

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Superfícies de pontos dinâmicas

Anderson Luis Nakano

Orientador: Prof. Dr. Gustavo Carlos Buscaglia

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências - Ciências de Computação e Matemática Computacional.

USP – São Carlos
Fevereiro/2009

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

Superfícies de pontos dinâmicas

Anderson Luis Nakano

“Somente a forte pressão da refinada experiência científica logra libertar o pensamento humano das suas convicções habituais e arraigadas.”

- Moritz Schlick

Agradecimentos

Agradeço ao amigo e co-autor deste trabalho, João Paulo Gois, a dedicação imensa e competência científica. Ao professor Gustavo C. Buscaglia agradeço a genialidade de suas ideias e o rigor e postura profissionais, qualidades que contribuíram muito para o desenvolvimento deste trabalho.

Ao professor Luis Gustavo Nonato agradeço a concepção e originalidade deste projeto de pesquisa, e também o interesse e contribuição com parte significativa das ideias expostas nesta monografia.

Agradeço à FAPESP o apoio financeiro.

Agradeço aos cientistas cujos trabalhos foram citados nesta monografia. Parafraseando Isaac Newton, se este trabalho deu frutos, foi porque subi no ombro de gigantes.

Finalmente, agradeço à minha família e amigos queridos o carinho, compreensão e conforto em todos os momentos de minha vida. Em especial aos amigos Bráulio “Kazuma” Albuquerque, Mariana Yukari Noguti, Leandro “Spawn” Andrade, Karen Caires, Gustavo Menegaz Monteiro, Lísia Tabosa de Castro e Pavel Dodonov.

Resumo

O estudo do comportamento de fluidos é um antigo domínio das ciências da natureza. Ultimamente, fenômenos de engenharia que eram estudados empiricamente passaram a ser estudados com auxílio computacional. A Dinâmica de Fluidos Computacional (DFC) é a área da ciência da computação que estuda métodos computacionais para simulação de escoamento de fluidos, e muitas vezes é a forma mais prática, ou a única, de se observar fenômenos de interesse no escoamento.

Este projeto de Mestrado procurou investigar, no âmbito da simulação de um escoamento bifásico, métodos computacionais para representar a interface entre dois fluidos imiscíveis. A separação dos fluidos por meio de uma interface é necessária para assegurar que, propriedades como viscosidade e densidade, específicas de cada fluido, sejam utilizadas corretamente para o cálculo do movimento de seus respectivos fluidos. Desenvolvemos um método lagrangeano sem a utilização de malhas com o objetivo de suprir algumas restrições de trabalhos prévios. Para representar a interface entre os dois fluidos, este método utiliza uma técnica de reconstrução de superfícies baseada em aproximações de superfícies algébricas de alta ordem.

Os resultados numéricos reportados neste documento evidenciam o potencial da nossa abordagem.

Abstract

The study of the behaviour of fluids is an ancient field in natural sciences. Recently, engineering phenomena that were empirically studied started to be done with computational aid. The Computational Fluid Dynamics (CFD) is the area of science that studies computational methods for computer simulation of fluid flow, and often is the most practical way, or the only, to observe phenomena of interest in flow.

This Masters degree project sought to investigate, in the context of the simulation of biphasic flows, computational methods to represent the interface between two immiscible fluids. The separation of fluids by the means of an interface is required to ensure that, during the simulation, the physical properties of a fluid, like density and viscosity (specific of each fluid) are properly used in the calculus of the respective fluid motion. We developed a lagrangean method without the use of mesh with the goal of alleviating some of the previous works restrictions. To represent the interface between the two fluids, this method uses a surface reconstruction technique based on approximations of high order algebraic surfaces.

The numerical results reported herein show the potential of our approach.

Sumário

1	Introdução	1
1.1	Representação de Superfícies Dinâmicas	3
1.1.1	Métodos de Acompanhamento de Fronteira	3
1.1.2	Métodos de Captura de Fronteira	3
1.1.3	Métodos Híbridos	4
1.2	Contribuições e Organização	5
2	Representação por Superfícies de Pontos	7
2.1	Superfícies MLS	8
2.2	Superfícies MLS Simplificadas	13
2.2.1	Direção de Menor Covariância	15
2.2.2	Média de Normais	16
2.3	Superfícies MLS Puramente Implícitas	16

2.4	Superfícies MLS Algébricas	17
2.5	A Orientação do Vetor Normal	22
3	Metodologia	25
3.1	Estudos de Convergência das Superfícies AMLS	26
3.2	Características Afiadas	31
3.3	Superfícies AMLS com Robustez Aprimorada	32
3.3.1	Superfícies RAMLS	32
3.3.2	Estudo de Convergência das Superfícies RAMLS e de Robustez na Presença de <i>Sharp Features</i>	37
3.4	Acompanhamento de Fronteira com Superfícies RAMLS	39
3.4.1	O Algoritmo	39
3.4.2	Implementação da Interseção de Raios com a Superfície RAMLS	44
4	Resultados	49
4.1	Convergência do Método de Acompanhamento de Fronteira Com Su- perfícies RAMLS	50
4.2	Resultados numéricos	51
4.2.1	Disco de Zalesak e Esfera de Zalesak	52
4.2.2	Escoamento de um Vórtice	54
4.2.3	Deformação Tridimensional	63
4.3	Testes com Mudanças Topológicas	65
5	A Biblioteca RAMLS	79
5.1	Projeto da Biblioteca RAMLS	79
5.2	<i>Software</i> para Acompanhamento de Fronteira com Superfícies RAMLS	81

<i>SUMÁRIO</i>	ix
5.3 Exemplos de Simulações Realizadas com o <i>Software</i>	82
6 Conclusão	85
Referências Bibliográficas	87

Lista de Figuras

1.1	Simulação de um escoamento bifásico ar-água por meio de técnicas computacionais [18].	2
2.1	Exemplos de scanners 3D - esquerda: Scanner de contato. direita: Scanner à laser.	8
2.2	Projeção do ponto \mathbf{p} pelo método MLS. As cores dos pontos p_i representam a influência destes para o cálculo da projeção (cores claras indicam menor influência).	9
2.3	Curvas obtidas pelo método AMLS (ver Seção 2.4) para diversos valores de Δ : $\Delta = 3h$ (em preto), $\Delta = 5h$ (em verde), $\Delta = 7h$ (em cinza) e $\Delta = 10h$ (em rosa).	11
2.4	Aproximação por um plano de uma região de alta curvatura. O plano aproximado, com normal a , é perpendicular ao desejado.	12

- 2.5 Um ponto pertence à superfície extrema se ele é um mínimo da função e ao longo da reta l que passa pelo ponto \mathbf{x} e tem direção $\mathbf{n}(\mathbf{x})$ [9]. 14
- 2.6 O arco ab requer 10^4 pontos devido à singularidade à esquerda. Os pontos tracejados representam o eixo medial da superfície [16]. 18
- 2.7 Exemplos da função implícita F gerada pelo método AMLS (contornos azuis) e superfícies AMLS \mathcal{S} (linha preta) a partir de dois conjunto de pontos (quadrados pretos). (a) Amostra de um segmento de reta perturbado aleatoriamente (b) Amostra de uma característica afiada (*sharp feature*) perturbada aleatoriamente. 23
- 3.1 (a) \mathcal{P}_h correspondente a $h = 1,5$ para o caso da elipse, aliado à curva AMLS baseada em círculos \mathcal{S}_h (b) Gráficos de convergência de $d(\Sigma, \mathcal{S}_h)$ (denotado por erro geométrico), de $E_N(\Sigma, \mathcal{S}_h)$ (denotado por erro das normais) e de $E_C(\Sigma, \mathcal{S}_h)$ (denotado por erro da curvatura). 28
- 3.2 (a) \mathcal{P}_h correspondente a $h = 1,5$ para o caso da elipse, aliado à curva AMLS baseada em retas \mathcal{S}_h (b) Gráficos de convergência de $d(\Sigma, \mathcal{S}_h)$ (denotado por erro geométrico) e de $E_N(\Sigma, \mathcal{S}_h)$ (denotado por erro das normais). 29
- 3.3 Gráficos de convergência de $d(\Sigma, \mathcal{S}_h)$ (denotado por erro geométrico), de $E_N(\Sigma, \mathcal{S}_h)$ (denotado por erro das normais), e de $E_C(\Sigma, \mathcal{S}_h)$ (denotado por erro da curvatura) para o caso tridimensional do elipsoide. (a) Superfície AMLS baseada em esferas. (b) Superfície AMLS baseada em planos. . . . 31
- 3.4 (a) Superfície AMLS baseada em planos para o caso do elipsoide (b) Superfície AMLS baseada em esferas para o caso do elipsoide. O conjunto de pontos consiste de 265 pontos a uma distância média de $h = 0,75$ 31

- 3.5 Superfícies AMLS baseadas em esferas para o caso 3/4-esfera de um conjunto de pontos constituído de 1.030 pontos (a), de 2.031 pontos (b) e de 4.748 pontos (c). Superfícies espúrias detalhadas na parte superior da imagem ilustram a presença de artefatos na vizinhança de características afiadas. 33
- 3.6 (a) \mathcal{P}_h correspondente a $h = 1,5$ para o caso da elipse, aliado à curva RAMLS baseada em círculos \mathcal{S}_h (b) Gráficos de convergência de $d(\Sigma, \mathcal{S}_h)$ (denotado por erro geométrico), de $E_N(\Sigma, \mathcal{S}_h)$ (denotado por erro das normais) e de $E_C(\Sigma, \mathcal{S}_h)$ (denotado por erro da curvatura). 38
- 3.7 (a) \mathcal{P}_h correspondente a $h = 1,5$ para o caso do elipsoide, aliado à curva RAMLS baseada em círculos \mathcal{S}_h (b) Gráficos de convergência de $d(\Sigma, \mathcal{S}_h)$ (denotado por erro geométrico), de $E_N(\Sigma, \mathcal{S}_h)$ (denotado por erro das normais) e de $E_C(\Sigma, \mathcal{S}_h)$ (denotado por erro da curvatura). 39
- 3.8 Superfícies RAMLS baseadas em esferas para o caso 3/4-esfera de um conjunto de pontos constituído de 1.030 pontos (a), de 2.031 pontos (b) e de 4.748 pontos (c). 40
- 3.9 Exemplo do comportamento da curvatura exata (a) e da estimativa pelo método RAMLS por esferas (b) para o conjunto de pontos do elipsoide com $h = 1/128$ 41
- 3.10 Exemplos da função implícita F gerada pelo método RAMLS (contornos azuis) e superfícies RAMLS \mathcal{S} (linha preta) a partir de dois conjunto de pontos (quadrados pretos). (a) Amostra de um segmento de reta perturbado aleatoriamente (b) Amostra de uma característica afiada (*sharp feature*) perturbado aleatoriamente. Comparar com as superfícies AMLS da Fig. 2.7 42

3.11	Novo conjunto de pontos $\mathcal{Q}_h(t + \Delta t)$ gerado pela interseção do conjunto de raios (R_{1h}, R_{2h}) com a Superfície $\mathcal{S}_h(t + \Delta t)$	44
3.12	Interseções raio-esfera: três interseções produzem o mesmo zero na superfície. Neste caso, somente o primeiro ponto é adicionado ao novo conjunto de pontos.	47
4.1	Resultados do método PLS [18]: (a) distribuição de pontos próxima à interface, (b) solução obtida (em preto) após uma revolução.	52
4.2	Rotação do disco de Zalesak: resultados obtidos após uma revolução. (a) Original, (b) $h=1/64$, (c) $h = 1/128$, (d) $h = 1/256$, (e) $h = 1/512$	53
4.3	Rotação da esfera de Zalesak com $h = 1/256$ nos instantes $t = 0, 79, 157, 236, 314, 393, 471, 550$ e 628 (da esquerda à direita e de cima para baixo).	55
4.4	Solução exata (preto) e solução pelo método RAMLS (azul): $h = 1/256$ e $h = 1/512$ com $\Delta t = 0,01$ onde (a) e (b) correspondem ao instante $t = 1$, (c) e (d) correspondem ao instante $t = 3$, e (e) e (f) correspondem ao instante $t = 5$	56
4.5	Simulações do escoamento em torno de um vórtice (grade computacional com espaçamento $1/128$ e $t = 3$): (a) método LPSL [23], (b) em vermelho temos a solução pelo método LS, em verde, a solução exata e em azul a solução pelo método PLS [18].	57
4.6	Comparação do método proposto (curva azul) com o método de acompanhamento de fronteira desenvolvido por Du et al [17] (curva vermelha) para o caso do escoamento de um vórtice no tempo $t = 3$. Para ambos métodos foi fixado $h = 1/256$ e os pontos são re-gerados a cada 0,01 unidades de tempo. A curva preta ilustra a solução exata (obtida por transporte lagrangeano de 100.000 pontos) para efeitos comparativos.	58

4.7 Círculo exato (em preto), e obtidos numericamente pelo método proposto com $h = 1/64$ (em verde), $h = 1/128$ (em laranja), $h = 1/256$ (em azul), $h = 1/512$ (em pink). 60

4.8 (a) e (b): resultados do método LPLS [23] para grades computacionais com espaçamento de $1/128$ e $1/1000$, respectivamente; (c): soluções por LS [36] para grades computacionais com espaçamento de $1/64$ (desapareceu), $1/128$ (vermelho) e $1/256$ (azul), e a solução exata (preto); (d): resultados para o método PLS [18], em que o número inicial de partículas corresponde a 3376, 15040 e 59072 para as grades computacionais com espaçamentos $1/64$ (verde) , $1/128$ (vermelho) e $1/256$ (azul), e a solução exata (preto). 60

4.9 Resultados numericamente obtidos em $t = T = 8$ para o escoamento de um vórtice modulado no tempo. O parâmetro h é fixado em $h = 1/256$ e os pontos são re-gerados a cada 0,01 unidades de tempo. A curva azul é obtida pelo método RAMLS, enquanto que a curva vermelha é obtida pelo método de Du et al. [17]. A curva preta ilustra a solução exata (obtida por transporta lagrangeano de 100.000 pontos) para efeitos comparativos. 61

4.10 Evolução do número de pontos em $Q_h(t)$ ao decorrer da simulação modulada no tempo do escoamento de um vórtice. 62

4.11 Evolução do número de partículas geradas pelo método LPSL [23] no decorrer da simulação modulada no tempo do escoamento de um vórtice para uma grade computacional com espaçamento $1/128$ 62

4.12	Deformação tridimensional ($h = 1/512$) obtida numericamente pelo método proposto em diferentes instantes de tempo (espaçados igualmente, com o campo de velocidades sendo invertido em $t = 2$). O número de pontos que representam a superfície são (da esquerda para direita e de cima para baixo), respectivamente, 65.000, 69.314, 109.728, 187.571, 277.021, 193.238, 114.898, 72.774, 68.031.	64
4.13	A superfície do canto inferior direito da Fig. 4.12 vista de um outro ângulo. Uma pequena silhueta pode ser observada na superfície.	65
4.14	Deformação tridimensional ($h = 1/512$) modulada no tempo. Da esquerda para a direita e de cima para baixo: obtidas nos instantes $t = 0, 10\Delta t, 20\Delta t, 30\Delta t, 40\Delta t, 50\Delta t, 60\Delta t, 70\Delta t, 90\Delta t, 110\Delta t, 130\Delta t, 150\Delta t$, onde $\Delta t = 3/150$. O número de pontos que representam a superfície são: 65.000, 73.790, 88.263, 136.299, 183.720, 228.456, 263.896, 280.663, 263.853, 183.932, 88.545 e 69.295	66
4.15	A superfície do canto inferior direito da Fig. 4.14 vista de um outro ângulo. Uma pequena silhueta pode ser observada na superfície.	66
4.16	Deformação tridimensional de uma esfera ($h = 1/256$). Resultados do método RAMLS nos instantes $t = 0, t = 1, 5$ e $t = 3$	67
4.17	Deformação tridimensional de uma esfera ($h = 1/256$). Resultados do método proposto por Du et al. [17] nos instantes $t = 0, t = 1, 5$ e $t = 3$	67
4.18	Solução do caso de deformação tridimensional pelos métodos (grade computacional com espaçamento de 1/100): (a) LS [36](b) PLS [18] e (c) LPLS [23].	68
4.19	Resultados do teste tridimensional do anel aberto nos instantes: (a) 0, (b) 0,2, (c) 0,4, (d) 0,6, (e) 0,8, (f) 1,0, (g) 1,2 e (h) 1,4. A mudança de topologia ocorre no instante $t = 0,6$	69

- 4.20 Resultado do teste bidimensional do “anel aberto”: artefatos numéricos podem ser observados na solução obtida ($t = 1, 0$). 69
- 4.21 Ilustração dos conjuntos de nível utilizados para executar as operações de coalescência e ruptura da superfície: uma ruptura da superfície no conjunto de nível $-\epsilon$ (em vermelho) (em comparação com o conjunto de nível zero (em azul)) pode ser observada, enquanto que no nível ϵ (em laranja) uma coalescência pode ser observada. 70
- 4.22 A função implícita gerada pelo método RAMLS subestima a função distância quando duas “folhas” da superfície estão próximas. Em preto, temos o seu conjunto de nível zero; em rosa, temos o conjunto de nível $\vartheta/2$, onde ϑ é a distância entre as duas “folhas” da superfície; em azul temos diversas outras curvas de nível. 73
- 4.23 Ilustração da operação de coalescência: primeiramente, os pontos em laranja são obtidos pela interseção entre o grid R_h e o conjunto de nível ϵ . Em seguida os pontos em laranja são transportados na direção do gradiente da função, definindo os pontos em azul. Finalmente, os pontos em azul são utilizados para executar corretamente a mudança topológica, definindo uma superfície com diferente topologia (curva azul-marinho). 74
- 4.24 Resultado do teste bidimensional do “anel aberto” utilizando a heurística proposta. 74
- 4.25 Simulação na qual coalescência e ruptura ocorrem no mesmo passo de tempo. 75
- 4.26 O parâmetro ϵ permite controlar a distância máxima entre duas superfícies opostas. (a) $\epsilon = 0, 5h$: sem mudanças topológicas; (b) $\epsilon = 1, 5h$, coalescência entre os braços dos bonecos. 75

4.27	Simulação em que ocorre uma ruptura da superfície: a imagem à esquerda mostra a configuração inicial. Após quatro passos de tempo, a imagem do meio ilustra a ausência de mudanças topológicas (considerando $\epsilon = 0,4h$), enquanto que na imagem à direita pode-se observar uma ruptura da superfície (considerando $\epsilon = 1,0h$	76
4.28	Coalescência entre dois modelos tridimensionais do “Hommer Simpson”	77
5.1	Diagrama de classes da biblioteca RAMLS.	80
5.2	Diagrama de classes do <i>Software</i> para Acompanhamento da Frente com Superfícies RAMLS. Classes com bordo vermelho pertencem à biblioteca RAMLS, enquanto que as classes com bordo em azul são específicas do <i>software</i> desenvolvido.	82
5.3	Deformação do Stanford Bunny ($h = 1/512$).	83
5.4	Deformação do Hommer Simpson.	84

Lista de Tabelas

4.1	Erro geométrico obtido pelo método proposto na rotação rígida de um elipsoide ao redor do eixo x_3 (uma revolução).	51
4.2	Disco de Zalesak: método RAMLS	54
4.3	Escoamento de um vórtice modulado no tempo para $T = 8$. Resultados obtidos pelo método RAMLS.	59

CAPÍTULO 1

Introdução

O estudo do comportamento de fluidos é um antigo domínio das ciências da Natureza. Os princípios empíricos para o estudo de fluidos começaram a ser desenvolvidos desde a Grécia antiga, por Aristóteles. As deduções matemáticas da Dinâmica dos Fluidos pela mecânica newtoniana, entretanto, são relativamente recentes, dadas pelas equações de Euler e de Navier-Stokes. Esta última descreve o movimento dos fluídos por meio de leis físicas de conservação de massa, energia e momento.

