elemento que será gerado pelo poligonalizador após o deslocamento obtido. O algoritmo se apóia apenas na idéia citada anteriormente de que a superfície S deve intersectar as arestas de $\mathcal{E}(S)$ nos seus pontos extremos ou em seu ponto médio.

As figuras 4.2(a)-(d) apresentam os resultados obtidos com o modelo Bunny de Stanford pelo poligonalizador sem melhoramento (figuras 4.2(a)-(b)) e com melhoramento (figuras 4.2(c)-(d)) propostos no trabalho de Gois et al. [19]. Nas figuras 4.2(e)-(f) é apresentado o resultado obtido com o método proposto neste capítulo. A medida usada para aferir a qualidade dos triângulos do modelo apresentado na Figura 4.2 é a medida da “razão dos raios” $R/2r$, onde R é o raio do circuncírculo do triângulo e r é o raio do círculo inscrito do triângulo. O valor da razão para um triângulo equilátero é igual a 1, e também o menor valor possível para esta razão. Portanto, quanto mais próximo de 1 é a medida desta razão, mais próximo do equilátero é o triângulo e melhor é a sua qualidade.

Figura 4.2: As figuras (a) e (b) apresentam o modelo Bunny de Stanford gerado pelo poligonalizador do trabalho de Gois et al. [19] sem o melhoramento proposto originalmente. As figuras (d) e (e) apresentam o mesmo modelo gerado pelo poligonalizador com o melhoramento proposto também em [19]. As figuras (g) e (h) apresentam o mesmo modelo gerado com o melhoramento de malhas superficiais proposto neste capítulo. A escala de cores indica o vermelho como melhor qualidade ($R/2r \approx 1$) e o azul como pior qualidade ($R/2r > 5$). Acompanham as figuras os seus respectivos histogramas de qualidade.

4.2 Algumas considerações

O trabalho apresentado em [13] busca deslocar os vértices de uma forma diferente, pois é baseado no algoritmo de Marching Cubes [32], que trabalha com uma malha hexaedral para decompor o espaço. Nesse trabalho o deslocamento dos vértices é feito, em um primeiro passo, para esticar as arestas que são intersectadas pela superfície de forma que os vértices destas arestas fiquem distantes da isosuperfície. Depois, em um segundo passo, transformações são feitas para ajustar as direções das arestas no sentido normal a superfície. Dessa forma a extração da malha superficial gerará elementos de melhor qualidade. No entanto é uma técnica baseada em heurística assim como a proposta apresentada por Gois et. al [19] que também desloca os vértices na direção do vetor normal à superfície, porém de forma mais simples.

Melhoramento de malhas tetraedrais

A maioria dos trabalhos de melhoramento de malhas de tetraedros utilizam três tipos de operações em malha:

1. *Suavização de malha*: realocação dos pontos da malha de forma a melhorar a qualidade mas sem mudar sua topologia.
2. *Operações topológicas*: mudança nas conexões entre os vértices da malha, sem alterar a posição dos vértices.
3. *Inserção de vértices*: adição de vértices extras na malha (através de divisão de arestas, faces ou tetraedros) e subsequente criação e ligação de elementos nas regiões da malha afetadas pelo novo vértice.

Outra técnica para melhoramento de malhas baseada em suavização é a aplicação de molas virtuais nos elementos da malha. A aplicação de molas virtuais em malhas é encontrada nos estudos sobre malhas dinâmicas. Malhas dinâmicas são comumente utilizadas em problemas de simulação sobre domínios cuja geometria varia com o tempo. Neste trabalho a aplicação de molas virtuais funciona

apenas como uma operação de suavização da malha. Neste caso, os pontos da malha são realocados de acordo com as molas virtuais aplicadas sobre cada aresta da malha.

A seguir detalhamos cada uma destas operações.

5.1 Melhoramento de malhas por suavização

Os métodos de suavização de malhas tem como objetivo melhorar a qualidade de uma malha deslocando os seus vértices sem mudar sua topologia (Figura 5.1). Em geral, os algoritmos de suavização de malhas utilizam técnicas locais que ajustam a posição de cada vértice individualmente para melhorar a qualidade dos elementos incidentes ao vértice.

Existem na literatura dois tipos principais de métodos para melhoramento de malhas baseados em suavização, os baseados na suavização Laplaciana e os baseados em otimização.

Suavização Laplaciana: Os métodos baseados na suavização Laplaciana [26, 15] buscam posicionar os vértices no centro geométrico do polígono determinado pelos seus vértices incidentes (Figura 5.1). Este método é computacionalmente barato, mas opera de forma heurística e não garante a melhora na qualidade dos elementos. De fato, é possível produzir uma malha inválida, ou seja, com elementos invertidos ou de área negativa. O uso deste tipo de suavização em alguns casos pode eliminar características finas do objeto representado pela malha superficial, porém, é um método muito utilizado devido ao custo computacional barato e à facilidade de implementação.

Otimização: Os métodos baseados em otimização [1, 36, 41] evitam a criação de elementos inválidos e buscam uma localização ótima para os vértices da malha utilizando técnicas de otimização. Estas técnicas geralmente buscam determinar o mínimo local de uma função-base aplicada sobre os elementos de uma região da malha. Esta função-base é criada a partir de um critério de qualidade dos elementos da malha e são construídas de forma que o seu valor seja mínimo se os elementos da malha estão próximos do desejado, ou seja, são elementos com boa qualidade. Estas técnicas tem a vantagem de oferecer a garantia de melhoramento e validação das malhas. Entretanto, esta garantia demanda um custo computacional maior em comparação com o método por suavização de Laplace.

Figura 5.1: Exemplo 2D de melhoria de malha por Suavização: o vértice é deslocado para uma região do polígono determinado pelos triângulos incidentes ao vértice para melhorar a qualidade destes triângulos.

5.2 Melhoria de malhas por operações topológicas

O método de melhoria de malhas por transformações topológicas funciona por meio de operações que removem elementos de uma malha substituindo-os por um conjunto diferente de elementos, de melhor qualidade, ocupando o espaço deixado pelos elementos removidos. O preenchimento do espaço deixado por esses elementos pode ser realizado de maneira arbitrária ou através de um conjunto de *operações topológicas* descritas abaixo:

- flip 2-3, e o seu inverso, flip 3-2 (ver Figura 5.2).
- flip 4-4, e a versão para elementos de borda, flip 2-2 (ver Figura 5.2): esta operação exige que seja escolhido o par de vértices (dentre as duas opções disponíveis) no qual será feito o flip.
- Remoção de aresta (ver Figura 5.3): esta operação é uma generalização do flip 3-2, do flip 4-4 e do flip 2-2. Também pode ser realizada em arestas de fronteira. Existem várias formas de conectar os vértices dos elementos adjacentes a aresta removida. No trabalho de Freitag e Ollivier-Gooch [16] é utilizado moldes de triangulação e, já no trabalho de Klingner e Shewchuk [23, 44], os autores utilizam o algoritmo de Klinksek [22] de forma a maximizar a qualidade mínima do conjunto de tetraedros criados.