Soluções analíticas para as equações de Navier-Stokes foram encontradas apenas para alguns casos específicos. A dificuldade de se encontrar soluções mais gerais se deve à não-linearidade das equações diferenciais parciais (EDPs) que fazem parte das equações de Navier-Stokes. Por essa razão, procedimentos empíricos vêm sendo utilizados para estudar o escoamento de fluidos, alguns fazendo uso, por exemplo, de túneis de vento e tanques d'água. Devido ao custo de tais procedimentos, as medições das propriedades do escoamento são feitas em apenas alguns pontos de interesse.

Ultimamente, fenômenos de engenharia que eram estudados empiricamente passaram a ser estudados com auxílio computacional. A Dinâmica de Fluidos Computacional (DFC) é a área da Ciência da Computação que estuda métodos computacionais para simulação de escoamento de fluidos, e muitas vezes é a forma mais prática, ou a única, de se observar fenômenos de interesse no escoamento.

Dentro deste contexto, este projeto estuda métodos computacionais no âmbito da simulação de escoamentos bifásicos, que consiste na simulação do escoamento simultâneo de dois fluidos (A e B) com propriedades físicas diferentes, contidos em um domínio finito Ω , com bordo $\partial\Omega$. Para simular escoamentos bifásicos, deve-se decidir, para todo o domínio Ω , qual região cada fluido ocupa, afim de assegurar que, propriedades como viscosidade e densidade, específicas de cada fluido, sejam utilizadas corretamente para o cálculo do movimento de seus respectivos fluidos. Isto é feito por meio de uma representação da interface que separa os dois fluidos. Esta interface recebe o nome de superfície dinâmica. A título de exemplo, a Fig. 1.1 ilustra a simulação computacional de um escoamento bifásico ar-água, em que há a necessidade de delimitar o espaço ocupado por ambos fluidos.



Figura 1.1: Simulação de um escoamento bifásico ar-água por meio de técnicas computacionais [18].

1.1 Representação de Superfícies Dinâmicas

Os métodos de representação de superfícies dinâmicas podem ser classificados basicamente em três categorias: captura de fronteira (*front-capturing*), acompanhamento de fronteira (*front-tracking*) e métodos híbridos. Nas próximas Seções detalharemos tais métodos.

1.1.1 Métodos de Acompanhamento de Fronteira

Os métodos de acompanhamento de fronteira (puramente lagrangeanos) utilizam partículas marcadoras na superfície, as quais são movimentadas de acordo com um campo de velocidade. Estes métodos possibilitam uma representação bastante precisa da superfície e são utilizados em diversos trabalhos [28, 44, 43, 19, 39, 17]. Por outro lado, devido à representação explícita da superfície (geralmente feita por meio de um conjunto de pontos conectados), mudanças topológicas ocorridas durante a movimentação da fronteira são muito difíceis de serem tratadas. Uma outra desvantagem desta abordagem é a dificuldade de manter a densidade de pontos consistente com o nível de discretização desejada: há o acúmulo de pontos em algumas áreas da superfície em detrimento de outras áreas, sendo necessárias estratégias de criação e remoção de pontos no decorrer da simulação.

1.1.2 Métodos de Captura de Fronteira

Também chamados de métodos eulerianos, os métodos de captura de fronteira, tais como *Volume-of-Fluid* (VOF) [27, 35] e *Level-Set* (LS) [36, 32, 37] (ou combinação de ambos [40, 41]), em geral não fazem uso de uma representação explícita da superfície, a qual é representada implicitamente pelos conjuntos de nível uma função [33]. A principal vantagem desta abordagem é que mudanças abruptas na topologia podem ser facilmente

manipuladas durante a movimentação da superfície, fazendo com que estes métodos sejam preferidos em problemas tais como coalescência de bolhas, quebra de ondas, etc [46, 13, 14]. Tais métodos são baseados no transporte de um campo escalar ϕ definido em todo o domínio da simulação. No método VOF, este campo escalar representa o conteúdo que um dos fluidos ocupa em cada célula de um grid euleriano. Já no método LS, ϕ define implicitamente a interface como seu conjunto de nível zero. O principal problema dos métodos de *level-set* é que erros numéricos devido ao transporte da função ϕ causam problemas na robustez e estabilidade do método, levando a perdas ou ganhos espúrios de matéria, o que pode ocasionar erros qualitativos na simulação.

1.1.3 Métodos Híbridos

Os métodos híbridos procuram combinar a precisão dos métodos lagrangeanos com a flexibilidade e robustez dos métodos eulerianos. Du *et al.* [17] combinaram um método puramente lagrangeano com um método de reconstrução sobre um grid euleriano que é aplicado apenas nos locais em que dificuldades no tratamento de mudanças topológicas aparecem. Como as partículas marcadoras da interface mantêm uma estrutura de conectividade (malha lagrangeana), a reconstrução de regiões em que uma parte da malha cruza a outra é extremamente sofisticada, especialmente em três dimensões.

Torres e Brackbill [42] introduziram um método híbrido denominado *Point-Set Method*, no qual o acompanhamento da fronteira é realizado sem conectividade entre os pontos. Este método transporta partículas marcadoras de acordo com um campo de velocidade, e então constrói uma função ϕ resolvendo a equação de Laplace no domínio Ω . Definindo $\phi = 1$ nas células que contém partículas marcadoras e $\phi = 0$ em $\partial\Omega$, a função resultante ϕ identifica a região ocupada pelo fluido A . Um procedimento para suavizar a aproximação baseado em *B-splines* é aplicado, seguido de uma técnica de correção, para assegurar que o conjunto de nível zero de ϕ passe pelas partículas marcadoras. A partir disso a interface é re-gerada como sendo a projeção do centro das células

de um *grid* euleriano no conjunto de nível zero de ϕ .

Enright *et al.* [18], por outro lado, partem de um método de LS e o tornam mais preciso incorporando informações sobre a posição de partículas lagrangeanas. Este método, conhecido como *Particle Level Set* (PLS), tem obtido êxito em diversas aplicações [31, 32], já que mudanças topológicas são tratadas facilmente pela parte euleriana do método. Outro método que utiliza partículas lagrangeanas para correção da interface é o método *Lagrangian Particle Level Set* (LPLS), que utiliza uma técnica baseada em *Smooth Particle Hydrodynamics* (SPH).

1.2 Contribuições e Organização

Este projeto introduz o uso de superfícies de pontos (*Point Set Surfaces*) [6], em particular, de superfícies MLS (*moving-least-squares*) [29], como novo esquema de representação para superfícies dinâmicas. Superfícies de pontos, de fato, podem ser vistas como uma representação híbrida, pois, como nos métodos de captura de fronteira, a superfície é representada implicitamente, embora pontos próximos a tal superfície devam ser explicitamente representados. Tais superfícies, originalmente concebidas a partir de operadores de projeção [29, 6], possuem características que as colocam naturalmente em uma posição diferenciada em termos de esquema de representação de fronteiras. De fato, a superfície propriamente dita é representada implicitamente, como nos métodos de captura, embora sejam necessários pontos marcadores para guiar tal representação implícita [5, 10], uma característica presente nos métodos de acompanhamento de fronteira. A metodologia proposta evita o custo de resolver os sistemas lineares inerentes ao método de Torres e Brackbill [42], e pode ser paralelizada de forma natural. Mais vantagens da abordagem proposta sobre os métodos híbridos citados são encontradas no Capítulo 4.

Este trabalho está organizado da seguinte forma: no Capítulo 2 será apresentado um

levantamento bibliográfico sobre representação de superfícies por meio de um conjunto não-estruturado de pontos. O Capítulo 3 apresenta a metodologia utilizada para a realização dos testes numéricos, os quais são apresentados no Capítulo 4. O Capítulo 5 descreve a biblioteca desenvolvida para que pesquisadores da área tenham um acesso prático ao trabalho desenvolvido. O Capítulo 6 apresenta as conclusões deste projeto, direcionando trabalhos futuros na área.

CAPÍTULO 2

Representação por Superfícies de Pontos

Em computadores digitais, a forma de um objeto físico geralmente é descrita por meio de elementos de geometria. Objetos assim representados são denominados modelos geométricos e são frequentemente usados em diversas áreas do conhecimento humano, tais como Computação Gráfica, Desenho Industrial, Arquitetura, Simulação de Fenômenos Físicos e Entretenimento.

A visualização de modelos geométricos e a geração de imagens a partir destes é fonte de inúmeras pesquisas em Computação Gráfica. Por exemplo, a reconstrução da superfície de um modelo a partir de pontos obtidos por meio de um *Scanner* 3D (Figura 2.1) tem se consolidado como uma importante área de pesquisa nas últimas duas décadas [24, 25].

Há inúmeros métodos para a reconstrução de superfícies de pontos na literatura. A maioria dos métodos busca construir estruturas conexas entre os pontos, transformando-



Figura 2.1: Exemplos de scanners 3D - esquerda: Scanner de contato. direita: Scanner à laser.

os em modelos poligonais. Há várias técnicas de triangulação que permitem transformar um conjunto de pontos em uma malha de triângulos [8, 12, 21]. Estes métodos, não raro, geram superfícies irregulares, contendo túneis e buracos. Além disso, mudanças topológicas em superfícies dinâmicas dificultam a utilização de estruturas conexas para a reconstrução de tais superfícies.

2.1 Superfícies MLS

Uma outra abordagem é a reconstrução da superfície por meio da representação implícita dada por uma nuvem de pontos sem informação topológica. Recentemente foram propostas técnicas locais de representação de superfícies por nuvem de pontos. O nome *Point Set Surfaces* (PSS) refere-se a tais superfícies e foi introduzido por Alexa et al. [7]. Estas técnicas foram amplamente difundidas na literatura nos últimos anos devido à sua alta flexibilidade (utilização em diversos segmentos da ciência sem a necessidade de grandes mudanças da abordagem) e facilidade de implementação.

As técnicas de Levin [30] e de Alexa et al. [7] representam a superfície pelo método de *moving-least-squares* (MLS). Este método foi utilizado por Levin [29] para construir

operadores de projeção de pontos na superfície. Estes operadores envolvem duas etapas: (i) uma otimização não-linear para encontrar um plano de referência local; (ii) uma aproximação polinomial local. A seguir descrevemos com detalhes estas etapas.

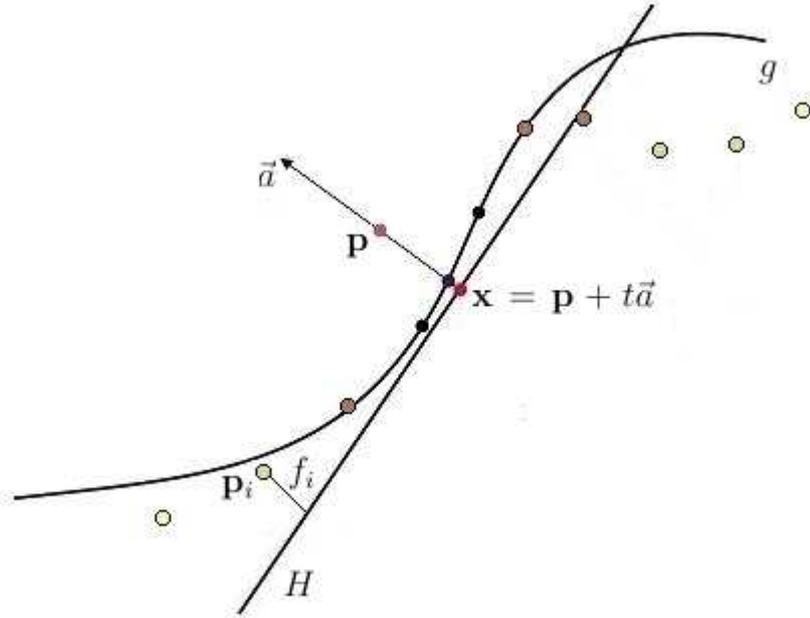


Figura 2.2: Projeção do ponto \mathbf{p} pelo método MLS. As cores dos pontos p_i representam a influência destes para o cálculo da projeção (cores claras indicam menor influência).

Seja $\mathcal{P} = \{p_i \in \mathbb{R}^n, i \in \{1, \dots, \eta\}\}$ um conjunto de pontos quase regularmente espaçados de uma distância h amostrados sobre uma $(n - 1)$ -variedade Σ , que pode ser uma curva (1-variedade) se $n = 2$ ou uma superfície (2-variedade) se $n = 3$.

Seja $\mathbf{p} \in \mathbb{R}^n$ um ponto que queremos projetar na superfície. Primeiro, minimiza-se localmente uma função energia e_{MLS} do plano com normal \vec{a} que passa pelo ponto $\mathbf{x} = \mathbf{p} + t\vec{a}$, onde t é a distância de \mathbf{p} ao plano. Um raio de influência Δ é definido para escolher quais pontos de \mathcal{P} influenciarão no cálculo do plano de referência local. Suponhamos por simplicidade que tais pontos sejam $\mathcal{P}' = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_m\}$, ou seja, $\|\mathbf{x} - \mathbf{p}_i\| < \Delta, i = 1, \dots, m$.

Para cada ponto $\mathbf{p}_i \in \mathcal{P}'$, definimos $w_i(\mathbf{x})$ como sendo

$$w_i(\mathbf{x}) = \psi\left(\frac{\|\mathbf{x} - \mathbf{p}_i\|}{\Delta}\right), \quad (2.1)$$

onde $\psi : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ é uma função contínua e não-negativa (denominada na literatura como função núcleo (*kernel*) ou função peso) que satisfaz:

$$\psi(0) = 1, \quad \psi'(\xi) \leq 0, \quad \psi(\xi \rightarrow \infty) = 0. \quad (2.2)$$

Finalmente, a função energia e_{MLS} é uma soma ponderada das distâncias dos pontos \mathbf{p}_i ao plano:

$$e_{MLS}(\vec{a}, t) = \sum_{\mathbf{p}_i \in \mathcal{P}'} (\langle \vec{a}, \mathbf{p}_i \rangle - \langle \vec{a}, \mathbf{p} + t\vec{a} \rangle)^2 w_i(\mathbf{p} + t\vec{a}), \quad (2.3)$$

onde \langle, \rangle denota o produto interno canônico em R^n .

A função ψ é utilizada para ponderar a influência dos pontos p_i conforme suas distâncias ao ponto \mathbf{p} . Adotamos neste projeto:

$$\psi(\xi) = \begin{cases} (1 - \xi^2)^4 & \text{se } \xi < 1 \\ 0 & \text{caso contrário} \end{cases}. \quad (2.4)$$

Alterando-se o parâmetro Δ pode-se controlar a suavidade da aproximação: para Δ pequeno, poucos pontos são utilizados, tornando a aproximação mais local, enquanto que para Δ grande, a aproximação torna-se mais global, suavizando mudanças bruscas na superfície. O parâmetro Δ pode variar de acordo com características locais da superfície. Por simplicidade, utilizamos neste projeto $\Delta = 3h$, constante e escolhido após diversas simulações numéricas com diferentes valores de Δ . A Figura 2.3 mostra diversas curvas obtidas pelo método AMLS (ver Seção 2.4). É evidente que a curva se torna cada vez mais suave de acordo com o aumento do parâmetro Δ , por isto Δ deve ser escolhido o

menor possível. Todavia, a escolha de um Δ muito pequeno pode levar a buracos na superfície, pois a redução do parâmetro Δ diminui o domínio da função de projeção de pontos (como veremos adiante).

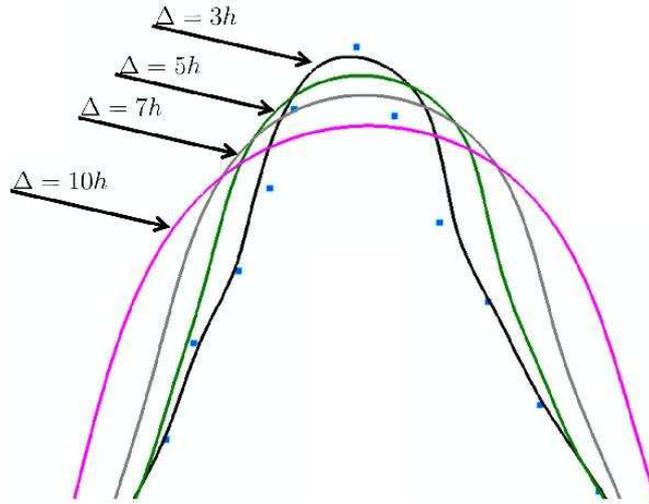


Figura 2.3: Curvas obtidas pelo método AMLS (ver Seção 2.4) para diversos valores de Δ : $\Delta = 3h$ (em preto), $\Delta = 5h$ (em verde), $\Delta = 7h$ (em cinza) e $\Delta = 10h$ (em rosa).

Como o plano não depende da orientação do vetor \vec{a} , podemos reescrever a Equação 2.3 como:

$$e_{MLS}(\mathbf{x}, a) = \sum_{\mathbf{p}_i \in \mathcal{P}'} (a^\top (\mathbf{x} - \mathbf{p}_i))^2 w_i(\mathbf{x}). \quad (2.5)$$

Quando a função energia e_{MLS} possui mais de um mínimo local, escolhe-se aquele para o qual t é mais próximo de zero [7]. O plano definido pelo mínimo local da função energia é uma aproximação local da superfície Σ . Um dos problemas inerentes a esta abordagem é que nem sempre este plano é uma boa aproximação para a superfície, ou seja, nem sempre pode-se aproximar adequadamente a superfície por um polinômio a partir deste plano de referência. Em regiões de alta curvatura, é necessário um número muito grande de pontos para que o plano aproxime bem a superfície. Este problema pode

ser visualizado na Figura 2.4, onde o plano aproximado é perpendicular ao desejado.

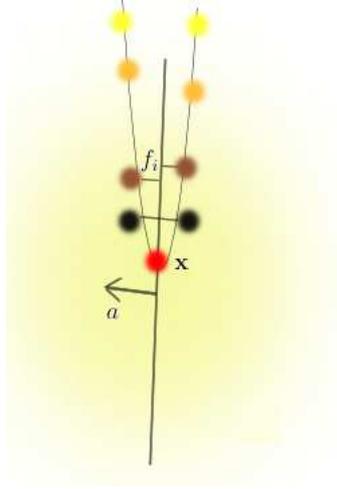


Figura 2.4: Aproximação por um plano de uma região de alta curvatura. O plano aproximado, com normal a , é perpendicular ao desejado.

No segundo passo, o plano H encontrado na etapa de minimização é utilizado como sistema de coordenadas locais para aproximar a superfície Σ em uma vizinhança de \mathbf{p} por um polinômio local g . Seja x_i a projeção de \mathbf{p}_i em H e f_i a altura de \mathbf{p}_i com relação a H , ou seja, $f_i = a^\top(\mathbf{x} - \mathbf{p}_i)$. Os coeficientes de g podem ser encontrados pelo método dos mínimos quadrados ponderados, por meio da minimização da função:

$$\sum_{i=1}^m (g(y_i) - f_i)^2 w_i(\mathbf{x}), \quad (2.6)$$

onde y_i é a representação de x_i no sistema de coordenadas locais referente a H . Assim, a projeção de \mathbf{p} em H é definida como sendo o valor do polinômio local g na origem, i.e. $\mathbf{x} + g(0)a$.

A partir deste procedimento podemos formular a seguinte definição:

Definição 2.1.1 *Seja $\mathcal{P}(\mathbf{x})$ a projeção do ponto \mathbf{x} utilizando o método descrito acima. A superfície MLS do conjunto de pontos \mathcal{P} é definida como sendo o conjunto de pontos*

estacionários de \mathcal{P} , ou seja, $\{\mathbf{p} \in R^n : \mathcal{P}(\mathbf{p}) = \mathbf{p}\}$.

Levin provou que a superfície MLS definida por 2.1.1 é uma $(n - 1)$ -variedade e infinitamente suave se a função núcleo $\psi \in C^\infty$ [29].

2.2 Superfícies MLS Simplificadas

Amenta e Kil [9] demonstraram que, se considerarmos apenas o primeiro passo do método de projeção de Levin, a superfície MLS pode ser definida como uma superfície extrema, generalizando e definindo explicitamente tais superfícies.

Definição 2.2.1 *A superfície extrema das funções $\mathbf{n} : \mathbb{R}^3 \rightarrow \mathbb{P}^2$ (espaço das direções na 2-esfera) e $\mathbf{e} : \mathbb{R}^3 \times \mathbb{P}^2 \rightarrow \mathbb{R}$ é definida como:*

$$\mathcal{S}_E = \{\mathbf{x} | \mathbf{x} \in \text{arglocalmin}_{\mathbf{y} \in l_{\mathbf{x}, \mathbf{n}(\mathbf{x})}} \mathbf{e}(\mathbf{y}, \mathbf{n}(\mathbf{x}))\}, \quad (2.7)$$

onde $l_{\mathbf{x}, \mathbf{n}(\mathbf{x})}$ é a reta que passa por \mathbf{x} na direção $\mathbf{n}(\mathbf{x})$. A figura 2.5 ilustra este conceito: o ponto \mathbf{x} pertence à superfície MLS, pois ele é um mínimo da função energia ao longo da reta $l_{\mathbf{x}, \mathbf{n}(\mathbf{x})}$.

Assim, uma proposição equivalente à Definição 2.1.1 pode ser feita explicitamente:

Proposição 2.2.1 *Seja \mathbf{e} a função energia dada pela Eq. 2.5 e $\mathbf{n} = \text{argmin}_a \mathbf{e}(\mathbf{x}, a)$. A superfície extrema definida por estas duas funções é uma superfície MLS.*

A partir desta generalização, podemos mudar as funções \mathbf{e} e \mathbf{n} para produzir outras superfícies extremas. Para evitar a otimização não-linear (da primeira etapa do método de projeção descrito acima), Adamson e Alexa [3] utilizaram uma função energia simplificada, dada por $\mathbf{e}(\mathbf{x}, a) = |a^T \cdot (\mathbf{x} - c(\mathbf{x}))|$, onde

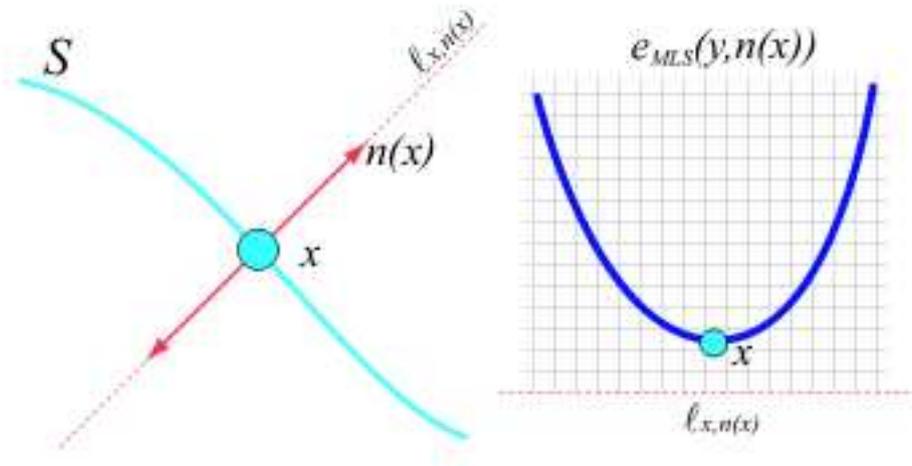


Figura 2.5: Um ponto pertence à superfície extrema se ele é um mínimo da função e ao longo da reta l que passa pelo ponto \mathbf{x} e tem direção $\mathbf{n}(\mathbf{x})$ [9].

$$c(\mathbf{x}) = \frac{\sum_{i=1}^m w_i(\mathbf{x}) \mathbf{p}_i}{\sum_{i=1}^m w_i(\mathbf{x})} \quad (2.8)$$

é o centro de massa do conjunto \mathcal{P}' ponderado por \mathbf{x} . Assim, quando $\mathbf{x} - c(\mathbf{x})$ é ortogonal a $\mathbf{n}(\mathbf{x})$, temos um mínimo local, ou seja:

$$\mathbf{x} - c(\mathbf{x}) \perp \mathbf{n}(\mathbf{x}) \Leftrightarrow \mathbf{n}(\mathbf{x})^T (\mathbf{x} - c(\mathbf{x})) = 0. \quad (2.9)$$

A equação 2.9 permite formular a seguinte função implícita:

$$F(\mathbf{x}) = \mathbf{n}(\mathbf{x}) \cdot (\mathbf{x} - c(\mathbf{x})) \quad (2.10)$$

Ainda com o objetivo de evitar a otimização não-linear, Zwicker et al. [47] utilizaram a função energia $e(y, \mathbf{n}(\mathbf{x})) = \frac{1}{2} \sum_{\mathbf{p}_i \in \mathcal{P}'} [(y - \mathbf{p}_i)^T \mathbf{n}(\mathbf{x})]^2 w_i(\mathbf{x})$, que nos permite formular a

seguinte função implícita:

$$F(\mathbf{x}) = \sum_{\mathbf{p}_i \in \mathcal{P}'} [(\mathbf{x} - \mathbf{p}_i)^T \mathbf{n}(\mathbf{x})] w_i(\mathbf{x}). \quad (2.11)$$

As funções implícitas acima permitem formular variantes simplificadas da superfície MLS, as quais denominaremos *Simplified MLS* (SMLS).

Definição 2.2.2 *Superfícies SMLS são obtidas considerando o conjunto zero da Função implícita 2.10 ou da função implícita 2.11.*

Observação: Não há garantias teóricas de que as superfícies SMLS sejam $(n - 1)$ -variedades. Mesmo assim as chamaremos de superfícies SMLS. A mesma observação é válida para as superfícies AMLS e RAMLS mencionadas posteriormente neste trabalho.

Adamson e Alexa [5] propõem duas técnicas para aproximar a função $\mathbf{n}(\mathbf{x})$, as quais descreveremos a seguir.

2.2.1 Direção de Menor Covariância

Para calcular o plano que melhor aproxima a tangente à superfície cuja normal $\mathbf{n}(\mathbf{x})$ passa por \mathbf{x} , podemos minimizar a soma dos quadrados das distâncias ponderadas dos pontos $\mathbf{p}_i \in \mathcal{P}'$ ao plano:

$$\min_{\|\mathbf{n}(\mathbf{x})\|=1} \sum_{\mathbf{p}_i \in \mathcal{P}'} w_i(\mathbf{x}) (\mathbf{n}(\mathbf{x})^\top (\mathbf{x} - \mathbf{p}_i))^2. \quad (2.12)$$

A solução deste problema se resume a encontrar o autovetor referente ao menor autovalor da matriz de covariância

$$\mathbf{C}(\mathbf{x}) = \sum_{\mathbf{p}_i \in \mathcal{P}'} w_i(\mathbf{x}) (\mathbf{x} - \mathbf{p}_i) (\mathbf{x} - \mathbf{p}_i)^\top. \quad (2.13)$$

Caso a função \mathbf{n} seja aproximada por este método, as funções implícitas 2.10 e 2.11 passam a ter um domínio específico U , dado por:

$$U = \{\mathbf{x} \in \mathbb{R}^n \mid \text{o número de pontos } \mathbf{p} \in \mathcal{P} \text{ tal que } \|\mathbf{x} - \mathbf{p}\| < \Delta \text{ seja } \geq 3.\} \quad (2.14)$$

Esta restrição do domínio das funções implícitas é devido a necessidade de pelo menos 3 pontos para que a matriz C seja não-singular. Ainda há a restrição de que estes 3 pontos não podem ser colineares.

Uma dificuldade inerente deste método é a escolha da orientação do autovetor encontrado. Este problema será novamente mencionado na Seção 2.5.

2.2.2 Média de Normais

Se assumirmos que para cada ponto $\mathbf{p}_i \in \mathcal{P}'$ temos um vetor \mathbf{n}_i associado, podemos utilizar uma estimativa para a normal no ponto \mathbf{x} baseada na média ponderada das normais dos vizinhos de \mathbf{x} , utilizando a seguinte equação:

$$\mathbf{n}(\mathbf{x}) = \sum_{i=1}^m \frac{w_i(\mathbf{x})\mathbf{n}_i}{w_i(\mathbf{x})}. \quad (2.15)$$

2.3 Superfícies MLS Puramente Implícitas

Shen et al. [38] propuseram um método puramente implícito para representar a superfície. Embora tenha sido proposto inicialmente para reconstruir superfícies a partir de um conjunto de polígonos, o mesmo método foi utilizado por Kolluri [26] no contexto de PSS. Utilizaremos o nome Implicit MLS Surfaces (IMLS) para fazer referências a este método.

Definição 2.3.1 A superfície IMLS é o conjunto zero da função implícita

$$F(\mathbf{x}) = \frac{\sum_{\mathbf{p}_i \in \mathcal{P}'} [(\mathbf{x} - \mathbf{p}_i)^T \mathbf{n}_i] w_i(\mathbf{x})}{\sum_{\mathbf{p}_i \in \mathcal{P}'} w_i(x)}, \quad (2.16)$$

onde \mathbf{n}_i é a normal à superfície no ponto \mathbf{p}_i .

A seguir definiremos alguns conceitos que nos permitirão definir importantes propriedades das superfícies IMLS.

Definição 2.3.2 A função *lfs* (local feature size) de um ponto $\mathbf{p} \in \mathbb{R}^n$ é a distância do ponto \mathbf{p} ao eixo medial de Σ .

Definição 2.3.3 \mathcal{P} é uma ϵ -amostragem de Σ se a distância de qualquer ponto $\mathbf{p} \in \Sigma$ ao ponto mais próximo em \mathcal{P} é menor que $\epsilon \text{ lfs}(\mathbf{p})$. \mathcal{P} é uma ϵ -amostragem uniforme de Σ se \mathcal{P} é uma ϵ -amostragem de Σ e os pontos $\mathbf{p}_i \in \mathcal{P}$ são igualmente espaçados.

Supondo que \mathcal{P} é uma ϵ -amostragem uniforme de Σ , Kolluri [26] demonstrou que a superfície representada pelo nível zero da função implícita gerada pelo método IMLS é uma $(n-1)$ -variedade e é geometricamente próxima e homeomórfica a Σ . Como indicam Dey e Sun [16], a restrição de que \mathcal{P} deve ser uma ϵ -amostragem uniforme de Σ é, na prática, muito difícil de ser obtida, como demonstrado na figura 2.6.

Dey e Sun [16] mostraram que os mesmos resultados provados por Kolluri [26] valem se \mathcal{P} é uma ϵ -amostragem *tout court*, ou seja, a amostragem dos pontos pode ser “adaptativa” ao *local feature size*.

2.4 Superfícies MLS Algébricas

Mesmo com provas de corretude das superfícies IMLS, a necessidade de uma alta amostragem de pontos em regiões de alta curvatura (regiões cujo *local feature size* é muito

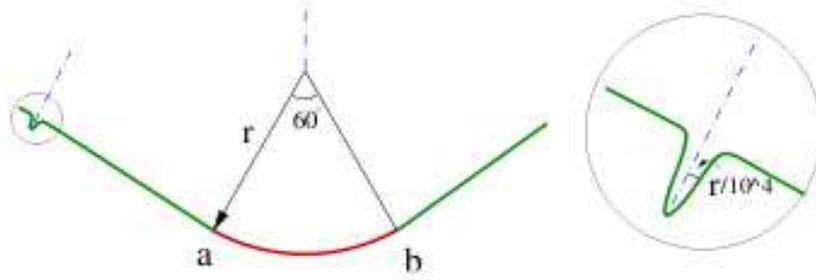


Figura 2.6: O arco ab requer 10^4 pontos devido à singularidade à esquerda. Os pontos tracejados representam o eixo medial da superfície [16].

pequeno) traz grandes dificuldades para seu uso em simulações de superfícies dinâmicas.

Para solucionar tal problema, Guennebaud e Gross [22] apresentaram uma nova definição de superfícies MLS a qual denominaram *Algebraic* MLS (AMLS). A ideia central deste método é aproximar a superfície por superfícies algébricas de alta ordem ao invés da aproximação planar, aumentando a estabilidade do método MLS nos casos em que a aproximação planar é ruim (ver Figura 2.4). A seguir descreveremos com detalhes esta técnica.

Seja $U \in \mathbb{R}^n$ uma vizinhança aberta do conjunto \mathcal{P} . Nosso objetivo é definir uma função implícita $F \in C^0(U)$ tal que seu conjunto de nível zero aproxime a superfície Σ . O primeiro passo é definir, para todo ponto $\mathbf{x} \in U$,

$$F(\mathbf{x}) = \sum_{k=1}^N \alpha_k(\mathbf{x}) q_k(\mathbf{x}), \quad (2.17)$$

onde a base $\{q_k\}_{k=1,N}$ consiste de polinômios. No caso da aproximação por esferas, temos:

$$N = 5, \quad q_1(\mathbf{x}) = 1, \quad q_2(\mathbf{x}) = x_1, \quad q_3(\mathbf{x}) = x_2, \quad q_4(\mathbf{x}) = x_3, \quad q_5(\mathbf{x}) = x_1^2 + x_2^2 + x_3^2 \quad (2.18)$$

Observação: Apesar de detalharmos a aproximação por esferas, o mesmo método pode ser modificado para casos bidimensionais (círculos) ou para aproximações por planos (3D) ou retas (2D). Por exemplo, no caso 2D, a aproximação por retas é dada por $N = 3$, com a base de polinômios dada por q_1 , q_2 e q_3 acima. Mostraremos exemplos destas bases alternativas no decorrer deste trabalho.

Denotando por $\underline{\alpha}$ a N -tupla de funções $(\alpha_1, \alpha_2, \dots, \alpha_N)$, fica claro que F é completamente determinada por $\underline{\alpha}$ por meio da Eq. 2.17. Toda vez que esta dependência precisar ser explicitada, indicaremos $\underline{\alpha}$ como um sub-índice; e.g., $F_{\underline{\alpha}}$. Cada $\underline{\alpha}$ também define, implicitamente, a superfície $\mathcal{S}_{\underline{\alpha}}$ dada por

$$\mathcal{S}_{\underline{\alpha}} = \{\mathbf{x} \in U \mid F_{\underline{\alpha}}(\mathbf{x}) = 0\}. \quad (2.19)$$

Considere inicialmente o caso em que cada α_k é constante. Neste caso, $\mathcal{S}_{\underline{\alpha}}$ é uma esfera. Denotando por $d(\mathbf{p}_i, \mathcal{S}_{\underline{\alpha}})$ a distância (com sinal) de \mathbf{p}_i a esta esfera, um conjunto ótimo de coeficientes $\underline{\alpha}^*$ pode ser definido como:

$$\underline{\alpha}^* = \arg \min_{\underline{\beta} \in \mathbb{R}^N} J(\underline{\beta}), \quad \text{com} \quad J(\underline{\beta}) = \sum_{i=1}^m \tilde{w}_i |d(\mathbf{p}_i, \mathcal{S}_{\underline{\beta}})|^2, \quad (2.20)$$

onde $\{\tilde{w}_i\}_{i=1, \dots, m}$ será usado posteriormente. Por enquanto, pode-se assumir $\tilde{w}_i = 1, i = 1, \dots, m$. Todavia, a minimização de J dada por (2.20) é numericamente inconveniente por dois motivos:

- devido à não-linearidade do problema, a solução só pode ser encontrada por meio de técnicas iterativas e computacionalmente caras.
- técnicas iterativas para a solução do problema são baseadas em representações paramétricas ou por centro e raio de esferas, logo tornam-se instáveis quando a aproximação ótima está próxima de um plano (esfera com raio muito grande).

Uma alternativa é substituir a distância geométrica pela distância algébrica. Esta abordagem define $\underline{\alpha}$ como:

$$\underline{\alpha} = \arg \min_{\underline{\beta} \in Q} \tilde{J}(\underline{\beta}), \quad \text{com} \quad \tilde{J}(\underline{\beta}) = \sum_{i=1}^m \tilde{w}_i |F_{\underline{\beta}}(\mathbf{p}_i)|^2. \quad (2.21)$$

sujeito a

$$Q = \{\underline{\beta} \in \mathbb{R}^N \mid \beta_2^2 + \beta_3^2 + \beta_4^2 - 4\beta_1\beta_5 = 1\} \quad (2.22)$$

A Eq. 2.22 é denominada restrição de Pratt [34] e assegura que a esfera algébrica aproximada é próxima à esfera geométrica aproximada. A razão para isto vem do fato que, sobre a superfície $\mathcal{S}_{\underline{\beta}}$,

$$\|\nabla F_{\underline{\beta}}\|^2 = \beta_2^2 + \beta_3^2 + \beta_4^2 - 4\beta_1\beta_5 \quad (2.23)$$

e, portanto, $F_{\underline{\beta}}(\mathbf{p}_i) = d(\mathbf{p}_i, \mathcal{S}_{\underline{\beta}}) + \mathcal{O}(|d(\mathbf{p}_i, \mathcal{S}_{\underline{\beta}})|^2)$ para todo $\underline{\beta} \in Q$.