- Remoção multi-faces: operação idealizada por Cougny e Shephard [11] é o inverso da operação de remoção de aresta, mostrado na Figura 5.3. É uma generalização do flip 2-3 e do flip 4-4; necessita de programação dinâmica para selecionar o subconjunto de faces opostas a dois vértices que serão removidas, no intuito de obter a melhor configuração que melhora a qualidade dos elementos.

O trabalho de Freitag e Ollivier-Gooch [16] utiliza as primeiras três operações. Pode-se notar que a remoção multi-faces pode ser decomposta em uma sequência de um único flip 2-3 seguido de vários flips 3-2. O trabalho de Klingner e Shewchuk [23] utilizam todas as operações listadas acima.

5.3 Melhoramento de malhas por inserção de vértices

A estratégia de melhoramento por inserção de vértices mais conhecida na literatura é o algoritmo de refinamento Delaunay desenvolvido por Ruppert [43]. Neste caso, porém, a malha tem que ser uma triangulação de Delaunay e a inserção de um vértice cria novos elementos que precisam respeitar o critério de Delaunay da malha. A principal contribuição do trabalho de Klingner e Shewchuk [23] é a estratégia criada para melhorar a malha através das transformações realizadas em decorrência da inserção de um novo vértice na malha. Esses autores mostraram que o melhoramento de malhas realizado por transformações originadas pela inserção de vértices é muito mais efetiva. A localização apropriada de um vértice é um problema difícil de ser resolvido. Klingner e Shewchuk descrevem um sofisticado e complexo algoritmo para inserir vértices que é similar a inserção de vértices do algoritmo de Delaunay e, juntamente com operações de suavização baseada em otimização e operações topológicas, permite que as malhas sejam melhoradas, podendo chegar a resultados expressivos, por exemplo, com todos os ângulos diedrais entre 31° e 149° , ou, usando uma função objetiva diferente, entre 23° e 136° .

5.4 Suavização de malhas através de molas virtuais

Para realizar a suavização sobre a malha através de molas virtuais devemos compreender os tipos de molas virtuais utilizadas comumente nos trabalhos de malhas dinâmicas.

Existem vários tipos de molas virtuais, porém os principais são os seguintes:

Figura 5.2: Operações topológicas: flip 2-3 e a operação inversa o flip 3-2, o flip 4-4 e o flip 2-2.

Figura 5.3: Remoção de arestas e Remoção multi-faces: nesta figura 2 operações topológicas são ilustradas, sendo uma o inverso da outra, a remoção de aresta elimina uma aresta do conjunto de tetraedros e acrescenta novas faces para gerar um novo conjunto de elementos, enquanto que a remoção multifaces, de forma contrária, elimina as faces e acrescenta uma aresta.

- Molas longitudinais: agem sobre arestas e controlam seus comprimentos.
- Molas torcionais: agem sobre os ângulos entre arestas e controlam áreas (volumes) de triângulos (tetraedros).
- Molas semi-torcionais: abordagem simplificada para mola torcional; assim como as molas torcionais, controlam áreas ou volumes.

I. P. Soares [47] explorou estes tipos de molas e criou outra para tratar os problemas de movimento de malhas e remalhamento de malhas superficiais.

5.4.1 Molas longitudinais

As molas longitudinais foram propostas por Batina [2]. O autor propôs que os coeficientes de rigidez das molas sejam valores inversamente proporcionais ao comprimento de suas respectivas arestas. Dessa forma, quanto menor a aresta, mais rígida é a mola, permitindo um controle sobre seu comprimento. Assim, para uma aresta conectando os vértices v_i e v_j , o coeficiente de rigidez da mola, denotado por κ_{ij} , é dado por:

$$\kappa_{ij} = \frac{a}{(l_{ij})^b}, \quad (5.1)$$

onde l_{ij} é o comprimento da aresta ligando v_i a v_j e a e b são constantes (geralmente iguais a 1) (ver Figura 5.4).

Figura 5.4: Ilustração de uma mola longitudinal.

Como o conceito de molas longitudinais está relacionado ao comprimento das arestas, não há diferenças significativas na aplicação de molas em malhas de triângulos (caso 2D) ou em malhas de tetraedros (caso 3D).

5.4.2 Molas torcionais

Em uma malha, as molas torcionais são uma maneira de controlar os ângulos entre as arestas, e consequentemente, controlar as áreas dos elementos da malha. A utilização de molas torcionais no problema de malhas dinâmicas foi proposto por Farhat et al. [14]. Considere o triângulo t_{ijk} cujos vértices são designados por v_i , v_j e v_k . Seja θ_i^{ijk} o ângulo entre as arestas e_{ij} e e_{ik} . Seja $A(t_{ijk})$ a área do triângulo t_{ijk} . O coeficiente da mola torcional no vértice v_i do triângulo t_{ijk} é dado por:

$$C_i^{ijk} = \frac{1}{1 + \cos(\theta_i^{ijk})} \frac{1}{1 - \cos(\theta_i^{ijk})} = \frac{1}{\sin(\theta_i^{ijk})^2} = \frac{(e_{ij})^2 (e_{ik})^2}{4A(t_{ijk})^2}. \quad (5.2)$$

Diferentemente das molas longitudinais que agem sobre arestas, as molas torcionais agem sobre cada vértice da malha, mais precisamente em cada ângulo incidente ao vértice (caso 2D) (ver Figura 5.5).

Figura 5.5: Ilustração de uma mola torcional [47].

A extensão da ideia de molas torcionais do caso 2D em triângulos para o caso 3D em tetraedros não é direta. No caso 2D o volume do triângulo é controlado através dos ângulos em cada vértice, porém, no caso 3D é preciso controlar o volume do tetraedro. Como os tetraedros possuem ângulos sólidos que são de difícil controle, a tarefa de utilizar molas torcionais em elementos 3D não é trivial.

5.4.3 Molas semi-torcionais

Como a abordagem de molas torcionais agrega informação de ângulos sobre cada vértice da malha, incorporar este conceito envolve uma complexidade maior do que a necessária quando se utiliza molas longitudinais. A mola semi-torcional é uma abordagem simplificada para agregar a informação dos ângulos num sistema composto por molas longitudinais. A ideia por trás das molas semi-torcionais é fazer com que os ângulos influenciem os coeficientes das molas longitudinais, sendo assim, o sistema contém, na realidade, apenas molas longitudinais. A ideia de molas semi-torcionais proposta por Zeng e Ethier [52] consiste em adicionar diretamente o coeficiente da uma mola torcional ao coeficiente da mola longitudinal oposta. Dessa forma, o coeficiente de rigidez longitudinal total κ_{ij}^{total} da mola ligando os vértices v_i e v_j é dado por (ver Figura 5.6):

$$\kappa_{ij}^{total} = \kappa_{ij} + C_k^{ijk} + C_l^{ijl}, \quad (5.3)$$

onde C é o coeficiente de rigidez da mola torcional apresentado na Equação (5.2).

Figura 5.6: Mola semi-torcional [47].

A aplicação das molas semi-torcionais em tetraedros (caso 3D) é mais simples que no caso da mola torcional. Para tetraedros o cálculo do coeficiente da mola semi-torcional oposta é feito substituindo-se o ângulo entre arestas pelo ângulo diedral na Equação (5.2). Dessa forma, no caso 3D, as molas semi-torcionais controlam os ângulos entre faces ao invés de ângulos entre arestas. Considere a Figura 5.7, onde uma mola semi-torcional é utilizada para controlar o ângulo diedral entre as faces t_{lij} e t_{kji} . A mola semi-torcional ϕ_{ij}^{kl} está presente no ângulo diedral na aresta e_{ij} e age sobre a mola longitudinal presente na aresta e_{kl} .

Figura 5.7: Mola semi-torcional em tetraedros.

5.4.4 A estratégia de suavização de malhas com molas virtuais

A aplicação de molas para malhas dinâmicas envolve a teoria de forças da Física, sendo necessário gerar um sistema de equações a ser resolvido para se manter o equilíbrio do sistema de molas. Em [47] o autor desenvolve a teoria da Física e mostra a maneira como o sistema de molas deve ser montado para aplicação em uma malha.

A realocação dos vértices da malha aplicando as molas virtuais funciona como um método de suavização Laplaciana. Na verdade, o método clássico de suavização Laplaciana é um caso particular do método de suavização utilizando molas virtuais, onde as molas assumem inicialmente tamanho igual a zero e coeficientes de rigidez únicos [53].

Em outras palavras, no método de suavização Laplaciana podemos considerar a malha como um sistema de molas, onde cada aresta que conecta os vértices da malha pode ser considerado uma mola longitudinal. Dado um vértice na malha, os elementos incidentes a este vértice formam um polígono. Se o coeficiente de rigidez das molas forem iguais, o sistema de molas terá como ponto de equilíbrio exatamente o centro geométrico do polígono [53] (ver Figura 5.8).

Em um sistema composto de molas, as forças atuantes em um vértice podem ser representadas pelo modelo massa-mola-amortecimento. Nesse modelo temos o seguinte sistema de equações diferenciais ordinárias de segunda ordem, obtida a partir da segunda lei de Newton:

$$M\ddot{\Delta} + D\dot{\Delta} + K\Delta = 0, \quad (5.4)$$

onde Δ é o vetor deslocamento dos vértices da malha, M é a matriz de massa associada aos vértices, D é a matriz dos coeficientes de amortecimento e K é a matriz de coeficientes de rigidez das molas.

Assumindo que todos os vértices possuem mesma massa e o mesmo coeficiente de amortecimento, o problema pode ser simplificado em um sistema conhecido como quase-estático [14, 4], dado por

$$F_K = K\Delta = 0. \quad (5.5)$$

Nessa forma simplificada o sistema se tornou linear com os deslocamentos dos vértices como incógnitas, sendo que na matriz K estão os coeficientes das molas no sistema. Assim como em [47], utilizaremos esta formulação para o sistema de molas implementado neste trabalho.

5.4.4.1 O algoritmo iterativo para suavização da malha

Os métodos para a solução do sistema de equações (5.5) diferem dependendo do tipo de analogia utilizada para o sistema de molas, que por sua vez depende do tipo de aplicação. Blom [4] apresenta dois tipos de abordagem, uma para utilização em malhas dinâmicas e outra para aplicações em suavização de malhas. Em [47] a aplicação é para malhas dinâmicas, enquanto neste trabalho o objetivo é realizar a suavização da malha. Para suavização de malhas, Blom [4] apresentou uma analogia chamada de “vertice-mola”. Nessa abordagem não há movimento inicial dos elementos de borda da malha. O sistema pode ser resolvido iterativamente pelo método de Gauss-Seidel.

A cada iteração, a nova posição do vértice v_i é calculada como uma média ponderada dos vértices vizinhos. Os pesos são dados pelos coeficientes de rigidez κ_{ij} (ver Figura 5.8). Sendo assim, a equação a ser resolvida em cada iteração é dada da seguinte forma:

$$v_i^{t+1} = \frac{\sum_{j=0}^k \kappa_{ij} v_j^t}{\sum_{i=0}^k \kappa_{ij}}, \quad (5.6)$$

onde v_j com $0 < j < k$ são os vértices que formam uma aresta com v_i e κ_{ij} é o coeficiente da mola correspondente a aresta formada por v_i e v_j .

O algoritmo que soluciona o sistema através do método de Gauss-Seidel é apresentado a seguir. Este algoritmo percorre todos os vértices internos da malha calculando novas posições para os vértices.

Figura 5.8: Interpretação física do processo de iteração em um sistema de molas.

Algoritmo 1 Suavização da malha

---

entrada:

Malha a ser suavizada

Tolerância de convergência ϵ

repita

para cada vértice interno v_i da malha **faça**

        Calcule v_i^{t+1} de acordo com a equação (5.6)

fim para cada

enquanto $\max_v \left(\frac{\|v_i^{t+1} - v_i^t\|}{e_{min}} \right) \geq \epsilon$

O algoritmo 1 pode ser usado para resolver sistemas compostos de molas longitudinais ou de molas semi-torcionais. A equação (5.6) pode ser mudada, dependendo do tipo de mola utilizada no sistema, ou seja, o que muda nessa equação é o valor dos coeficientes das molas que compõe o sistema. Por exemplo, num sistema composto de molas torcionais e semi-torcionais o cálculo do coeficiente de rigidez é realizado de acordo com a equação (5.3)

A convergência do algoritmo é satisfeita se a seguinte condição for verdadeira:

$$\max_v \left(\frac{\|v - v_n\|}{e_{min}} \right) \geq \epsilon, \quad (5.7)$$

onde e_{min} é o comprimento da menor aresta incidente no vértice iterado v . Como descreve Soares [47] a escolha deste critério para determinar a convergência do algoritmo é baseada na característica local da malha. Desta maneira a avaliação do deslocamento é feita observando-se o tamanho da menor aresta dentre as arestas incidentes ao vértice deslocado. O deslocamento estará convergindo somente se o seu tamanho não ultrapassar uma porção determinada do tamanho da menor aresta local. Isto faz com que o critério de parada do algoritmo seja mais relaxado em regiões onde a malha é mais

grosseira e mais restritivo onde a malha é mais refinada.