A vantagem da abordagem algébrica é que sua solução é numericamente conveniente. De fato, já que

$$F_{\underline{\beta}}(\mathbf{p}_i) = \sum_{k=1}^N \beta_k q_k(\mathbf{p}_i) \quad (2.24)$$

segue que

$$\begin{aligned} \tilde{J}(\underline{\beta}) &= \sum_{i=1}^m \tilde{w}_i \left| \sum_{k=1}^N \beta_k q_k(\mathbf{p}_i) \right|^2 \\ &= \sum_{k,\ell=1}^N \beta_k \beta_\ell \left(\sum_{i=1}^m \tilde{w}_i q_k(\mathbf{p}_i) q_\ell(\mathbf{p}_i) \right) \\ &= \underline{\beta}^T \underline{\underline{M}} \underline{\beta} \end{aligned} \quad (2.25)$$

onde os elementos da matriz $\underline{\underline{M}}$ são dados por:

$$M_{k\ell} = \sum_{i=1}^m \tilde{w}_i q_k(\mathbf{p}_i) q_\ell(\mathbf{p}_i). \quad (2.26)$$

Agora podemos reescrever o problema de minimização (2.21)-(2.22) como sendo:

$$\begin{aligned} \underline{\alpha} = \arg \min_{\underline{\beta} \in \mathbb{R}^N} \quad & \underline{\beta}^T \underline{\underline{M}} \underline{\beta} \\ & \underline{\beta}^T \underline{\underline{C}} \underline{\beta} = 1, \end{aligned} \quad (2.27)$$

$$\text{onde } \underline{\underline{C}} = \begin{bmatrix} 0 & 0 & \cdots & 0 & -2 \\ 0 & 1 & & & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & & & 1 & 0 \\ -2 & 0 & \cdots & 0 & 0 \end{bmatrix}.$$

Desta forma, $\underline{\alpha}$ é obtido encontrando-se o autovetor do seguinte problema de autovalor generalizado:

$$\underline{\underline{M}} \underline{\alpha} = \lambda \underline{\underline{C}} \underline{\alpha}, \quad (2.28)$$

correspondente ao mínimo autovalor λ , e normalizado tal que $\underline{\alpha}^T \underline{\underline{C}} \underline{\alpha} = 1$.

Vale ressaltar que, para efeitos computacionais, a matriz $\underline{\underline{C}}^{-1}$ é dada por

$$\underline{\underline{C}}^{-1} = \begin{bmatrix} 0 & 0 & \cdots & 0 & -\frac{1}{2} \\ 0 & 1 & & & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & & & 1 & 0 \\ -\frac{1}{2} & 0 & \cdots & 0 & 0 \end{bmatrix}.$$

Portanto, o problema 2.27 pode ser reescrito como:

$$\underline{\underline{C}}^{-1} \underline{\underline{M}} \underline{\alpha} = \lambda \underline{\alpha}, \quad (2.29)$$

o qual resolvemos numericamente utilizando a fatorização de Schur, aplicável a matrizes não-simétricas, fazendo uso da biblioteca de domínio público *GNU Scientific Library* [20].

Para que $\underline{\underline{M}}$ seja não-singular, é necessário pelo menos quatro pontos, ou seja $m \geq 4$ (de fato, ainda é necessário que estes quatro pontos não pertençam a um círculo comum). Assim, o domínio U da função F pode ser definido como:

$$U = \{\mathbf{x} \in \mathbb{R}^n \mid \text{o número de pontos } \mathbf{p} \in \mathcal{P} \text{ tal que } \|\mathbf{x} - \mathbf{p}\| < \Delta \text{ seja } \geq 4\} \quad (2.30)$$

Para que esta metodologia gere superfícies de formas arbitrárias (não somente esferas), temos que acrescentar a estratégia de *moving-least-squares* (MLS) à minimização acima. Isto é feito simplesmente fazendo $\tilde{w}_i = w_i, i = 1, \dots, m$ onde w_i é dado por (2.1), já apresentado anteriormente. Agora podemos apresentar uma definição implícita das superfícies AMLS:

Definição 2.4.1 *A superfície \mathcal{S} dada pelo método AMLS é o conjunto zero da função implícita (2.17).*

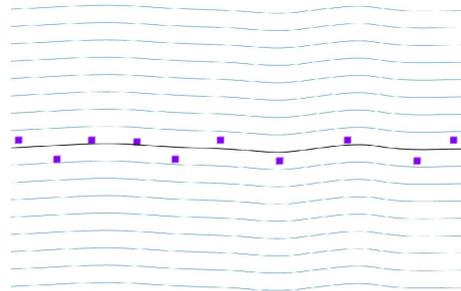
Alguns exemplos de F e \mathcal{S} são encontrados na Fig. 2.7.

2.5 A Orientação do Vetor Normal

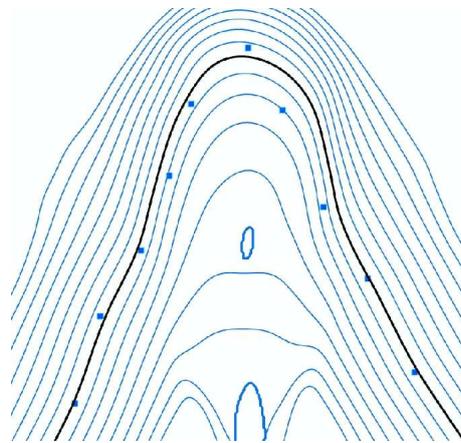
Vimos na Seção 2.2.1 que a aproximação do vetor normal $\mathbf{n}(\mathbf{x})$ à superfície SMLS pelo método da direção de menor covariância possui uma orientação arbitrária. Isto não interfere no nível zero da função implícita definida por 2.10 ou 2.11, porém interfere arbitrariamente no seu sinal. Isto também acontece com o cálculo da Função implícita 2.17 para superfícies AMLS. Neste caso, a normal estimada no ponto \mathbf{x} é aproximada pelo gradiente de F em \mathbf{x} , dado por:

$$\mathbf{g}_{\underline{\alpha}}(\mathbf{x}) = (\alpha_2(\mathbf{x}) + 2\alpha_5(\mathbf{x})x_1, \alpha_3(\mathbf{x}) + 2\alpha_5(\mathbf{x})x_2, \alpha_4(\mathbf{x}) + 2\alpha_5(\mathbf{x})x_3). \quad (2.31)$$

Qualquer algoritmo para detecção de interseções de uma reta (*ray*) com estas superfícies necessita que o sinal de F apenas se altere na interseção da superfície com a reta. Desta forma, é necessário orientar corretamente a aproximação do vetor normal



(a)



(b)

Figura 2.7: Exemplos da função implícita F gerada pelo método AMLS (contornos azuis) e superfícies AMLS \mathcal{S} (linha preta) a partir de dois conjunto de pontos (quadrados pretos). (a) Amostra de um segmento de reta perturbado aleatoriamente (b) Amostra de uma característica afiada (*sharp feature*) perturbada aleatoriamente.

encontrado. Para isto, assumimos que para cada ponto $\mathbf{p}_i \in \mathcal{P}$ temos um vetor \mathbf{n}_i associado, e calculamos uma aproximação das normais $\tilde{\mathbf{n}}(\mathbf{x})$ baseada no método descrito na Seção 2.2.2.

Finalmente, se o ângulo entre $\tilde{\mathbf{n}}(\mathbf{x})$ e $\mathbf{n}(\mathbf{x})$ (para SMLS) ou $\mathbf{g}_{\underline{\alpha}}(\mathbf{x})$ (para AMLS) for maior que $\pi/2$, então $\mathbf{n}(\mathbf{x})$ é substituído por $-\mathbf{n}(\mathbf{x})$ (para SMLS) ou $\underline{\alpha}(\mathbf{x})$ é substituído por $-\underline{\alpha}(\mathbf{x})$ (para AMLS).

Esta técnica funciona bem na maioria dos casos, porém há casos em que a geometria é suficientemente complexa fazendo com que esta técnica falhe.

CAPÍTULO 3

Metodologia

Nesse capítulo será detalhada a metodologia utilizada para a realização dos testes numéricos envolvendo superfícies dinâmicas. Para a representação da superfície dinâmica que separa os dois fluidos, optamos por utilizar as superfícies AMLS, apresentadas na Seção 2.4, pelas vantagens já discutidas neste trabalho. Primeiramente, realizamos um estudo de convergência das superfícies AMLS, o qual nos permitiu analisar a precisão destas superfícies quando utilizadas em um método de acompanhamento de fronteira. Em seguida, estudamos a robustez das superfícies AMLS na presença de características afiadas (*sharp features*) e verificamos problemas nestas regiões. Para solucionar este problema, desenvolvemos um método robusto na presença de características afiadas, e utilizamos este método de representação de superfície para desenvolver uma técnica de acompanhamento de fronteira, a qual utilizaremos nos testes numéricos descritos no Capítulo 4.

3.1 Estudos de Convergência das Superfícies AMLS

O sentido no qual uma superfície AMLS converge para uma superfície exata considerada *a priori* depende da aplicação. Se o conjunto \mathcal{P} foi obtido por meio de um *scanner* 3D com um certo nível de ruído, uma abordagem estatística é necessária para definir convergência. Para nossa aplicação, como será justificado posteriormente, assumimos que os pontos do conjunto \mathcal{P} estão exatamente sobre uma superfície Σ . Seja \mathcal{S} a superfície AMLS de \mathcal{P} . Desejamos estudar o quão próximo \mathcal{S} está de Σ .

Definimos a distância de Hausdorff entre duas superfícies como sendo:

$$d(\Sigma, \mathcal{S}) = \max\left\{\max_{\mathbf{x} \in \Sigma} \min_{\mathbf{y} \in \mathcal{S}} \|\mathbf{x} - \mathbf{y}\|, \max_{\mathbf{x} \in \mathcal{S}} \min_{\mathbf{y} \in \Sigma} \|\mathbf{x} - \mathbf{y}\|\right\} \quad (3.1)$$

Seja \mathcal{P}_h uma família de nuvem de pontos regularmente espaçados obtidas por amostragem da superfície “exata” Σ , com um espaçamento médio h entre os pontos. Seja \mathcal{S}_h a família de superfícies obtidas pelo método AMLS a partir de \mathcal{P}_h . Queremos saber se $d(\Sigma, \mathcal{S}_h) \rightarrow 0$ quando $h \rightarrow 0$ e, caso afirmativo, qual é a taxa de convergência r , no sentido que

$$d(\Sigma, \mathcal{S}_h) = \mathcal{O}(h^r) . \quad (3.2)$$

Caso de teste: Elipse

Estudamos a convergência das superfícies AMLS para o caso bidimensional de uma elipse Σ dada por

$$\frac{x^2}{9} + \frac{y^2}{4} = 1.$$

Para este propósito, geramos diversos conjuntos de pontos \mathcal{P}_h sobre a elipse com distâncias médias $h = 1, 5, 0, 75, 0, 375, \dots, 2, 34 \times 10^{-2}$, correspondendo a N_h pontos sobre a elipse, com $N_h = 14, 27, 54, \dots, 860$. O conjunto de pontos correspondente a $h = 1, 5$ e a correspondente curva AMLS são ilustrados na Fig. 3.1(a).

A Fig. 3.1(b) mostra o erro geométrico $d(\Sigma, \mathcal{S}_h)$ em função de h . A convergência é próxima a $\mathcal{O}(h^3)$. Com apenas 54 pontos amostrados a uma distância média de 0,375, a distância máxima entre \mathcal{S}_h e Σ já é muito pequena ($7,3 \times 10^{-4}$).

Neste caso bidimensional, a função implícita é dada por

$$F(\mathbf{x}) = \alpha_1(\mathbf{x}) + \alpha_2(\mathbf{x})x_1 + \alpha_3(\mathbf{x})x_2 + \alpha_4(\mathbf{x})(x_1^2 + x_2^2). \quad (3.3)$$

Em cada ponto \mathbf{x} de \mathcal{S}_h também comparamos a normal e a curvatura aproximada com a normal e a curvatura exata da elipse. A normal aproximada $\mathbf{n}_{\underline{\alpha}}(\mathbf{x})$ correspondente aos coeficientes de $\underline{\alpha}(\mathbf{x})$ é definida como a normal ao círculo aproximado pelo método AMLS em \mathbf{x} , e é dado por

$$\mathbf{n}_{\underline{\alpha}}(\mathbf{x}) = (\alpha_2(\mathbf{x}) + 2\alpha_4(\mathbf{x})x_1, \alpha_3(\mathbf{x}) + 2\alpha_4(\mathbf{x})x_2). \quad (3.4)$$

Seja $\mathbf{y}_{\mathbf{x}}$ o ponto em Σ mais próximo a \mathbf{y} . O erro das normais é definido por

$$E_N(\Sigma, \mathcal{S}_h) = \max_{\mathbf{x} \in \mathcal{S}_h} \|\mathbf{n}_{\underline{\alpha}}(\mathbf{x}) - \mathbf{n}(\mathbf{y}_{\mathbf{x}})\|, \quad (3.5)$$

onde $\mathbf{n}(\mathbf{y}_{\mathbf{x}})$ é a normal exata de Σ em $\mathbf{y}_{\mathbf{x}}$.

A curvatura aproximada é o inverso do raio do círculo aproximado pelo método AMLS, e é dada por

$$\kappa_{\underline{\alpha}}(\mathbf{x}) = \left(\frac{\alpha_2^2 + \alpha_3^2}{4\alpha_4^2} - \frac{\alpha_1}{\alpha_4} \right)^{-\frac{1}{2}}. \quad (3.6)$$

O erro da curvatura é definido como sendo

$$E_C(\Sigma, \mathcal{S}_h) = \max_{\mathbf{x} \in \mathcal{S}_h} \|\kappa_{\underline{\alpha}}(\mathbf{x}) - \kappa(\mathbf{y}_{\mathbf{x}})\|, \quad (3.7)$$

onde $\kappa(\mathbf{y}_{\mathbf{x}})$ é a curvatura exata de Σ em $\mathbf{y}_{\mathbf{x}}$.

A Fig. 3.1(b) ilustra gráficos de E_N e E_C em função de h .

Observação: Note que $\mathbf{n}_{\underline{\alpha}}(\mathbf{x})$ e $\kappa_{\underline{\alpha}}(\mathbf{x})$ não são, respectivamente, o vetor normal e a curvatura correspondente a \mathcal{S}_h (mais detalhes em [5, 22]), são apenas aproximações correspondentes ao círculo gerado pelo método AMLS em \mathbf{x} . Apesar disso, E_N e E_C apresentam taxas de convergência $\mathcal{O}(h^2)$ e $\mathcal{O}(h)$, respectivamente, ou seja, possuem o mesmo comportamento esperado das normais e curvaturas correspondentes a \mathcal{S}_h .

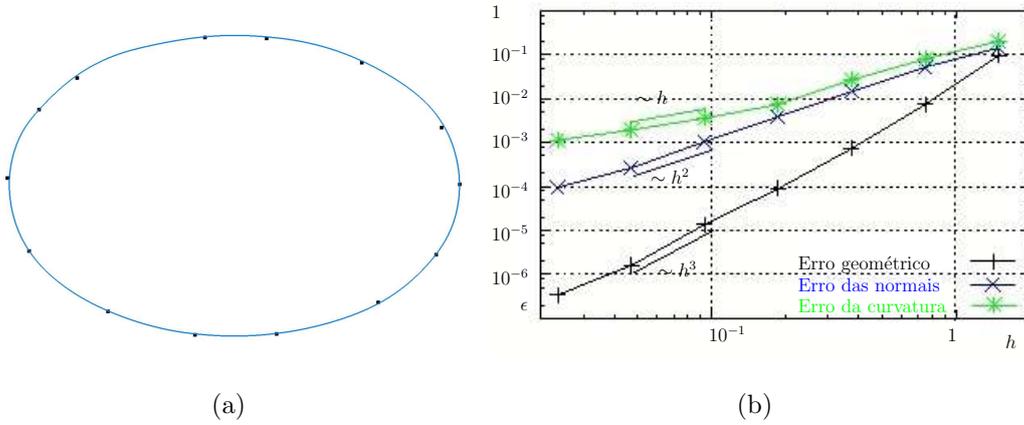


Figura 3.1: (a) \mathcal{P}_h correspondente a $h = 1, 5$ para o caso da elipse, aliado à curva AMLS baseada em círculos \mathcal{S}_h (b) Gráficos de convergência de $d(\Sigma, \mathcal{S}_h)$ (denotado por erro geométrico), de $E_N(\Sigma, \mathcal{S}_h)$ (denotado por erro das normais) e de $E_C(\Sigma, \mathcal{S}_h)$ (denotado por erro da curvatura).

O exemplo anterior ilustra a precisão das curvas AMLS baseada em círculos ao aproximar curvas suaves bidimensionais. O comportamento das curvas AMLS baseadas em retas foi estudado analogamente ao exemplo anterior, a menos da curvatura, já que a aproximação é feita por retas. A Fig. 3.2(a) ilustra a superfície resultante \mathcal{S}_h . A Fig. 3.2(b) mostra o gráfico de $d(\Sigma, \mathcal{S}_h)$ e de $E_N(\Sigma, \mathcal{S}_h)$ em função de h . Neste caso, o erro geométrico $d(\Sigma, \mathcal{S}_h)$ e o erro das normais apresentam taxas de convergência $\mathcal{O}(h^2)$ e $\mathcal{O}(h)$, respectivamente. Observamos a perda de uma ordem de convergência ao aproximar a superfície pelo método AMLS por retas, ao invés de círculos.

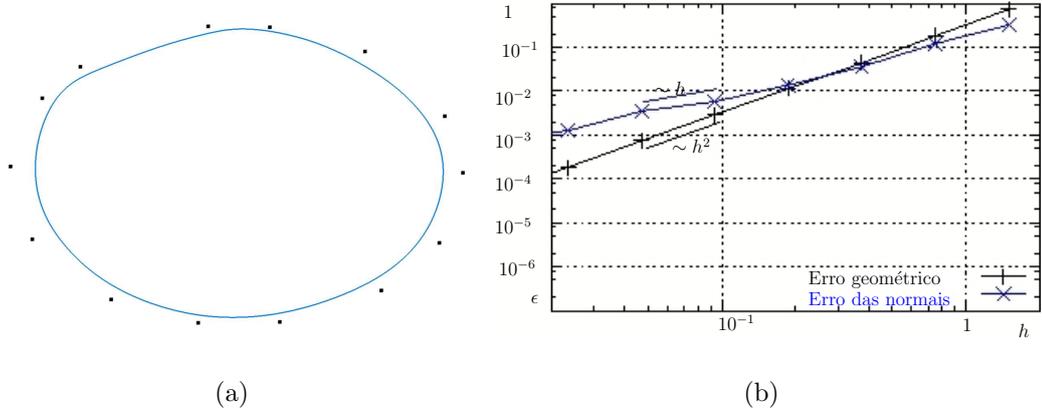


Figura 3.2: (a) \mathcal{P}_h correspondente a $h = 1,5$ para o caso da elipse, aliado à curva AMLS baseada em retas \mathcal{S}_h (b) Gráficos de convergência de $d(\Sigma, \mathcal{S}_h)$ (denotado por erro geométrico) e de $E_N(\Sigma, \mathcal{S}_h)$ (denotado por erro das normais).

Caso de teste: Elipsoide

Também estudamos a convergência das superfícies AMLS para o caso tridimensional de um elipsoide Σ dado por

$$\frac{x_1^2}{16} + \frac{x_2^2}{9} + \frac{x_3^2}{4} = 1. \quad (3.8)$$

Analogamente ao exemplo anteriormente estudado, \mathcal{P}_h tem um espaçamento médio $h = 0,75, 0,375, \dots, 2,34 \times 10^{-2}$, o que corresponde a $N_h = 265, 1.085, 4.369, \dots, 276.414$ pontos de Σ , respectivamente.

A Fig. 3.3(a) ilustra os gráficos de convergência para o método AMLS baseado em esferas. Vale ressaltar que em 3D as expressões para a normal e a curvatura da esfera aproximada no ponto \mathbf{x} são dadas por:

$$\mathbf{n}_{\underline{\alpha}}(\mathbf{x}) = (\alpha_2(\mathbf{x}) + 2\alpha_5(\mathbf{x})x_1, \alpha_3(\mathbf{x}) + 2\alpha_5(\mathbf{x})x_2, \alpha_4(\mathbf{x}) + 2\alpha_5(\mathbf{x})x_3) \quad (3.9)$$

e

$$\kappa_{\underline{\alpha}}(\mathbf{x}) = \left(\frac{\alpha_2^2 + \alpha_3^2 + \alpha_4^2}{4\alpha_5^2} - \frac{\alpha_1}{\alpha_5} \right)^{-\frac{1}{2}}. \quad (3.10)$$

Em comparação com o caso bidimensional, a taxa de convergência do erro geométrico neste exemplo 3D é reduzida para $\mathcal{O}(h^2)$. Evidentemente, a base de polinômios 2.18, com somente um elemento quadrático, é insuficiente para se obter uma taxa de convergência cúbica. Todavia, a aproximação ainda é precisa: com apenas 4.369 pontos, a uma distância média de $h = 0,375$, a distância entre Σ e \mathcal{S}_h é de $2,3 \times 10^{-3}$, duas ordens de magnitude menores que a distância média entre os pontos. O erro das normais E_N , por sua vez, possui taxa de convergência $\mathcal{O}(h)$. A curvatura da esfera aproximada, no entanto, não converge para a curvatura exata de Σ ($E_C = \mathcal{O}(1)$). Este é um problema que precisa ser resolvido em trabalhos futuros envolvendo, por exemplo, forças capilares. Caso seja realmente necessária uma aproximação convergente da curvatura, pode-se calcular uma aproximação de segunda ordem em um sistema de coordenadas locais, como no trabalho de Du et al [17].

Calculamos também as superfícies AMLS baseadas em planos para os mesmos conjuntos de pontos. A Fig. 3.3(b) mostra os gráficos de convergência. Podemos observar que as taxas de convergência são idênticas à das superfícies AMLS baseadas em esferas, e o erro das normais, $E_N(\Sigma, \mathcal{S}_h)$, é semelhante em ambas aproximações. Não obstante o erro geométrico $d(\Sigma, \mathcal{S}_h)$ é uma ordem de magnitude maior, indicando que o termo quadrático melhora substancialmente a aproximação. Finalmente, as Figuras 3.4(a) e (b) ilustram as superfícies AMLS baseadas em planos e esferas correspondente a $h = 0,75$ (265 pontos), respectivamente. Observamos que a convexidade do elipsoide faz com que a superfície AMLS baseada em planos fique dentro do fecho convexo dos pontos, enquanto que a superfície AMLS baseada em esferas aproxima com ótima precisão o elipsoide, considerando o número reduzido de pontos.

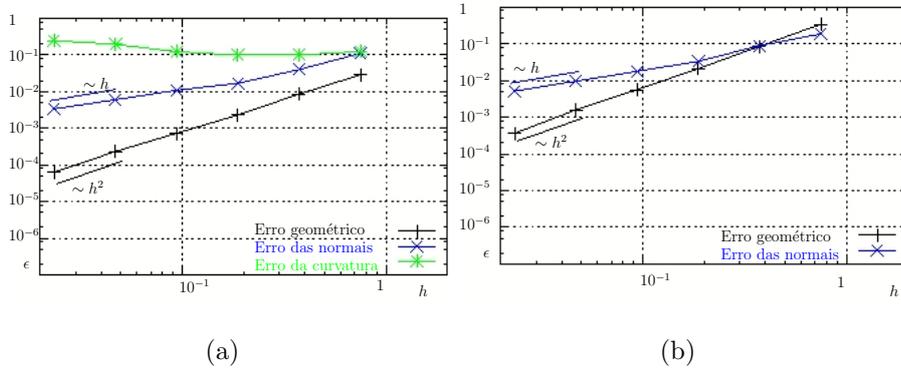


Figura 3.3: Gráficos de convergência de $d(\Sigma, \mathcal{S}_h)$ (denotado por erro geométrico), de $E_N(\Sigma, \mathcal{S}_h)$ (denotado por erro das normais), e de $E_C(\Sigma, \mathcal{S}_h)$ (denotado por erro da curvatura) para o caso tridimensional do elipsoide. (a) Superfície AMLS baseada em esferas. (b) Superfície AMLS baseada em planos.

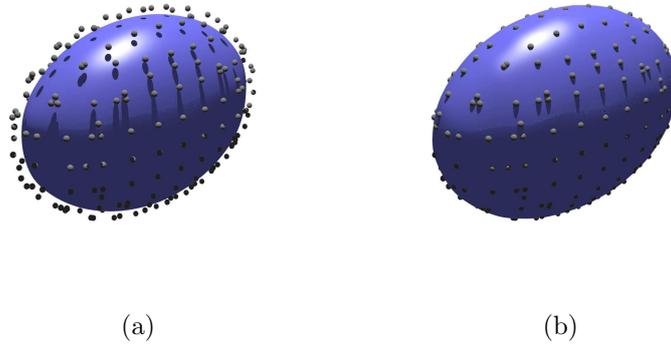


Figura 3.4: (a) Superfície AMLS baseada em planos para o caso do elipsoide (b) Superfície AMLS baseada em esferas para o caso do elipsoide. O conjunto de pontos consiste de 265 pontos a uma distância média de $h = 0,75$.

3.2 Características Afadas

Ao simular escoamentos bifásicos, a interface entre dois fluidos evolui no tempo, possivelmente apresentando características afadas (*sharp features*) ou mudanças de topologia

no decorrer da simulação. A capacidade de um método de lidar com superfícies não-suaves possui um relevante impacto na *robustez* da simulação numérica. Métodos menos precisos, como os métodos de captura de fronteira, são frequentemente utilizados em aplicações reais devido à robustez de suas soluções.

Nesta Seção apresentamos um estudo do comportamento das superfícies AMLS em presença de geometrias não-suaves. Como caso de teste, analisamos uma superfície tridimensional Σ obtida pela remoção de um quarto de uma esfera de raio unitário. Denotaremos este caso de 3/4-esfera. Construimos diversos conjuntos de pontos \mathcal{P}_h amostrando a 3/4-esfera com espaçamento médio de $h \simeq 0,12$ (1.030 pontos), $h \simeq 0,08$ (2.031 pontos) e $h \simeq 0,05$ (4.748 pontos).

A Fig. 3.5 mostra as superfícies AMLS baseadas em esferas obtidas destes conjuntos de pontos. É evidente que a aproximação apresenta defeitos na vizinhança das características afiadas (onde as duas seções planares interceptam a parte esférica). De fato, uma superfície espúria pode ser observada na vizinhança das regiões com características afiadas. Isto representa uma falha do método, e constitui um problema de robustez, podendo levar a comportamentos instáveis quando utilizado em uma simulação de um escoamento bifásico.

3.3 Superfícies AMLS com Robustez Aprimorada

3.3.1 Superfícies RAMLS

O último parágrafo da Seção anterior mostra que a metodologia AMLS requer aprimoramentos para que seu comportamento no caso de superfícies com características afiadas seja adequado e robusto. Para resolver este problema, Guennebaud and Gross [22] propuseram um método baseado em uma minimização utilizando as normais aproximadas pelo método AMLS nos pontos $\mathbf{p}_i \in \mathcal{P}$. Este método consiste de duas etapas:

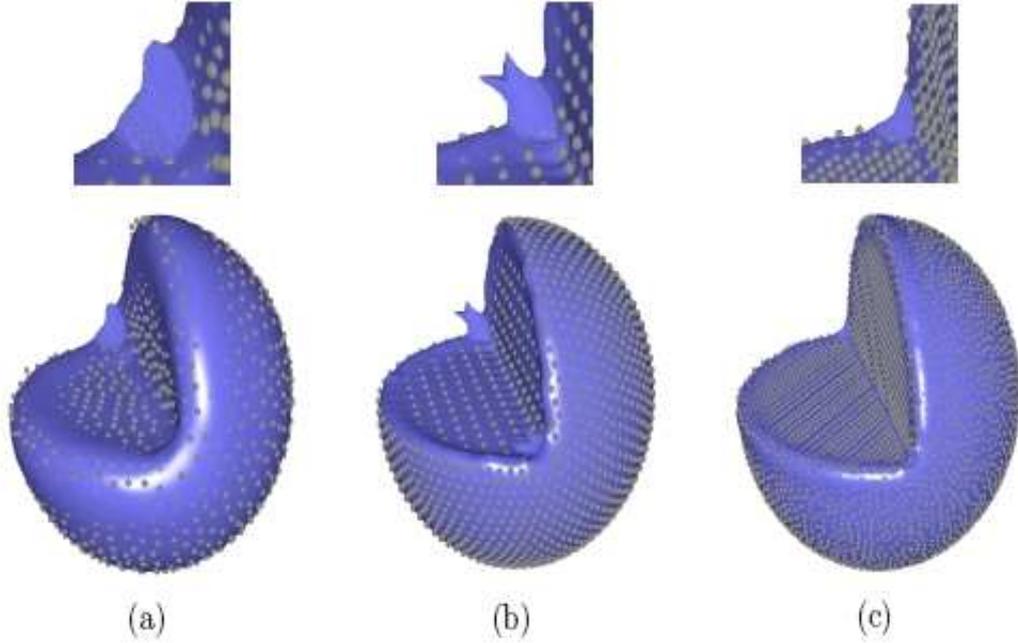


Figura 3.5: Superfícies AMLS baseadas em esferas para o caso 3/4-esfera de um conjunto de pontos constituído de 1.030 pontos (a), de 2.031 pontos (b) e de 4.748 pontos (c). Superfícies espúrias detalhadas na parte superior da imagem ilustram a presença de artefatos na vizinhança de características afiadas.

Etapa A *Calcular as normais aproximadas pelo método AMLS para todo $\mathbf{p}_i \in \mathcal{P}_h$:*

Este processo é feito calculando $\underline{\alpha}$ a partir de (2.28), porém somente nos pontos \mathbf{p}_i do conjunto \mathcal{P}_h . As normais $\mathbf{N}_i = \mathbf{n}_{\underline{\alpha}}(\mathbf{p}_i)$, $i = 1, \dots, m$ são assim obtidas pela Eq. (3.9) no caso tridimensional, e pela Eq. (3.4) no caso bidimensional. O problema da orientação (ver Seção 2.5) é solucionado por meio de uma árvore geradora mínima [22].

Etapa B *Calcular a função implícita $F(\mathbf{x})$ para $\mathbf{x} \in U$:* Para isto, os coeficientes $\underline{\alpha}(\mathbf{x})$ não são encontrados pela minimização de \tilde{J} como definido em (2.21)-(2.22). Ao

invés disso, $\underline{\alpha}(\mathbf{x})$ é calculado como sendo

$$\underline{\alpha}(\mathbf{x}) = \arg \min_{\underline{\beta} \in \mathbb{R}^N} G(\underline{\beta}) \quad (3.11)$$

com

$$G(\underline{\beta}) = \sum_{i=1}^m w_i |F_{\underline{\beta}}(\mathbf{p}_i)|^2 + K \sum_{i=1}^m w_i \|\mathbf{n}_{\underline{\beta}}(\mathbf{p}_i) - \mathbf{N}_i\|^2. \quad (3.12)$$

Se compararmos a Formulação acima com a Formulação (2.21)-(2.22), podemos observar que a minimização (3.11) é irrestrita. A condição $\|\nabla F_{\underline{\beta}}\| \simeq 1$ em \mathcal{S}_{β} , a qual faz com que $F_{\underline{\beta}}$ aproxime a distância com sinal, é de fato assegurada pelo segundo termo em $G(\underline{\beta})$, Eq. 3.12. Este termo considera os desvios das normais aproximadas pelo método AMLS nos pontos de \mathcal{P} em relação às normais calculadas na Etapa 1 do algoritmo. Como a segunda etapa do algoritmo é um problema de minimização sem restrição, ela pode ser resolvida por meio de um sistema linear pequeno, e seu custo computacional é bem menor em comparação ao problema de autovalor da Eq. (2.28).

O problema do método acima é a escolha do parâmetro K em (3.12). Este parâmetro ajusta a importância da consideração dos desvios das normais, explicitada acima, em comparação com a minimização da distância algébrica dada pelo primeiro termo da Eq. (3.12). A escolha deste parâmetro não é trivial e pode afetar a precisão e estabilidade numérica do método. Guennebaud e Gross utilizam $K = 10^6 h^2$, sem apresentar estudo numérico algum para a utilização deste valor. Para evitar introduzir um parâmetro no nosso método numérico, propomos neste projeto uma variante do método descrito acima. De fato, esta variante pode ser vista como o limite da formulação acima quando $K \rightarrow \infty$. Uma importante observação é o fato de que o segundo termo de G (Eq. 3.12) não depende de α_1 , assim, quando K é muito grande, o primeiro termo de G é utilizado apenas para determinar α_1 . Ao invés disso, podemos dividir o vetor de coeficientes em duas partes, separando o primeiro coeficiente:

$$\underline{\beta} \in \mathbb{R}^N, \quad \underline{\beta} = (\beta_1, \hat{\underline{\beta}}),$$

onde $\underline{\hat{\beta}} = (\beta_2, \dots, \beta_N)$ pertence a \mathbb{R}^{N-1} . Como para qualquer vetor de coeficientes $\underline{\beta}$ a normal em qualquer ponto \mathbf{x} definida por (3.9) não depende de β_1 , a denotamos por $\mathbf{n}_{\underline{\beta}}$.

Detalharemos o método proposto por meio de um algoritmo para calcular os coeficientes $\underline{\alpha}(\mathbf{x})$ em um ponto $\mathbf{x} \in U$. A partir de agora denominaremos este método de RAMLS (*Robust AMLS*), e as superfícies geradas por este método de Superfícies RAMLS. A função implícita do método RAMLS é a mesma: $F(\mathbf{x}) = \sum_{k=1}^N \alpha_k(\mathbf{x}) q_k(\mathbf{x})$, e continuará sendo denotada por $F(\mathbf{x})$, já que a única diferença entre os métodos AMLS e RAMLS é a escolha dos coeficientes $\underline{\alpha}(\mathbf{x})$. Do mesmo modo que as Superfícies AMLS, as Superfícies RAMLS são o conjunto de nível zero da função implícita $F(\mathbf{x})$.

Algoritmo proposto (método RAMLS):

Etapa A *Calcular as normais aproximadas $\mathbf{N}_i = \mathbf{n}_{\underline{\alpha}}(\mathbf{p}_i)$ pelo método AMLS para todo $\mathbf{p}_i \in \mathcal{P}_h$. Este processo é feito calculando $\underline{\alpha}$ a partir de (2.28), porém somente nos pontos \mathbf{p}_i do conjunto \mathcal{P}_h . As normais \mathbf{N}_i , são assim obtidas por (3.9) no caso tridimensional, e por (3.4) no caso bidimensional. Como em \mathbf{p}_i já temos uma normal aproximada $\tilde{\mathbf{n}}_i$, a orientação de \mathbf{N}_i é escolhida tal que o ângulo entre \mathbf{N}_i e $\tilde{\mathbf{n}}_i$ seja menor que $\pi/2$.*

Etapa B *Calcular a função implícita $F(\mathbf{x})$ para $\mathbf{x} \in U$: O cálculo dos coeficientes $\underline{\alpha}(\mathbf{x})$ é dividido em duas sub-etapas, descritos a seguir:*

Sub-etapa B.1 Determinar $\hat{\underline{\alpha}}(\mathbf{x})$ por meio de

$$\hat{\underline{\alpha}}(\mathbf{x}) = \arg \min_{\underline{\hat{\beta}} \in \mathbb{R}^{N-1}} \sum_{i=1}^m w_i \|\mathbf{n}_{\underline{\hat{\beta}}}(\mathbf{p}_i) - \mathbf{N}_i\|^2. \quad (3.13)$$

Como consequência, $\hat{\underline{\alpha}}(\mathbf{x})$ é a solução do sistema linear

$$\underline{A} \hat{\underline{\alpha}} = \underline{b} \quad (3.14)$$

onde

$$A_{k\ell} = \sum_{i=1}^m w_i \nabla q_k(\mathbf{p}_i) \cdot \nabla q_\ell(\mathbf{p}_i) \quad (3.15)$$

$$b_k = \sum_{i=1}^m w_i \nabla q_k(\mathbf{p}_i) \cdot \mathbf{N}_i \quad (3.16)$$

e em particular para o método baseado em esferas (base definida na Eq. 2.18), e denotando por (X_i, Y_i, Z_i) as coordenadas de \mathbf{p}_i e por (N_{i1}, N_{i2}, N_{i3}) as componentes de \mathbf{N}_i ,

$$\underline{\underline{A}} = \begin{pmatrix} \sum_{i=1}^m w_i & 0 & 0 & 2 \sum_{i=1}^m w_i X_i \\ 0 & \sum_{i=1}^m w_i & 0 & 2 \sum_{i=1}^m w_i Y_i \\ 0 & 0 & \sum_{i=1}^m w_i & 2 \sum_{i=1}^m w_i Z_i \\ 2 \sum_{i=1}^m w_i X_i & 2 \sum_{i=1}^m w_i Y_i & 2 \sum_{i=1}^m w_i Z_i & 4 \sum_{i=1}^m w_i \|\mathbf{p}_i\|^2 \end{pmatrix} \quad (3.17)$$

$$\underline{\underline{b}} = \begin{pmatrix} \sum_{i=1}^m w_i N_{i1} \\ \sum_{i=1}^m w_i N_{i2} \\ \sum_{i=1}^m w_i N_{i3} \\ 2 \sum_{i=1}^m w_i \mathbf{p}_i \cdot \mathbf{N}_i \end{pmatrix} \quad (3.18)$$

Devido à simplicidade da matriz $\underline{\underline{A}}$, $\hat{\alpha}$ pode ser encontrado diretamente, sem a necessidade de métodos para solução de sistemas lineares.

Sub-etapa B.2 Agora que $\hat{\alpha}(\mathbf{x}) = (\alpha_2(\mathbf{x}), \dots, \alpha_N(\mathbf{x}))$ é conhecido, obtemos $\alpha_1(\mathbf{x})$ por meio da minimização

$$\sum_{i=1}^m w_i |F_{\underline{\underline{\beta}}}(\mathbf{p}_i)|^2 \quad (3.19)$$

sobre todos $\underline{\underline{\beta}}$ tal que $\hat{\underline{\underline{\beta}}} = \hat{\alpha}(\mathbf{x})$. A solução para este problema é dada por

$$\alpha_1(\mathbf{x}) = - \frac{\sum_{i=1}^m w_i \left(\sum_{k=2}^N \alpha_k(\mathbf{x}) q_k(\mathbf{p}_i) \right)}{\sum_{i=1}^m w_i}. \quad (3.20)$$

Para o caso do método RAMLS baseado em esferas, a equação acima é descrita por

$$\alpha_1(\mathbf{x}) = -\frac{\sum_{i=1}^m w_i [\alpha_2(\mathbf{x}) X_i + \alpha_3(\mathbf{x}) Y_i + \alpha_4(\mathbf{x}) Z_i + \alpha_5(\mathbf{x}) \|\mathbf{p}_i\|^2]}{\sum_{i=1}^m w_i} \quad (3.21)$$

É importante ressaltar que, como os coeficientes $\alpha_2, \dots, \alpha_N$ são determinados por uma minimização de desvios de normais, e *não* pela minimização da distância algébrica, as taxas de convergência do erro geométrico, do erro das normais e da curvatura podem ser diferentes das obtidas pelo método AMLS. A Seção 3.3.2 apresenta os estudos de convergência para o método proposto.