Neste trabalho utilizamos o algoritmo 1 para realizar a suavização na malha de superfície e também na malha volumétrica. Entretanto, a configuração de molas usada para realizar a suavização na malha superficial é diferente da configuração escolhida na suavização da malha volumétrica. Para realizar a suavização na malha superficial foi escolhida apenas a mola longitudinal, enquanto que na suavização da malha volumétrica utilizamos a configuração de molas longitudinais com molas semi-torcionais nas arestas da malha. A escolha de uma configuração diferente para malhas volumétricas é pelo fato de que, neste caso, as molas semi-torcionais controlam o volume dos elementos. Nas seções seguintes apresentamos as estratégias utilizadas na suavização em malhas superficiais e em malhas volumétricas.

5.4.5 Suavização por molas em malhas superficiais

Ao realizar a suavização na malha superficial pelo algoritmo 1 apresentado anteriormente, os vértices podem ser reposicionados de maneira que não representam mais a superfície da malha original. Isto pode acontecer em regiões onde a curvatura da superfície é alta. Para eliminar este problema adotamos uma estratégia simples que projeta o vértice novamente na malha superficial a partir da nova posição calculada pela equação (5.6). Dessa forma um passo é acrescentado no algoritmo 1 logo após o passo que calcula a nova localização para o vértice iterado. O algoritmo 2 mostra como fica a estratégia de suavização para malhas superficiais.

Algoritmo 2 Suavização de malha superficial

---

entrada:

Malha superficial

Tolerância de convergência ϵ

repita

para cada vértice interno v_i da malha **faça**

        Calcule v_i^{t+1} de acordo com a equação (5.6)

        Atualiza a localização de v_i^{t+1} com sua projeção na superfície

fim para cada

enquanto $\max_v \left(\frac{\|v_i^{t+1} - v_i^t\|}{e_{min}} \right) \geq \epsilon$

A Figura 5.9 apresenta a malha superficial de uma elipse suavizada pelo processo descrito no

algoritmo 2. Os histogramas na Figura 5.9 mostram a qualidade dos triângulos da superfície medidas pela razão $R/2r$, onde R é a medida do raio do círculo circunscrito e r é a medida do raio do círculo inscrito do triângulo. Para um triângulo equilátero esta razão é igual a 1.

Figura 5.9: (a) e (b) malha superficial de uma elipse e histograma da medida de qualidade; (c) e (d) malha superficial suavizada e seu histograma de qualidade.

Nota-se pelos histogramas da Figura 5.9 que o número de elementos com qualidade próxima a 1 aumentou consideravelmente na malha suavizada pelo algoritmo 2.

5.4.6 Suavização por molas em malhas volumétricas

O algoritmo 1 descrito anteriormente funciona para malhas bidimensionais compostas por triângulos e para malhas tridimensionais compostas por tetraedros. Em ambos os casos os vértices de fronteira (curva no caso 2D e superfície no caso 3D) não são visitados pelo algoritmo. Neste trabalho utilizamos as molas longitudinais e semi-torcionais na execução do algoritmo de suavização em malhas volumétricas. Assim, no algoritmo 1 o coeficiente κ_{ij} é o mesmo descrito na equação (5.3).

As Figuras 5.10 apresentam a malha superficial de uma elipse suavizada pelo processo descrito no algoritmo 1. Ao lado de cada figura estão os histogramas de qualidade dos elementos das malhas aferidas pela métrica “ângulo mínimo”, que calcula o menor ângulo diedral interno de cada elemento.

Figura 5.10: Modelo de uma elipse; (a) malha volumétrica inicial; (c) a malha suavizada pelo algoritmo 1.

Ao observar os histogramas da Figura 5.10 nota-se que a malha inicial continha a maioria dos seus elementos com ângulo diedral próximo de 45° . Na malha suavizada pelo algoritmo 1, a distribuição dos elementos ficou mais uniforme, com a maioria dos elementos entre os ângulos 35° e 50° . Porém, o menor ângulo diedral encontrado na malha da Figura 5.10(a) é igual 12.5° enquanto que na malha suavizada da Figura 5.10(c) o menor ângulo diedral encontrado é igual a 17.9° .

Resultados

Neste capítulo apresentamos os principais resultados obtidos com a técnica de poligonalização da superfície implícita apresentada no capítulo 4. É apresentado também alguns resultados das técnicas de melhoramento de malhas apresentadas no capítulo 5.4 com modelos de artérias do corpo humano.

Os exemplos apresentados neste capítulo foram obtidos a partir dos modelos “*renal artery*”, “*abdominal aortic*”, “*basilar artery*”, “*iliac aneurysm*”, “*abdominal aortic aneurysm*”, fornecidos pelo projeto HeMoLab (Laboratório de Modelagem em Hemodinâmica) do LNCC - Laboratório Nacional de Computação Científica.

6.1 Resultados do melhoramento de malhas superficiais

Para cada modelo utilizado nos resultados apresentados nesta seção foram geradas malhas superficiais que, para efeito de comparação, nomeamos da seguinte forma:

- **Malha A:** Malha superficial gerada pelo poligonalizador apresentado em [37, 19] sem melhoramento.
- **Malha B:** Malha superficial extraída da malha volumétrica obtida com o gerador apresentado em [10].

- **Malha C:** Malha superficial gerada pela técnica de poligonalização da superfície apresentada no capítulo 4
- **Malha D:** Malha B melhorada pelo algoritmo de suavização de superfícies por molas virtuais apresentado no capítulo 5.4.

A qualidade das malhas mostradas na Tabela 6.1 é aferida sobre cada triângulo em termos do menor ângulo (α_{min}) e maior ângulo (α_{max}). A comparação é realizada através da porcentagem de triângulos das malhas com ângulos maiores que 30° e a porcentagem de triângulos com ângulos menores que 100° .

As figuras 6.1, 6.2 e 6.3 mostram as malhas obtidas para cada modelo com uma escala por cor da qualidade dos triângulos aferidas pela métrica “razão dos raios”. Também é apresentado um histograma da qualidade dos triângulos de cada malha aferidos por essa mesma métrica. Para esta medida de qualidade o menor (e o melhor) valor possível é 1 (cor vermelha), que representa a qualidade de um triângulo equilátero.