Também é relevante observar que o custo computacional do método RAMLS é muito menor em comparação com o método AMLS. Neste método, um problema de autovalor deve ser resolvido para cada \mathbf{x} onde a função implícita será avaliada, enquanto que no método RAMLS, este mesmo problema é resolvido apenas para cada \mathbf{p} , e o cálculo de F para cada \mathbf{x} envolve somente a solução de um sistema linear de dimensão $N - 1$.

Observação: Note que o problema da orientação discutido na Seção 2.5 é tratado de forma natural no método RAMLS. No método AMLS, *para cada* avaliação da função implícita há a necessidade de orientar o vetor de coeficientes encontrado. Já no método RAMLS, a orientação é dada naturalmente pela minimização baseada nas normais, logo não é necessário preocupar-se com a orientação do vetor de coeficientes.

3.3.2 Estudo de Convergência das Superfícies RAMLS e de Robustez na Presença de *Sharp Features*

Realizamos exatamente os mesmos testes da Seção 3.1, a saber, os testes de convergência com a elipse e o elipsoide, e o teste de *sharp features* com o modelo 3/4-esfera porém agora utilizando o método RAMLS. As Figs. 3.6(b) e 3.7(b) mostram os gráficos de

convergência para os dois primeiros casos, respectivamente. Observamos o mesmo comportamento assintótico em termos de convergência para o método RAMLS em comparação com o método AMLS. Contudo, o aprimoramento em robustez é significativo, como ilustra a Fig. 3.8, em comparação à Fig. 3.5.

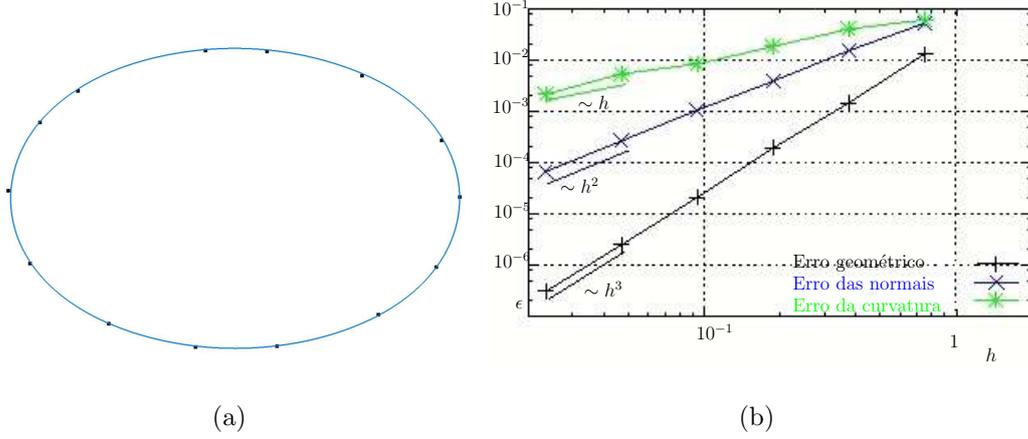
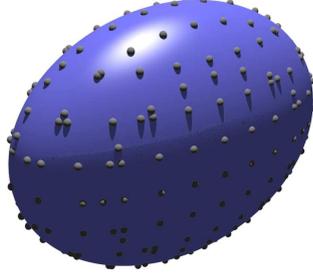


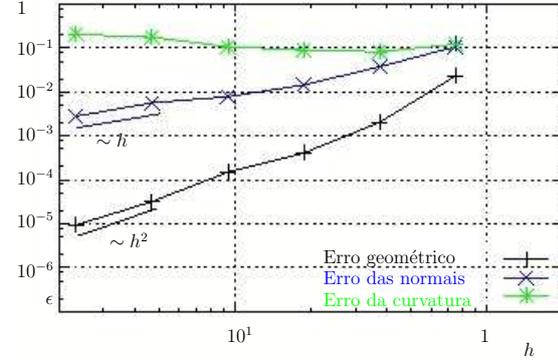
Figura 3.6: (a) \mathcal{P}_h correspondente a $h = 1,5$ para o caso da elipse, aliado à curva RAMLS baseada em círculos \mathcal{S}_h (b) Gráficos de convergência de $d(\Sigma, \mathcal{S}_h)$ (denotado por erro geométrico), de $E_N(\Sigma, \mathcal{S}_h)$ (denotado por erro das normais) e de $E_C(\Sigma, \mathcal{S}_h)$ (denotado por erro da curvatura).

Observação: Trabalhos anteriores [22] utilizam a curvatura da esfera aproximada pelo método AMLS como uma estimativa da curvatura da superfície aproximada Σ . Entretanto, os estudos de convergência deste projeto mostram que tal estimativa para ambas Superfícies AMLS e RAMLS somente convergem para a curvatura exata de Σ no caso bidimensional. Para análises qualitativas, a Fig. 3.9 ilustra como tal estimativa da curvatura pelo método RAMLS se comporta para o caso do elipsoide, com o parâmetro $h = 1/128$.

Para fins de comparação, a Fig. 3.10 apresenta a curva e a função implícita gerada pelo método RAMLS para o mesmo conjunto de pontos da Fig. 2.7.



(a)



(b)

Figura 3.7: (a) \mathcal{P}_h correspondente a $h = 1,5$ para o caso do elipsoide, aliado à curva RAMLS baseada em círculos \mathcal{S}_h (b) Gráficos de convergência de $d(\Sigma, \mathcal{S}_h)$ (denotado por erro geométrico), de $E_N(\Sigma, \mathcal{S}_h)$ (denotado por erro das normais) e de $E_C(\Sigma, \mathcal{S}_h)$ (denotado por erro da curvatura).

3.4 Acompanhamento de Fronteira com Superfícies RAMLS

3.4.1 O Algoritmo

Nesta Seção introduziremos um método de acompanhamento de fronteira (*front-tracking*) baseado em superfícies RAMLS. Consideraremos o caso tridimensional, já que o caso bidimensional é análogo e facilmente deduzido do caso 3D. Dada uma superfície inicial $\mathcal{S}(0) \subset \Omega$, dentro de um domínio delimitado $\Omega \subset \mathbb{R}^3$, junto com um campo de velocidade $\mathbf{v}(\mathbf{x}, t)$ para todo $\mathbf{x} \in \Omega$, onde T é o tempo da simulação, a trajetória lagrangiana $\varphi(\mathbf{x}, s, t)$ de uma partícula que está em \mathbf{x} no instante s e se move de acordo com o campo de velocidade $\mathbf{v}(\mathbf{x}, t)$ é a solução de

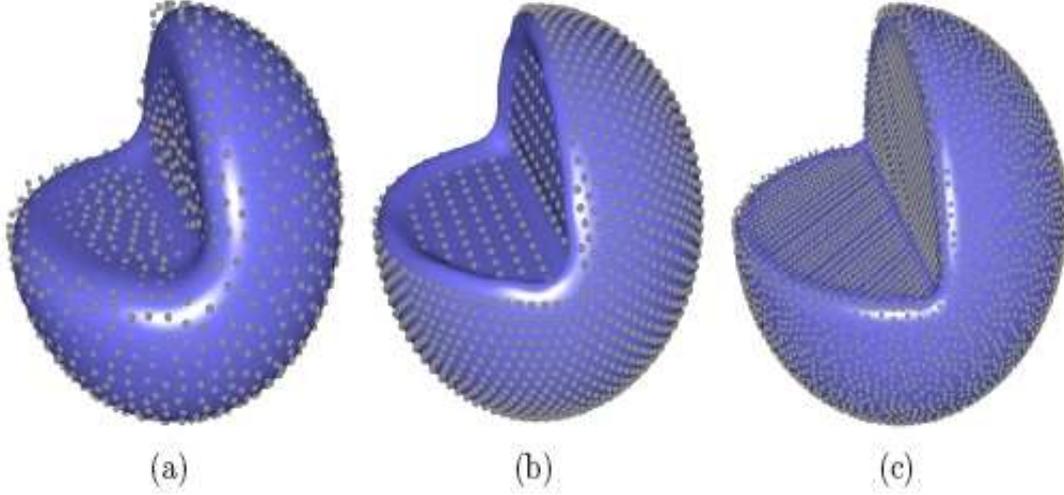


Figura 3.8: Superfícies RANLS baseadas em esferas para o caso 3/4-esfera de um conjunto de pontos constituído de 1.030 pontos (a), de 2.031 pontos (b) e de 4.748 pontos (c).

$$\frac{\partial \varphi}{\partial t}(\mathbf{x}, s, t) = \mathbf{v}(\varphi(\mathbf{x}, s, t), t); \quad \varphi(\mathbf{x}, s, s) = \mathbf{x} \quad (3.22)$$

e o problema a ser solucionado é encontrar $\mathcal{S}(t)$ definido por

$$\mathcal{S}(t) = \{\mathbf{y} \in \Omega, \mathbf{y} = \varphi(\mathbf{x}, 0, t), \mathbf{x} \in \mathcal{S}(0)\}. \quad (3.23)$$

Assumimos que $\mathcal{S}(t) \subset \Omega \forall t \in [0, T]$.

A discretização é feita utilizando-se uma malha estruturada para subdividir o domínio Ω . Consideraremos neste trabalho uma malha cartesiana \mathcal{T}_h com espaçamento uniforme h , embora a extensão deste trabalho para malhas não-uniformes seja possível. A malha é utilizada apenas para definir um conjunto de retas R_h (ou “raios”, no jargão de Computação Gráfica):

$$R_h = R_{1h} \cup R_{2h} \cup R_{3h} \quad (3.24)$$

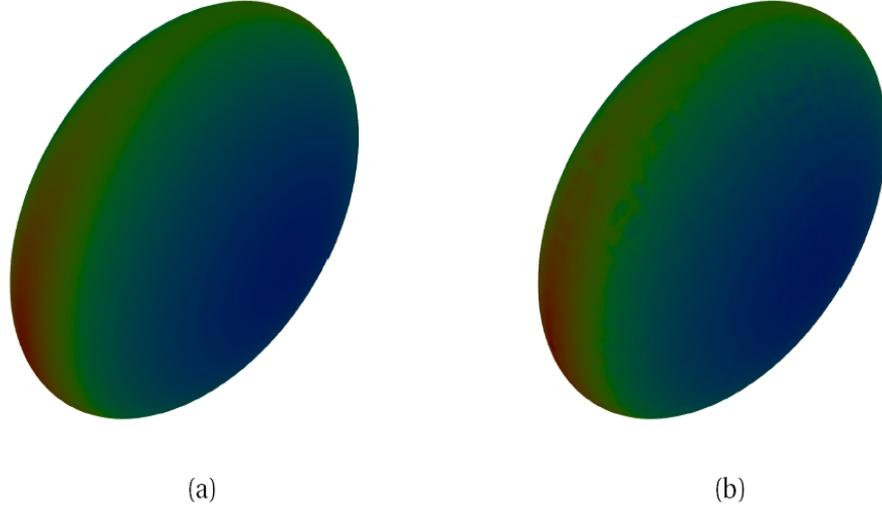


Figura 3.9: Exemplo do comportamento da curvatura exata (a) e da estimativa pelo método RAMLS por esferas (b) para o conjunto de pontos do elipsoide com $h = 1/128$.

onde os conjuntos R_{1h} , R_{2h} e R_{3h} , os quais estão alinhados com os eixos x_1 , x_2 , e x_3 , são dados, respectivamente, por

$$R_{1h} = \bigcup_{j,k \in \mathbb{Z}} r_{1jk}; \quad r_{1jk} = \{\mathbf{x} = (x_1, x_2, x_3) \in \Omega, x_2 = jh, x_3 = kh\} \quad (3.25)$$

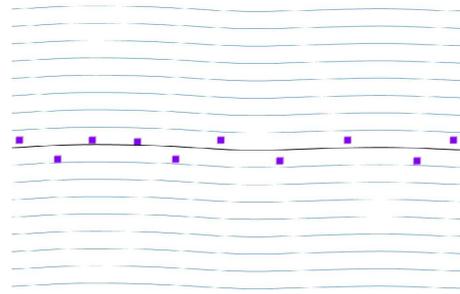
$$R_{2h} = \bigcup_{i,k \in \mathbb{Z}} r_{2ik}; \quad r_{2ik} = \{\mathbf{x} = (x_1, x_2, x_3) \in \Omega, x_1 = ih, x_3 = kh\} \quad (3.26)$$

$$R_{3h} = \bigcup_{i,j \in \mathbb{Z}} r_{3ij}; \quad r_{3ij} = \{\mathbf{x} = (x_1, x_2, x_3) \in \Omega, x_1 = ih, x_2 = jh\}. \quad (3.27)$$

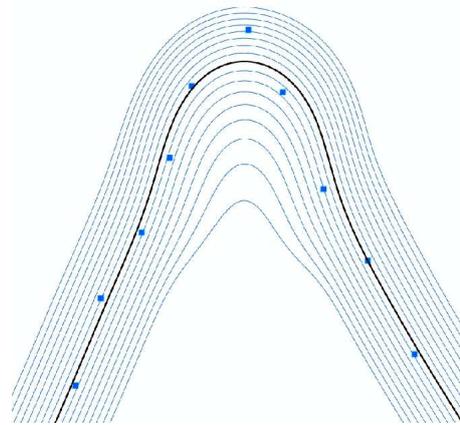
O conjunto de retas R_h , por sua vez, define um conjunto de pontos $\mathcal{Q}_h(0)$ como sendo a interseção de R_h com $\mathcal{S}(0)$; i.e.,

$$\mathcal{Q}_h(0) = R_h \cap \mathcal{S}(0). \quad (3.28)$$

O conjunto de pontos $\mathcal{Q}_h(0)$ é então transportado de acordo com o campo de velocidade \mathbf{v} de $t = 0$ a $t = \Delta t$ (passo de tempo), por meio da solução de (3.22), o que leva



(a)



(b)

Figura 3.10: Exemplos da função implícita F gerada pelo método RAMLS (contornos azuis) e superfícies RAMLS \mathcal{S} (linha preta) a partir de dois conjunto de pontos (quadrados pretos). (a) Amostra de um segmento de reta perturbado aleatoriamente (b) Amostra de uma característica afiada (*sharp feature*) perturbado aleatoriamente. Comparar com as superfícies AMLS da Fig. 2.7

a:

$$\mathcal{P}_h(\Delta t) = \varphi(\mathcal{Q}_h(0), 0, \Delta t) = \{\mathbf{y} \in \Omega, \mathbf{y} = \varphi(\mathbf{x}, 0, \Delta t), \mathbf{x} \in \mathcal{Q}_h(0)\}, \quad (3.29)$$

onde $\varphi(\mathcal{Q}_h(s), s, s + \Delta t)$ denota o conjunto de pontos obtidos ao transportar os pontos de $\mathcal{Q}_h(s)$ ao longo das trajetórias induzidas pelo campo de velocidade \mathbf{v} do instante s

ao instante $s + \Delta t$.

Finalmente, conjunto de pontos $\mathcal{P}_h(\Delta t)$ define a nova superfície $\mathcal{S}_h(\Delta t)$ por meio da técnica RAMLS descrita anteriormente. Denotando por $\mathbf{S}_{\text{RAMLS}}$ o operador que associa um conjunto de pontos com a Superfície RAMLS correspondente, temos que

$$\mathcal{S}_h(\Delta t) = \mathbf{S}_{\text{RAMLS}}(\mathcal{P}_h(\Delta t)). \quad (3.30)$$

A partir disto, o procedimento pode ser repetido *ad indefinitum*. O método proposto é definido da seguinte maneira:

Inicialização: Definir um conjunto de raios R_h , o passo de tempo Δt , e a superfície inicial $\mathcal{S}(0)$.

Passo 0: Definir $\mathcal{Q}_h(0) = R_h \cap \mathcal{S}(0)$.

Passo 1: Com $\mathcal{Q}_h(t)$ em mãos, computar as trajetórias de todos pontos $\mathcal{Q}_h(t)$ de t a $t + \Delta t$. Definir

$$\mathcal{P}_h(t + \Delta t) = \varphi(\mathcal{Q}_h(t), t, t + \Delta t) \quad (3.31)$$

Passo 2: Sendo $\mathcal{S}_h(t + \Delta t)$ a Superfície RAMLS associada a $\mathcal{P}_h(t + \Delta t)$ (i.e., $\mathcal{S}_h(t + \Delta t) = \mathbf{S}_{\text{RAMLS}}(\mathcal{P}_h(t + \Delta t))$), computar $\mathcal{Q}_h(t + \Delta t)$ por meio da interseção de R_h com $\mathcal{S}_h(t + \Delta t)$,

$$\mathcal{Q}_h(t + \Delta t) = R_h \cap \mathcal{S}_h(t + \Delta t) \quad (3.32)$$

Passo 3: Atribuir $t \leftarrow t + \Delta t$ e voltar ao Passo 1.

Uma iteração do método pode ser resumida como sendo o *transporte do conjunto de pontos obedecendo suas trajetórias*, seguido pela *interseção dos raios com a superfície definida pelos pontos deslocados* a fim de gerar um novo conjunto de pontos. O primeiro item considerado (Passo 1) consiste em resolver uma equação diferencial ordinária (3.22)

no tempo, e para isso foi utilizado neste projeto um método de Runge-Kutta de quarta ordem. O segundo item considerado (Passo 2), o qual é a essência do método proposto, será abordado com mais detalhes na Seção 3.4.2.

A Fig. 3.11 ilustra um exemplo bidimensional do método de acompanhamento de fronteira utilizando Superfícies RAMLS.

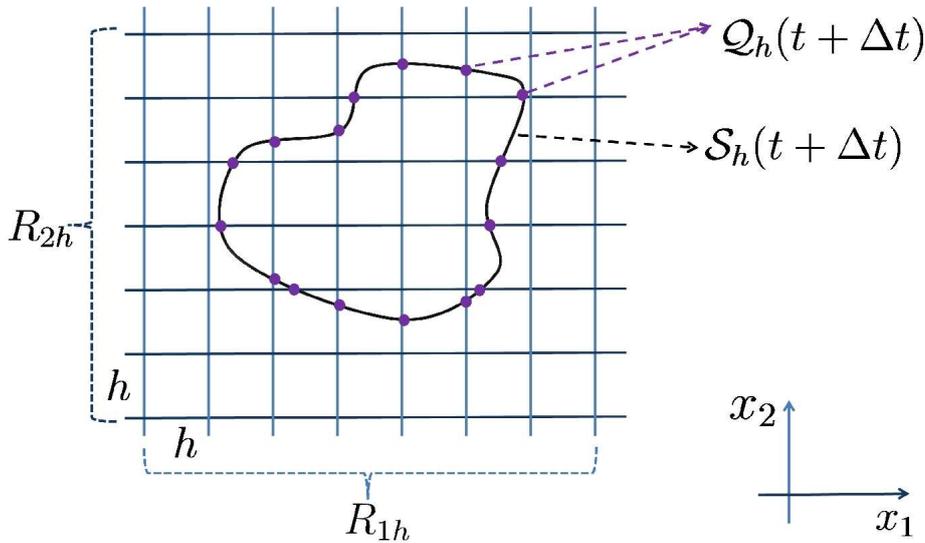


Figura 3.11: Novo conjunto de pontos $\mathcal{Q}_h(t + \Delta t)$ gerado pela interseção do conjunto de raios (R_{1h} , R_{2h}) com a Superfície $\mathcal{S}_h(t + \Delta t)$.

3.4.2 Implementação da Interseção de Raios com a Superfície RAMLS

O Passo 2 do método proposto consiste em determinar as interseções do conjunto de raios R_h com a superfície $\mathcal{S}_h(t + \Delta t)$ definida pelo conjunto de pontos $\mathcal{P}_h(t + \Delta t)$. Este procedimento é similar à técnica de *ray-tracing* [4, 45] utilizada para calcular a iluminação de cenas em computação gráfica. No nosso caso, como acontece no cálculo de iluminação em que a cena consiste de superfícies semi-transparentes [4], todas interseções que acontecem ao longo do raio são calculadas (e não apenas a primeira, no caso de

objetos opacos). Em termos da função implícita F , para cada raio $r \in R_h$, é necessário encontrar todos pontos $\mathbf{x} \in r$ tal que $F(\mathbf{x}) = 0$.

Implementamos tal procedimento baseado em duas considerações:

- a função F é definida apenas na vizinhança U de $\mathcal{S}_h(t + \Delta t)$;
- a avaliação da função F é computacionalmente cara (cerca de 1.000 operações de ponto flutuante no caso tridimensional).

Seja \mathcal{B} o conjunto de esferas de raio h centrada nos pontos $\mathbf{p}_i \in \mathcal{P}_h(t + \Delta t)$, i.e.,

$$\mathcal{B} = \{\mathbf{b}_i : \mathbf{x} \in \mathbf{b}_i \Rightarrow \|\mathbf{x} - \mathbf{p}_i\| \leq h, \mathbf{p}_i \in \mathcal{P}_h(t + \Delta t), i \in 1, \dots, m\}, \quad (3.33)$$

e seja V a união destas esferas ($V = \bigcup_{i=1}^m \mathbf{b}_i$). Por construção, temos que para qualquer $\mathbf{q} \in \mathcal{Q}_h(t)$, existe outro ponto $\mathbf{p} \in \mathcal{Q}_h(t)$ tal que $\|\mathbf{q} - \mathbf{p}\| < \sqrt{3}h$ (diagonal do cubo de tamanho h do *grid* euleriano) então, para um Δt pequeno o suficiente, podemos garantir que $\mathcal{S}_h(t + \Delta t) \subset V \subset U$. A ideia é calcular analiticamente o segmento $\overline{\mathbf{p}_{in}\mathbf{p}_{out}}$, definido como sendo a interseção do raio r com a esfera $\mathbf{b}_i \in \mathcal{B}$, e então encontrar o zero da função F neste segmento pelo método regula-falsi.

Destarte, a implementação do Passo 2 do método proposto é feita da seguinte maneira:

Passo 2:

2.0: Pré-processamento

2.0.1: Remover pontos de $\mathcal{Q}_h(t + \Delta t)$ para que dois pontos quaisquer não estejam a uma distância menor que $\frac{h}{2}$ entre eles.

2.0.2: Para cada $\mathbf{p}_i \in \mathcal{P}_h(t + \Delta t)$, computar as normais \mathbf{N}_i por meio da técnica descrita no **Passo A** do método RAMLS (Seção 3.3.1).

2.1: Para cada raio $r \in R_h$

2.1.1: Encontrar o subconjunto de esferas $\mathcal{W} \subset \mathcal{B}$ que satisfazem $\mathbf{b}_i \cap r \neq \emptyset$

2.1.2: Para cada $\mathbf{b}_i \in \mathcal{W}$,

2.1.2.1: Computar os dois pontos de interseção \mathbf{p}_{in} e \mathbf{p}_{out} entre r e \mathbf{b}_i . Avaliar F em \mathbf{p}_{in} e \mathbf{p}_{out} , respectivamente denominados $F(\mathbf{p}_{in})$ e $F(\mathbf{p}_{out})$ utilizando o **Passo B** do método RAMLS (Seção 3.3.1).

2.1.2.2: Se $F(\mathbf{p}_{in})F(\mathbf{p}_{out}) < 0$, encontrar $s^* \in]0, 1[$ tal que $F(\mathbf{p}_{in} + s^*(\mathbf{p}_{out} - \mathbf{p}_{in})) = 0$ pelo método regula-falsi.

2.1.2.3: Adicionar $\mathbf{p} = \mathbf{p}_{in} + s^*(\mathbf{p}_{out} - \mathbf{p}_{in})$ à $\mathcal{Q}_h(t + \Delta t)$ se este ponto já não foi adicionado anteriormente (por outra esfera \mathbf{b}_i).

Como mencionado no Passo 2.1.2.3, pode acontecer de várias esferas gerarem o mesmo ponto de interseção, como ilustra a Fig. 3.12. Este problema é solucionado com uma simples verificação: se já encontramos uma interseção do raio r e esta interseção está dentro da esfera que estamos avaliando, não procuramos por outra interseção dentro desta esfera. Este procedimento, aliado ao teste de sinal no Passo 2.1.2.2, assegura que somente uma interseção é considerada por esfera, para cada raio. Se dois lados opostos da superfície estão a uma distância menor que h , a interseção não é encontrada e aquele detalhe da superfície é automaticamente removido.

Foi observado em testes numéricos que a aproximação da superfície pelo método RAMLS deteriora se há pontos muito próximos entre si. Por isso o Passo 2.0.1 é empregado antes do processo de reconstrução da superfície: removemos os pontos próximos utilizando uma estratégia de *hashing*. É importante mencionar que todas as buscas pelos pontos “vizinhos” de um ponto (necessárias para as avaliações da função implícita) são efetuadas utilizando algoritmos eficientes baseados na construção de uma *kd-tree* [15] do conjunto de pontos. Além disso, o Passo 2.1 pode ser trivialmente paralelizado, já que cada raio é independente. Logo, o método proposto é escalável.

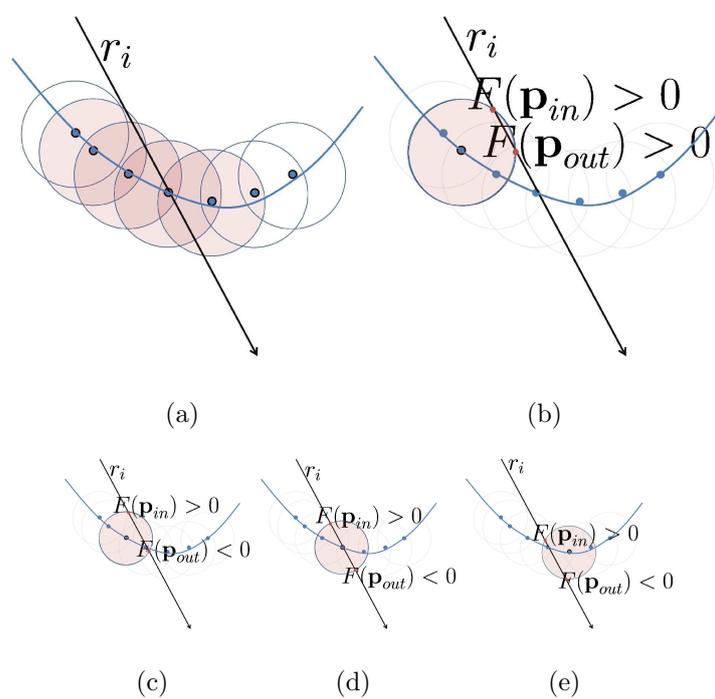


Figura 3.12: Interseções raio-esfera: três interseções produzem o mesmo zero na superfície. Neste caso, somente o primeiro ponto é adicionado ao novo conjunto de pontos.

CAPÍTULO 4

Resultados

Este Capítulo é dedicado à apresentação dos resultados obtidos utilizando o método de acompanhamento de fronteira proposto na Seção 3.4. Primeiramente analisaremos a convergência do método e posteriormente estudaremos o seu comportamento quando aplicado a problemas clássicos da literatura. A última parte deste capítulo (Seção 4.3) é dedicada ao estudo de casos de mudança de topologia da superfície. Mostraremos que o método proposto pode gerar superfícies espúrias em casos de coalescência ou ruptura da superfície. Desenvolvemos uma heurística com o intuito de eliminar estes problemas, apresentando resultados satisfatórios.

4.1 Convergência do Método de Acompanhamento de Fronteira Com Superfícies RAMLS

O erro do método proposto pode ser analisado em termos do erro do método RAMLS estudado na Seção 3.3.2. De fato,

$$\begin{aligned}
d(\mathcal{S}(t + \Delta t), \mathcal{S}_h(t + \Delta t)) &= d(\mathbf{S}_{\text{RAMLS}}(\mathcal{P}_h(t + \Delta t)), \varphi(\mathcal{S}(t))) \\
&= d(\mathbf{S}_{\text{RAMLS}}(\varphi(\mathcal{Q}_h(t))), \varphi(\mathcal{S}(t))) \\
&\leq d(\mathbf{S}_{\text{RAMLS}}(\varphi(\mathcal{Q}_h(t))), \varphi(\mathcal{S}_h(t))) + \\
&\quad + d(\varphi(\mathcal{S}_h(t)), \varphi(\mathcal{S}(t)))
\end{aligned}$$

onde por simplicidade omitimos os últimos dois argumentos de φ , os quais podem ser considerados t e $t + \Delta t$, respectivamente (i.e.; $\varphi(\cdot) = \varphi(\cdot, t, t + \Delta t)$). Observe que o primeiro termo é apenas o erro do método RAMLS da superfície $\varphi(\mathcal{S}_h(t))$, devido aos pontos de $\mathcal{Q}_h(t)$ pertencerem a $\mathcal{S}_h(t)$ (de fato, $\mathcal{Q}_h(t) = R_h \cap \mathcal{S}_h(t)$ e a operação de transporte φ é considerada como sendo exata. Se o passo de tempo é pequeno o suficiente, a continuidade de φ implica que $\varphi(\mathcal{Q}_h(t))$ é uma amostragem de $\varphi(\mathcal{S}_h(t))$ com espaçamento médio $\simeq h$, tal que

$$d(\mathbf{S}_{\text{RAMLS}}(\varphi(\mathcal{Q}_h(t))), \varphi(\mathcal{S}_h(t))) \leq C_1 h^r \quad (4.1)$$

sendo r a taxa de convergência estudada na Seção 3.3.2 Do mesmo modo,

$$d(\varphi(\mathcal{S}_h(t)), \varphi(\mathcal{S}(t))) \leq (1 + C_2 \Delta t) d(\mathcal{S}_h(t), \mathcal{S}(t)) \quad (4.2)$$

com C_2 dependendo do campo de velocidade. Assim temos

$$d(\mathcal{S}(t + \Delta t), \mathcal{S}_h(t + \Delta t)) \leq C_1 h^r + (1 + C_2 \Delta t) d(\mathcal{S}(t), \mathcal{S}_h(t))$$

e, portanto

$$d(\mathcal{S}(t), \mathcal{S}_h(t)) \leq C_3 \frac{h^r}{\Delta t} + e^{C_2 t} d(\mathcal{S}(0), \mathcal{S}_h(0)). \quad (4.3)$$

O erro geométrico, desta forma, possui $\mathcal{O}(h^r/\Delta t)$, típico dos métodos semi-lagrangianos. Se o erro da discretização temporal for levado em conta, o erro total possui $\mathcal{O}(h^r/\Delta t) + \mathcal{O}(\Delta t^s)$, sendo s a ordem do método de integração temporal utilizado na Eq. 3.22.

A Tabela 4.1 apresenta o erro do método proposto considerando diferentes valores para os parâmetros h e Δt . O problema considerado é uma revolução completa do elipsoide descrito na Seção 3.1 ao redor do eixo x_3 com velocidade angular igual a 2π , ou seja, a revolução será feita em uma unidade de tempo. Os resultados obtidos estão de acordo com a predição dada pela Equação 4.3.

Tabela 4.1: Erro geométrico obtido pelo método proposto na rotação rígida de um elipsoide ao redor do eixo x_3 (uma revolução).

	$h = \frac{12}{32}$	$h = \frac{12}{64}$	$h = \frac{12}{128}$
$\Delta t = \frac{1}{16}$	0,06261	0,00494	0,00065
$\Delta t = \frac{1}{64}$	0,14635	0,01007	0,00246
$\Delta t = \frac{1}{256}$	0,34056	0,02561	0,00756
$\Delta t = \frac{1}{1024}$	0,76286	0,07314	0,02352
$\Delta t = \frac{1}{4096}$	1,00345	0,16459	0,06323

4.2 Resultados numéricos

Esta Seção apresenta resultados numéricos obtidos pela método de Front-Tracking utilizando Superfícies RAMLS, considerando alguns casos de testes clássicos da literatura para este problema. Para todos os testes 2D, considere como domínio Ω o quadrado unitário e para os testes 3D, o cubo unitário.

4.2.1 Disco de Zalesak e Esfera de Zalesak

Este caso de teste de transporte de interface consiste na rotação rígida de uma circunferência (ou esfera, em 3D) a menos de uma seção retangular. No caso bidimensional, o disco é centrado em $(0,5, 0,75)$, com raio de $0,15$ e seção retangular de comprimento $0,05$ e altura $0,25$ (ver Fig. 4.2). O campo de velocidade é dado por $\mathbf{v}(x_1, x_2) = \frac{\pi}{314}(0,5 - x_2, x_1 - 0,5)$, de modo que uma revolução completa é realizada em 628 unidades de tempo. O passo de tempo utilizado é $\Delta t = 1$, e diversos valores de h são considerados. É importante observar que, para um dado valor de h , por exemplo $h = 1/512$, o método não calcula $512^2 = 262.144$ interseções raio-superfície, e sim as interseções de $2 \times 512 = 1.024$ raios com a superfície definida pelo método RAMLS. A precisão do método proposto para este caso de teste com $h = 1/512$, em que a superfície é representada por 700 pontos, em média, é similar à precisão do método PLS[18] (Figura 4.1 em um *grid* euleriano 100×100 , utilizando 12.864 pontos (estimativa obtida utilizando 16 pontos por célula, como em [23]).

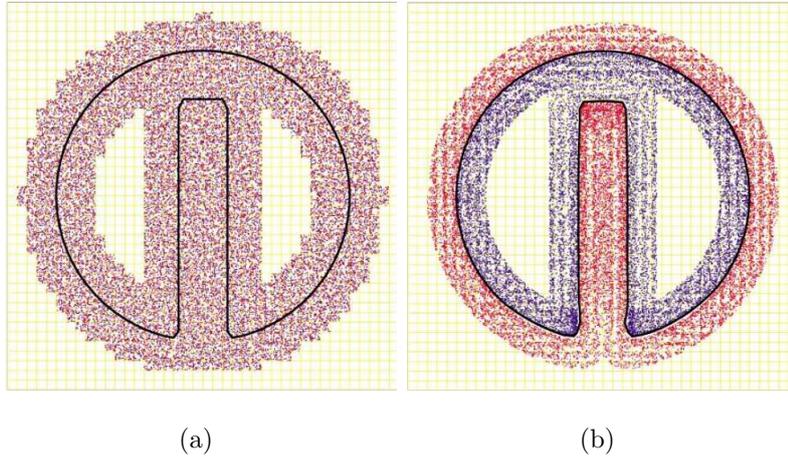


Figura 4.1: Resultados do método PLS [18]: (a) distribuição de pontos próxima à interface, (b) solução obtida (em preto) após uma revolução.

A Tabela 4.2.1 apresenta resultados quantitativos do mesmo caso de teste após uma

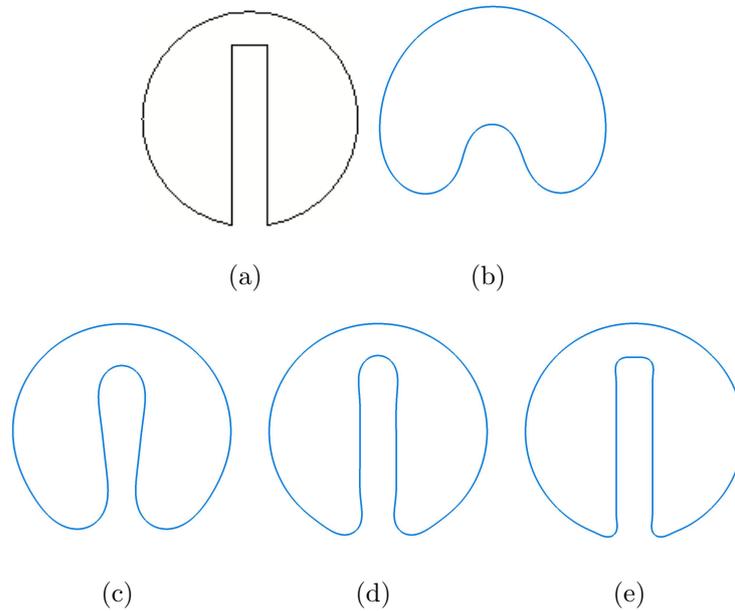


Figura 4.2: Rotação do disco de Zalesak: resultados obtidos após uma revolução. (a) Original, (b) $h=1/64$, (c) $h = 1/128$, (d) $h = 1/256$, (e) $h = 1/512$.

e duas revoluções. Em comparação com trabalhos anteriores, o método proposto foi capaz de preservar bem a área da interface. Por exemplo, considerando uma revolução do disco de Zalesak, o método proposto por Enright *et al.* [18] em um grid euleriano de 100×100 obteve uma perda de área de 0,31%, enquanto que no método proposto por Hieber & Koumoutsakos [23] esta perda é de 0,30%. É importante observar que a ordem de convergência do erro geométrico converge assintoticamente para 1, já que a falta de suavidade do disco de Zalesak faz com que a ordem máxima obtida por qualquer método seja 1.

No caso de teste tridimensional, a esfera de Zalesak consiste de uma esfera de raio 0,15 centrada inicialmente em $(0,5, 0,75, 0,75)$, com uma seção retangular de comprimento 0,10 e altura 0,20. O campo de velocidade é dado por $\mathbf{v}(x_1, x_2, x_3) = \frac{\pi}{314}(0,5 - x_2, x_1 - 0,5, 0)$. Novamente, o passo de tempo é de $\Delta t = 1$ e uma revolução é feita em 628 passos de tempo. A Fig. 4.3 mostra o formato da esfera de Zalesak para

Tabela 4.2: Disco de Zalesak: método RAMLS

	h	Area	Perda de área	Erro geométrico [†]	Ordem
	exato	0,05811580	-	-	-
Uma revolução	1/64	0,05880127	-0,17501%	0,03485	N/A
	1/128	0,05810201	0,02372%	0,01409	1,30
	1/256	0,05812257	-0,01166%	0,00628	1,16
	1/512	0,05811785	-0,00353%	0,00335	0,90
Duas revoluções	1/64	0,05859332	-0,82167%	0,04470	N/A
	1/128	0,05813988	-0,04144%	0,02052	1,12
	1/256	0,05812066	-0,00836%	0,00833	1,29
	1/512	0,05812647	-0,01835%	0,00415	1,00

os instantes $t = 0, 79, 157, 236, 314, 393, 471, 550$ e 628 . O parâmetro h foi fixado em $h = 1/256$, levando a uma média de 19.000 pontos representando a interface. Durante a simulação, as arestas e características afiadas da superfície são suavizados, porém a forma e o volume são bem preservados.