Para obter a função implícita utilizada na geração das malhas superficiais do modelo *abdominal aortic aneurism*, foi utilizado nível de refinamento máximo igual a 3 na triangulação J_1^a . Para os modelos *iliac aneurysm* e *basilar artery* o nível de refinamento máximo utilizado foi 4.

		$(30^\circ < \alpha_{min})$	$(\alpha_{max} < 100^\circ)$	# triângulos
<i>abdominal aortic aneurysm</i>	malha A	50.94%	70.59%	75164
	malha B	68.25%	83.84%	33250
	malha C	76.76%	80.85%	64302
	malha D	82.41%	79.93%	33250
<i>iliac aneurysm</i>	malha A	51.86%	72.49%	169032
	malha B	68.64%	84.04%	74962
	malha C	77.38%	81.35%	138898
	malha D	83.47%	82.08%	74962
<i>basilar artery</i>	malha A	48.40%	69.25%	40520
	malha B	68.05%	83.71%	54870
	malha C	75.81%	79.89%	103710
	malha D	82.79%	80.11%	54870

Tabela 6.1: Comparação da qualidade das malhas obtidas a partir dos modelos *abdominal aortic aneurism*, *iliac aneurysm* e *basilar artery*. Os valores de porcentagem indicam o número de elementos da malha com ângulos maiores que 30° ou menores que 100° .

6.1 Resultados do melhoramento de malhas superficiais

Figura 6.1: Malhas superficiais do Modelo *abdominal aortic aneurysm*.

Figura 6.2: Malhas superficiais do Modelo *iliac aneurysm*.

Figura 6.3: Malhas superficiais do Modelo *basilar artery*.

6.2 Resultados do melhoramento de malhas com molas virtuais

A principal aplicação deste trabalho é a geração de malhas volumétricas para utilização em simulações de escoamento de fluidos em artérias do corpo humano. Para este tipo de aplicação nossa abordagem não leva em consideração características finas dos modelos, pois são raros os modelos que necessitam de um tratamento neste sentido. Em geral, para simulações de fluidos em artérias, as superfícies dos modelos são suaves.

Em nosso estudo de geração de malhas para aplicações médicas, utilizamos especificamente malhas de artérias do corpo humano. Neste caso, o intuito é utilizar estas malhas para simular o escoamento sanguíneo em artérias do corpo humano. As malhas utilizadas como ponto de partida para as estratégias de melhoramento estudadas foram obtidas pelo gerador de malhas volumétricas apresentado no trabalho de Cunha et al. [10]. Neste trabalho a malha volumétrica é obtida com as ideias apresentadas em [24], porém, utilizando como decomposição espacial a triangulação J_1^a [7] apresentada no capítulo 2, e utilizando a estratégia de Partição da Unidade Implícita [18, 19] apresentada no capítulo 3 para gerar a função implícita do modelo.

Em Cunha et al. [10] as malhas são obtidas a partir de um conjunto de pontos no espaço que representa o modelo. O nome usado na literatura para este conjunto é nuvem de pontos. Uma nuvem de pontos pode ser obtida através de scanner volumétrico ou, em nosso caso, de imagens médicas.

Em nosso trabalho as nuvens de pontos utilizadas para gerar as malhas volumétricas iniciais foram obtidas de malhas superficiais. De cada malha superficial aproveitamos somente os pontos, descartando os outros elementos da malha, e obtendo uma nuvem de pontos. Com a nuvem de pontos é criada uma função implícita através do método de Partição da Unidade Implícita [19]. A função implícita de nível 0 representa a superfície do modelo. A partir da função implícita obtemos uma discretização da superfície analisando a função nos vértices dos tetraedros da triangulação J_1^a que decompõe o domínio. Os elementos da triangulação J_1^a que pertencem ao interior do modelo são aproveitados na malha volumétrica resultante. Porém, os elementos da triangulação J_1^a afetados pela intersecção com a função implícita no nível 0 precisam ser analisados e regenerados de forma que toda a malha final contenha apenas tetraedros. Este processo é feito em [10] de acordo com as ideias de

Figura 6.4: Diagrama de execução dos testes feitos para obtenção das malhas volumétricas apresentadas nesta seção.

Shewchuk em [24].

O melhoramento de malhas de Klingner e Shewchuk em [23] é utilizado nos nossos estudos pois é o estado da arte na área de melhoramento de malhas volumétricas. A ferramenta criada por eles é chamada de *Stellar* e combina as estratégias de melhoramento de malhas apresentadas no capítulo 5 para gerar uma malha melhorada. Em nosso trabalho combinamos a ferramenta *Stellar* com nossa estratégia de *suavização* por molas virtuais (Capítulo 5.4), obtendo resultados interessantes.

O diagrama da Figura 6.4 apresenta o fluxo de execução utilizado para a obtenção de cada malha apresentada nas figuras deste capítulo.

Para podermos comparar melhor os resultados as malhas foram obtidas com os mesmos parâmetros para a triangulação J_1^a . Somente no modelo *abdominal aortic aneurysm* o nível de refinamento máximo da triangulação J_1^a é igual a 3, nos demais modelos este parâmetro tem valor 4.

As Figuras 6.6, 6.8, 6.10, 6.12 e 6.14 apresentam as malhas com seus respectivos histogramas de qualidade na seguinte ordem: (a) detalhe aproximado da malha inicial; (b) detalhe aproximado da malha inicial melhorada pela ferramenta *Stellar* [23]; (c) detalhe aproximado da malha inicial melhorada pelo nosso método de *suavização* por molas virtuais; (d) detalhe aproximado da malha apresentada na Figura (d) melhorada pela ferramenta *Stellar*. Os histogramas mostram a qualidade dos elementos das malhas aferidas pela métrica “razão dos raios” para tetraedros.

A Tabela 6.2 apresenta os ângulos diedrais mínimos e máximos para cada modelo em cada es-

estratégia de melhoramento de malhas, a *supor*, a malha inicial utilizada, a malha melhorada com a ferramenta *Stellar*, a malha suavizada através da aplicação de molas virtuais e a malha obtida com a ferramenta *Stellar* após a *suavização* por molas virtuais.

	malha inicial		<i>Stellar</i>		<i>suavização</i>		<i>Stellar + suavização</i>	
	<i>min</i>	<i>max</i>	<i>min</i>	<i>max</i>	<i>min</i>	<i>max</i>	<i>min</i>	<i>max</i>
<i>renal artery</i>	7.4117	162.19	14.96	158.84	7.2185	164.8	17.988	154.57
<i>abdominal aortic</i>	7.0378	161.08	12.597	154.87	1.7102	177.37	16.927	155.11
<i>basilar artery</i>	7.219	162.49	14.492	156.17	7.1605	163.08	19.124	153.32
<i>iliac aneurysm</i>	6.544	160.52	15.545	155.72	3.7584	169.55	16.881	153.24
<i>abdominal aortic aneurysm</i>	6.7657	160.21	19.818	151.06	3.8039	167.03	22.836	147.89

Tabela 6.2: **Medidas de Qualidade.** Os valores das tabelas correspondem aos ângulos diedrais mínimos e máximos encontrados em cada uma das malhas listadas.

A Figura 6.15 apresenta o modelo *abdominal aortic aneurysm* seccionado para a visualização dos elementos internos das malhas em cada etapa ilustrada no diagrama da Figura 6.4. Podemos notar nesta figura que próximo a superfície do modelo a malha é mais fina (elementos menores) e, no interior do modelo a malha é mais grossa (elementos maiores). Isto acontece devido a adaptatividade proporcionada pela triangulação J_1^a .

Ao observarmos as figuras de todos os modelos, notamos que os elementos da superfície nas malhas suavizadas com nossa estratégia apresenta um aspecto mais uniforme, embora ainda existem elementos vizinhos com aparente diferença em seus tamanhos. Além disso podemos observar que a ferramenta *Stellar* pouco mudou o visual da malha suavizada com nossa estratégia.

Entretanto, observando a Tabela 6.2 notamos que a ferramenta *Stellar* melhorou significativamente o menor e o maior ângulo diedral presente na malha. Ainda observando a Tabela 6.2 notamos que o processo de suavização por molas virtuais aplicado na malha inicial não traz muitas vantagens, porém, ao aplicar o melhoramento *Stellar* na malha suavizada obtemos resultados melhores do que aplicando o melhoramento *Stellar* diretamente na malha inicial.

Embora a ferramenta *Stellar* pouco modificou os elementos da superfície, acontece o contrário com relação aos elementos internos da malha. Observando a Figura 6.15 notamos que a ferramenta *Stellar* modificou drasticamente o interior das malhas na qual foi aplicado o seu melhoramento. Esta diferença é aparente quando observamos a diferença no número de elementos das malhas. No exemplo mostrado na Figura 6.15 o número de elementos da malha melhorada pela ferramenta *Stellar* após

realizada a suavização diminuiu em 32,4% em relação à malha inicial. A Tabela 6.3 mostra o número de elementos para cada malha obtida nos testes aqui apresentados. Diferentemente da estratégia de melhoramento aplicada pela ferramenta *Stellar*, a estratégia de suavização por molas não modifica o número de elementos da malha inicial.

	renal artery		abdominal aortic		basilar artery		iliac aneurysm		abdominal aortic aneurysm	
	#	%	#	%	#	%	#	%	#	%
malha inicial	300823	-	750338	-	277536	-	382820	-	1140659	-
suavização	300823	0	750338	0	277536	0	382820	0	1140659	0
<i>Stellar</i>	189257	37,1	443015	41,0	174001	37,3	234422	38,8	646519	43,3
suavização + <i>Stellar</i>	178428	40,7	419013	44,2	156400	43,6	232110	39,4	770920	32,4

Tabela 6.3: **Número de elementos das malhas.** Para cada coluna o valor da esquerda (#) representa o número de elementos (tetraedros) na malha; o valor da direita (%) representa a porcentagem da diminuição no número de elementos com relação ao número de elementos da malha inicial.

Figura 6.5: Malha volumétrica do modelo *renal artery*.

(a) Malha Inicial

(b) *Stellar*

(c) *Suavização*

(d) *Stellar + Suavização*

Figura 6.6: Histogramas de qualidade das malhas do modelo *renal artery*.

Figura 6.7: Malha volumétrica do modelo *abdominal aortic*.

(a) Malha Inicial

(b) Stellar

(c) Suavização

(d) Stellar + Suavização

Figura 6.8: Histogramas de qualidade das malhas do modelo *abdominal aortic*.

Figura 6.9: Malha volumétrica do modelo *basilar artery*.

(a) Malha Inicial

(b) *Stellar*

(c) *Suavização*

(d) *Stellar + Suavização*

Figura 6.10: Histogramas de qualidade das malhas do modelo *basilar artery*.

Figura 6.11: Malha volumétrica do modelo *iliac aneurysm*.

(a) Malha Inicial

(b) *Stellar*

(c) *Suavização*

(d) *Stellar + Suavização*

Figura 6.12: Histogramas de qualidade das malhas do modelo *iliac aneurysm*.

Figura 6.13: Malhas volumétricas do modelo *abdominal aortic aneurysm*.

(a) Malha Inicial

(b) *Stellar*

(c) *Suavização*

(d) *Stellar + Suavização*

Figura 6.14: Histogramas de qualidade das malhas do modelo *abdominal aortic aneurysm*.

Figura 6.15: O modelo *abdominal aortic aneurysm* seccionado para visualização dos elementos internos da malha. A figura (a) é a malha inicial; a figura (b) é a malha inicial melhorada com a ferramenta *stellar*; a figura (c) é a malha inicial melhorada pela suavização por molas e a figura (d) é a malha da figura (c) depois do melhoramento feito pela ferramenta *stellar*.

Considerações Finais

Nesta tese apresentamos uma nova técnica para poligonalizar uma superfície obtida implicitamente através de uma nuvem de pontos. Apresentamos também uma técnica de melhoramento de malha volumétrica que combina as técnicas baseadas em molas virtuais e a ferramenta *Stellar* [23]. Quanto as contribuições podemos destacar dois enfoques: poligonalização de isosuperfícies e melhoramento de malhas volumétricas.

Poligonalização de isosuperfícies

Os resultados obtidos pela nova técnica de poligonalização de isosuperfícies apresentada no Capítulo 4 indicou uma melhora na condição da maioria dos elementos da malha superficial. Notamos, porém, que esta técnica gera um número grande de vértices com muitas arestas incidentes. Este fato contribui para gerar elementos com ângulos muito pequenos, o que não é interessante em aplicações de simulações de fluídos. Observamos porém, que este método gera um número menor de elementos e com melhor qualidade em comparação com o poligonalizador apresentado por Gois et al. [19]. Portanto, a nova técnica de poligonalização de isosuperfícies apresentada neste trabalho é uma alternativa a ser considerada para gerar uma malha superficial a partir de uma função implícita.

Não encontramos na literatura nenhum trabalho que busca movimentar os vértices que decompõe

o domínio da função implícita através da minimização de uma função energia. Os trabalhos presentes na literatura movimentam os vértices do domínio apenas de forma heurística [18, 13]. A utilização da minimização de uma função energia busca encontrar uma posição ideal dos vértices da decomposição do domínio para que a poligonalização gere elementos com boa qualidade. Estudos futuros podem ser feitos trabalhando sobre a função energia para obter melhores resultados.

Melhoramento de malhas volumétricas

Os resultados deste trabalho mostram que somente uma técnica para melhoramento de malha não é suficiente para obter malhas com qualidade aceitável, mas uma combinação de técnicas de melhoramento de malhas superficiais e volumétricas pode gerar uma malha final de boa qualidade.

A avaliação da qualidade é feita através da aferição do menor e maior ângulo diedral encontrado nos tetraedros que compõem a malha. Entretanto, o único método que oferece garantias sobre a qualidade das malhas obtém resultados com ângulos entre 10° e 165° [24].

Nos teste feitos em modelos sintéticos conseguimos bons resultados de adaptatividade e qualidade, entretanto, em testes feitos em modelos reais de artérias do corpo humano observamos que na maioria das vezes apenas conseguimos obter a adaptatividade nas malhas. A melhora da qualidade dos elementos não foi observada, como nos testes feito nos modelos sintéticos. A utilização da ferramenta *Stellar* desenvolvida por Klingner e Shewchuk [23] melhora a qualidade da malha, no entanto, quando combinadas com o melhoramento por molas virtuais conseguimos melhores resultados.

A ferramenta *Stellar* utilizada nas malhas iniciais aumentou significativamente a qualidade das malhas. A utilização da técnica de melhoramento por molas virtuais nas malhas iniciais, não garante a melhora na qualidade dos elementos das malhas. Entretanto, ao utilizar a ferramenta *Stellar* após a aplicação da técnica de melhoramento por molas virtuais, obtemos uma qualidade da malha superior a qualidade obtida ao aplicar apenas a ferramenta *Stellar* na malha inicial.

Portanto, os melhores resultados para modelos de artérias do corpo humano foram obtidos aplicando a técnica de melhoramento por molas virtuais seguido da aplicação da ferramenta *Stellar*.

A característica adaptativa da triangulação J_1^a utilizada para gerar a malha de tetraedros diminui o número de elementos necessários para gerar a malha volumétrica dos modelos. A ferramenta *Stellar*, por sua vez, diminui consideravelmente o número de elementos da malha volumétrica. A utilização

---

da ferramenta *Stellar* nas malhas volumétricas obtidas com a triangulação J_1^a diminuiu o número de elementos da malha, porém, a adaptatividade conseguida através da triangulação J_1^a não foi perdida.

Assim, para modelos de artérias do corpo humano, podemos concluir que a utilização do gerador de malhas volumétricas baseado na triangulação J_1^a juntamente com o melhoramento de malhas por molas e a ferramenta *Stellar* proporciona a geração de malhas volumétricas adaptativas com boa qualidade.

Os dados utilizados nesta tese foram fornecidos pelo projeto HeMoLab - Laboratório de Modelagem em Hemodinâmica. Um objetivo futuro é a integração destas técnicas a uma ferramenta para simulação computacional do Sistema Cardiovascular Humano [3] desenvolvido no projeto HemoLab do LNCC - Laboratório Nacional de Computação Científica.

Referências Bibliográficas

- [1] N. Amenta, M. Bern, and D. Eppstein. Optimal point placement for mesh smoothing. *Journal of Algorithms*, 30(2):302–322, 1999.
- [2] J. T. Batina. Unsteady euler airfoil solutions using unstructured dynamic meshes. *AIAA Journal*, 28(8):1381–1388, 1990.
- [3] P. J. Blanco, M. R. Pivello, S. A. Urquiza, N. A. D. S. e Silva, and R. A. Feijoo. Coupled models technology in multi-scale computational haemodynamics. *International Journal of Biomedical Engineering and Technology*, 5(2):132–156, 2011.
- [4] F. J. Blom. Considerations on the spring analogy. *International Journal for Numerical Methods in Fluids*, 32(6):647–668, 2000.
- [5] A. Bowyer. Computing Dirichlet tessellations. *The Computer Journal*, 24(2):162–166, 1981.
- [6] A. Castelo. *Aproximação adaptativa de variedades implícitas com aplicações na modelagem implícita e em equações algébrico-diferenciais*. PhD thesis, Departamento de Matemática, Pontifícia Universidade Católica, Rio de Janeiro, Brasil, 1992.
- [7] A. Castelo, L. G. Nonato, M. Siqueira, R. Minghim, and G. Tavares. The j_1^a triangulation: An adaptive triangulation in any dimension. *Computer & Graphics*, 30(5):737–753, 2006.
- [8] S.-W. Cheng and T. K. Dey. Quality meshing with weighted delaunay refinement. *SIAM Journal on Computing*, 33(1):69–93, 2004.
- [9] L. P. Chew. Guaranteed-quality triangular meshes. Technical Report TR-89-983, Department of Computer Science, Cornell University, Abril 1989.
- [10] I. L. L. da Cunha, A. D. Lacassa, V. Polizelli-Junior, J. P. Gois, L. G. Nonato, and A. C. Filho. Adaptive algebraic mesh generation from implicit functions. In *Iberian Latin American Congress On Computational Methods in Engineering*, 2008.

- [11] H. L. de Cougny and M. S. Shephard. Refinement, derefinement, and optimization of tetrahedral geometric triangulations in three dimensions. 1995.
- [12] B. N. Delaunay. Sur la Sphère Vide. *Izvestia Akademia Nauk SSSR, VII Seria, Otdelenie Matematicheskii i Estestvennyka Nauk*, 7:793–800, 1934.
- [13] C. A. Dietrich, C. E. Scheidegger, J. Schreiner, a. L. D. C. Jo L. P. Nedel, and C. T. Silva. Edge transformations for improving mesh quality of marching cubes. *IEEE Transactions on Visualization and Computer Graphics*, 15(1):150–159, 2009.
- [14] C. Farhat, C. Degand, B. Koobus, and M. Lesoinne. Torsional springs for two-dimensional dynamic unstructured fluid meshes. *Computer Methods in Applied Mechanics and Engineering*, 163(1-4):231–245, 1998.
- [15] D. A. Field. Laplacian smoothing and delaunay triangulations. *Communications in Applied Numerical Methods*, 4(6):709–712, 1988.
- [16] L. A. Freitag and C. Ollivier-Gooch. Tetrahedral mesh improvement using swapping and smoothing. *International Journal for Numerical Methods in Engineering*, 40:3979–4002, 1997.
- [17] J. P. Gois. *Mínimos-quadrados e aproximação de superfície de pontos: novas perspectivas e aplicações*. PhD thesis, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, Brasil, 2008.
- [18] J. P. Gois, V. Polizelli-Junior, T. Etienne, E. Tejada, A. Castelo, T. Ertl, and L. G. Nonato. Robust and adaptive surface reconstruction using partition of unity implicits. In *SIBGRAPI '07: Proceedings of the XX Brazilian Symposium on Computer Graphics and Image Processing*, pages 95–104. IEEE Computer Society, 2007.
- [19] J. P. Gois, V. Polizelli-Junior, T. Etienne, E. Tejada, A. Castelo, L. G. Nonato, and T. Ertl. Twofold adaptive partition of unity implicits. *The Visual Computer*, 24(12):1013–1023, Dezembro 2008.
- [20] E. Hartmann. A marching method for the triangulation of surfaces. *The Visual Computer*, 14(3):95–108, 1998.
- [21] B. Joe. Construction of three-dimensional Delaunay triangulations using local transformations. *Computer Aided Geometric Design*, 8(2):123–142, 1991.
- [22] G. Klintsek. Minimal triangulations of polygonal domains. In P. L. Hammer, editor, *Combinatorics 79*, volume 9 of *Annals of Discrete Mathematics*, pages 121–123. Elsevier, 1980.
- [23] B. M. Klingner and J. R. Shewchuk. Agressive tetrahedral mesh improvement. In *Proceedings of the 16th International Meshing Roundtable*, pages 3–23, october 2007.

- [24] F. Labelle and J. R. Shewchuk. Isosurface stuffing: fast tetrahedral meshes with good dihedral angles. *ACM Transactions on Graphics*, 26(3):57, 2007.
- [25] C. L. Lawson. Software for C^1 surface interpolation. *Mathematical Software III*, pages 161–194, 1977.
- [26] S. H. Lo. A new mesh generation scheme for arbitrary planar domains. *International Journal for Numerical Methods in Engineering*, 21:1403–1426, 1985.
- [27] S. H. Lo. Automatic mesh generation over intersecting surfaces. *International Journal for Numerical Methods in Engineering*, 38(6):943–954, 1995.
- [28] R. Löhner. Extensions and improvements of the advancing front grid generation technique. *Communications in Numerical Methods in Engineering*, 12(10):683–702, 1996.
- [29] R. Löhner and J. R. Cebral. Parallel advancing front grid generation. In *in Proceedings of the Eighth International Meshing Roundtable*, pages 67–74, 1999.
- [30] R. Löhner and E. Oñate. Advancing front techniques for filling space with arbitrary separated objects. *Finite Elements in Analysis and Design*, 46(1-2):140–151, Jan. 2010.
- [31] R. Löhner and P. Parikh. Generation of three-dimensional unstructured grids by the advancing-front method. *International Journal for Numerical Methods in Fluids*, 8(10):1135–1149, 1988.
- [32] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *SIGGRAPH Computer Graphics*, 21(4):163–169, 1987.
- [33] G. M. Nielson. On marching cubes. *IEEE Transactions on Visualization and Computer Graphics*, 9(3):283–297, 2003.
- [34] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H.-P. Seidel. Multi-level partition of unity implicits. *ACM Transactions on Graphics*, 22(3):463–470, 2003.
- [35] S. J. Owen. A survey of unstructured mesh generation technology. In *in 7th International Meshing Roundtable*, pages 239–267, 1998.
- [36] V. N. Parthasarathy and S. Kodiyalam. A constrained optimization approach to finite element mesh smoothing. *Finite Elements in Analysis and Design*, 9(4):309–320, 1991.
- [37] V. Polizelli-Junior. Métodos implícitos para a reconstrução de superfícies a partir de nuvens de pontos. Master’s thesis, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, Brasil, Abril 2008.

- [38] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: the Art of Scientific Computing*, chapter 10.2, pages 402–405. Cambridge University Press, 2 edition, 1992.
- [39] J. Ruppert. A delaunay refinement algorithm for quality 2-dimensional mesh generation. *Journal of Algorithms*, 18(3):548–585, 1995.
- [40] W. J. Schroeder and M. S. Shephard. A combined octree/delaunay method for fully automatic 3-D mesh generation. *International Journal for Numerical Methods in Engineering*, 29(1):37–55, 1990.
- [41] M. S. Shephard and M. K. Georges. Automatic three-dimensional mesh generation by the finite octree technique. *International Journal for Numerical Methods in Engineering*, 32(4):709–749, 1991.
- [42] J. R. Shewchuk. Adaptive precision floating-point arithmetic and fast robust geometric predicates. *Discrete and Computational Geometry*, 18(3):305–363, Outubro 1997.
- [43] J. R. Shewchuk. *Delaunay Refinement Mesh Generation*. PhD thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania, May 1997. Available as Technical Report CMU-CS-97-137.
- [44] J. R. Shewchuk. Two discrete optimization algorithms for the topological improvement of tetrahedral meshes. In *In Unpublished manuscript*, 2002.
- [45] J. R. Shewchuk. What is a good linear finite element? interpolation, conditioning, anisotropy, and quality measures. In *Eleventh International Meshing Roundtable*, pages 115–126, 2002.
- [46] A. A. Shostko, R. Löhner, and W. C. Sandberg. Surface triangulation over intersecting geometries. *International Journal for Numerical Methods in Engineering*, 44(9):1359–1376, 1999.
- [47] I. P. Soares. *Movimento de malhas e remalhamento de malhas superficiais*. PhD thesis, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, Brasil, Junho 2007.
- [48] D. F. Watson. Computing the Delaunay tessellation with application to Voronoi polytopes. *The Computer Journal*, 24:167–172, 1981.
- [49] N. P. Weatherill and O. Hassan. Efficient three-dimensional delaunay triangulation with automatic point creation and imposed boundary constraints. *International Journal for Numerical Methods in Engineering*, 37(12):2005–2039, 1994.
- [50] M. Yerry and M. Shephard. A modified quadtree approach to finite element mesh generation. *IEEE Computer Graphics and Applications*, 3(1):39–46, 1983.

- [51] M. A. Yerry and M. S. Shephard. Automatic three-dimensional mesh generation by the modified-octree technique. *International Journal for Numerical Methods in Engineering*, 20(11):1965–1990, 1984.
- [52] D. Zeng and C. R. Ethier. A semi-torsional spring analogy model for updating unstructured meshes in 3D moving domains. *Finite Elements in Analysis and Design*, 41(11-12):1118–1139, June 2005.
- [53] T. Zhou and K. Shimada. An angle-based approach to two-dimensional mesh smoothing. In *In Proceedings, 9th International Meshing Roundtable*, pages 373–384, 2000.