4.2.2 Escoamento de um Vórtice

Embora o caso de teste considerado na Seção 4.2.1 seja importante para verificar difusão da interface, ele não testa a capacidade do método em preservar massa, geometria e topologia em simulações cuja deformação da interface é crítica. Para este fim, o método proposto foi utilizado para simular a interface em formato de vórtice [11] dada pelo campo de velocidade

$$\mathbf{v}(x_1, x_2) = 2 \left(-\sin^2(\pi x_1) \sin(\pi x_2), -\sin^2(\pi x_2) \sin(\pi x_1) \right).$$

A geometria da interface inicialmente consiste de uma circunferência centrada em

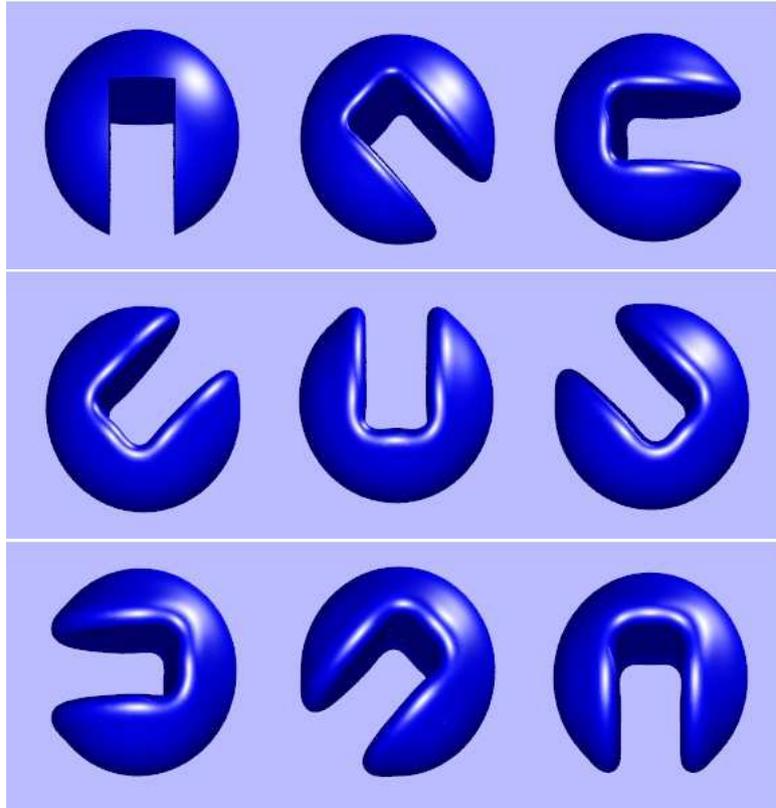


Figura 4.3: Rotação da esfera de Zalesak com $h = 1/256$ nos instantes $t = 0, 79, 157, 236, 314, 393, 471, 550$ e 628 (da esquerda à direita e de cima para baixo).

$(0, 5, 0, 75)$ com raio igual a $0, 15$. Consideramos $\Delta t = 0, 01$ e h igual a $1/256$ e $1/512$, o que leva a uma quantidade inicial de pontos de 233 e 469 , respectivamente. Este número de pontos é bem menor em comparação com outros métodos, como o *Lagrangian Particle Level Set* [23] (4.000 pontos) e o PLS (59.000 pontos). A Fig. 4.4 ilustra os instantes $t = 1, t = 3$ e $t = 5$ para ambos valores de h . Pode-se notar que os resultados obtidos com o método proposto (em azul) aproximam bem a solução exata (em preto), mesmo quando a interface é muito fina e podem ser comparados com os melhores resultados obtidos com outros métodos.

Para efeito de comparação, na Figura 4.5 apresentamos os resultados obtidos pelo método LPLS [23] em(a) e do método PLS [18] em (b). Ambos resultados foram si-

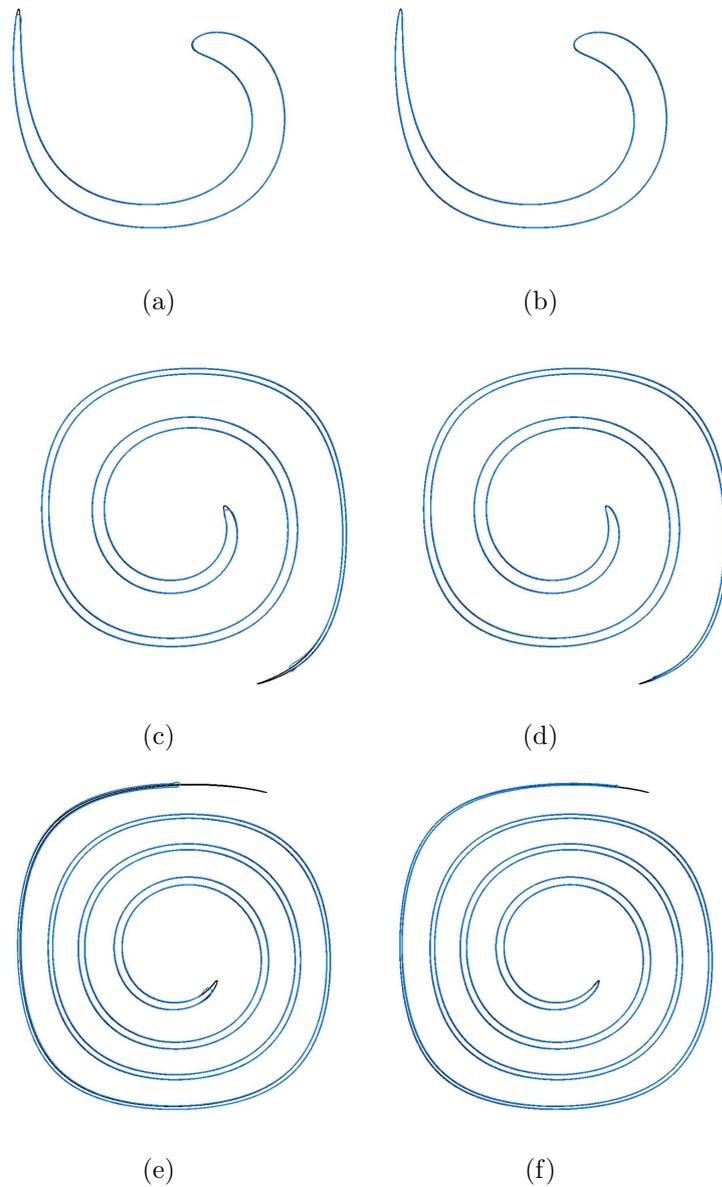


Figura 4.4: Solução exata (preto) e solução pelo método RANLS (azul): $h = 1/256$ e $h = 1/512$ com $\Delta t = 0,01$ onde (a) e (b) correspondem ao instante $t = 1$, (c) e (d) correspondem ao instante $t = 3$, e (e) e (f) correspondem ao instante $t = 5$.

mulados em uma grade computacional com espaçamento $1/128$. O número inicial de partículas para cada simulação foi de 1.160 para o LPLS e 15.040 para o PLS. Além

disso na simulação com o método LPLS, no tempo $t = 3$, o número de partículas é de 55.914.

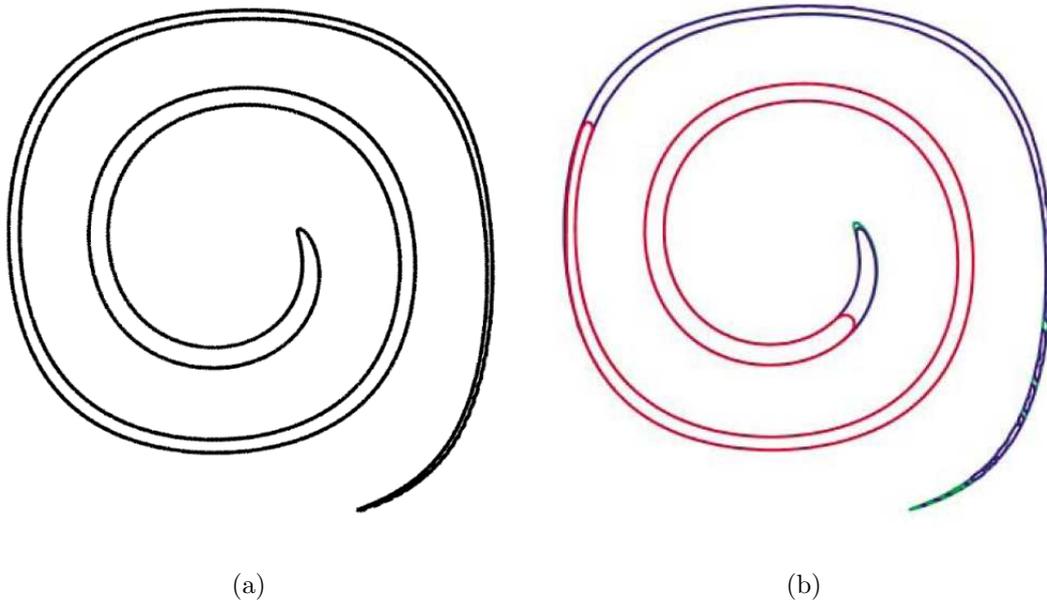


Figura 4.5: Simulações do escoamento em torno de um vórtice (grade computacional com espaçamento $1/128$ e $t = 3$): (a) método LPSL [23], (b) em vermelho temos a solução pelo método LS, em verde, a solução exata e em azul a solução pelo método PLS [18].

Com o objetivo de melhor analisar a precisão do método proposto, efetuamos a comparação com a biblioteca FronTier, que implementa o método desenvolvido por Du et al. [17] e está disponível no site <http://frontier.ams.sunysb.edu/download>. Esta biblioteca permite ao usuário ajustar o passo no tempo, que foi fixado em $\Delta t = 0,002$. Como a interface é reconstruída a cada cinco passos, a frequência de re-geração da superfície é a mesma utilizada no nosso método (i.e., re-gera-se os pontos a cada 0,01 unidades de tempo). A Figura 4.6 mostra a comparação feita das interfaces obtidas com a biblioteca FronTier e com o nosso método (utilizando $h = 1/256$). Podemos observar que o método proposto é mais preciso que o método de Du et al. principalmente nas

pontas do vórtice.

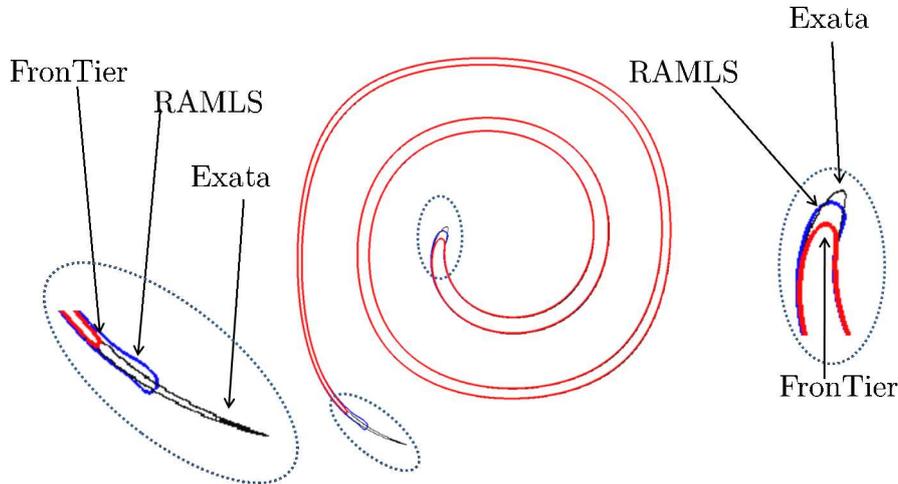


Figura 4.6: Comparação do método proposto (curva azul) com o método de acompanhamento de fronteira desenvolvido por Du et al [17] (curva vermelha) para o caso do escoamento de um vórtice no tempo $t = 3$. Para ambos métodos foi fixado $h = 1/256$ e os pontos são re-gerados a cada 0,01 unidades de tempo. A curva preta ilustra a solução exata (obtida por transporte lagrangeano de 100.000 pontos) para efeitos comparativos.

Também executamos este caso de teste com o campo de velocidade modulado no tempo pela expressão $\cos(\pi t/T)$, de forma que, para $t = T$, a interface volta a ser um círculo. Consideramos $T = 8$ e 800 passos de tempo. A Fig. 4.7 ilustra a forma da interface em $t = T$ gerada pelo método proposto para diferentes valores de h (1/64, 1/128, 1/256 e 1/512). A Tabela 4.3 apresenta quantitativamente os resultados deste caso de teste em termos da perda de área e do erro geométrico. Para $h = 1/256$, por exemplo, a perda de área de 0,36 % obtida é similar à perda obtida com métodos anteriores [23, 18].

A ordem de convergência obtida para o erro geométrico é próxima de 2, e não 3 como nossa estimativa de erro (ver 4.3) predizia. Este fato não é tão surpreendente, devido ao fim do vórtice não ser “bem” aproximado em nenhuma das simulações, mesmo

a correspondente a $h = 1/512$. A Tabela 3 ainda traz o erro L_1 , definido como:

$$\frac{1}{2\pi R} \int_0^{2\pi} \left| \frac{f(\theta)^2 - R^2}{2} \right| d\theta, \quad (4.4)$$

onde $f(\theta)$ denota a distância do centro do círculo à curva RAMLS e R é o raio do círculo. Esta integral foi calculada numericamente por meio da poligonalização da curva RAMLS em um grid extremamente fino (5000×5000). A ordem de convergência obtida com a norma L_1 é próxima de 2,5 e o erro é uma ordem de magnitude menor que o erro geométrico.

Tabela 4.3: Escoamento de um vórtice modulado no tempo para $T = 8$. Resultados obtidos pelo método RAMLS.

h	Área	Perda de área	Erro geométrico	Ordem	Erro L_1	Ordem
exato	0,07069	-	-	-		
64	0,09724	-37.57 %	0,10649	N/A	0,04147	N/A
128	0,073988	-4,67 %	0,03247	1,71	0,00730	2,50
256	0,070943	-0,36 %	0,01044	1,64	0,00107	2,77
512	0,070676	0,013%	0,00236	2,14	0,00018	2,59

Para fins comparativos, a Fig. 4.8 apresenta os resultados finais do teste do escoamento de um vórtice modulado no tempo para os métodos LPLS [23] e PLS [18].

Ainda para efeitos de comparação, na Fig. 4.9 comparamos os resultados do método de Du et al. [17] (obtidos pela biblioteca FrontTier) com o nosso método, utilizando $h = 1/256$ para ambos métodos. Novamente pode-se observar que o método proposto neste trabalho é significativamente mais preciso (o que não é surpreendente, já que o método para reconstrução da superfície utilizado é uma ordem mais preciso que o método de reconstrução utilizado por Du et al.). Em compensação, o código da biblioteca FrontTier foi três vezes mais rápido que o nosso código. Esta diferença de tempo computacional

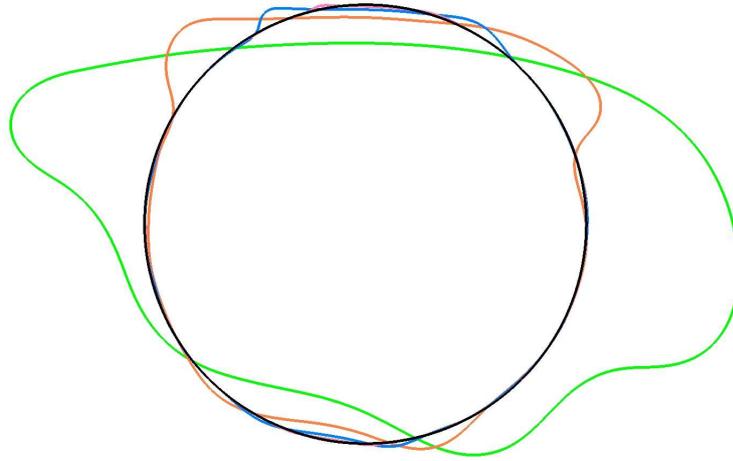


Figura 4.7: Círculo exato (em preto), e obtidos numericamente pelo método proposto com $h = 1/64$ (em verde), $h = 1/128$ (em laranja), $h = 1/256$ (em azul), $h = 1/512$ (em pink).

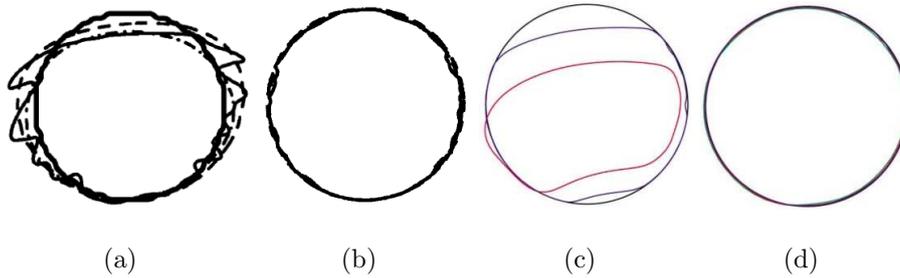


Figura 4.8: (a) e (b): resultados do método LPLS [23] para grades computacionais com espaçamento de $1/128$ e $1/1000$, respectivamente; (c): soluções por LS [36] para grades computacionais com espaçamento de $1/64$ (desapareceu), $1/128$ (vermelho) e $1/256$ (azul), e a solução exata (preto); (d): resultados para o método PLS [18], em que o número inicial de partículas corresponde a 3376, 15040 e 59072 para as grades computacionais com espaçamentos $1/64$ (verde), $1/128$ (vermelho) e $1/256$ (azul), e a solução exata (preto).

é devida à falta de uma estrutura de conectividade e à necessidade de se encontrar a vizinhança de um ponto a cada avaliação da função. Por outro lado, nosso método evita

as verificações de consistência topológica da malha (que são relativamente simples em duas dimensões, mas se tornam um pesadelo computacional em três dimensões, como veremos no próximo exemplo).

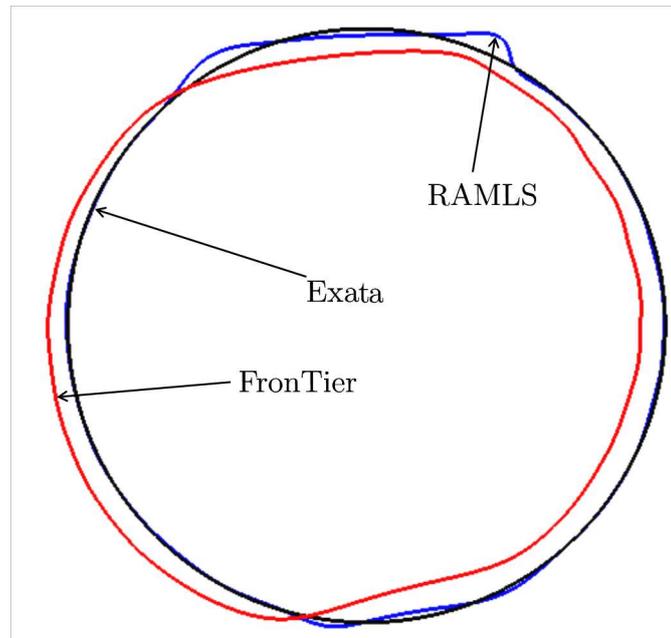


Figura 4.9: Resultados numericamente obtidos em $t = T = 8$ para o escoamento de um vórtice modulado no tempo. O parâmetro h é fixado em $h = 1/256$ e os pontos são re-gerados a cada 0,01 unidades de tempo. A curva azul é obtida pelo método RAMLS, enquanto que a curva vermelha é obtida pelo método de Du et al. [17]. A curva preta ilustra a solução exata (obtida por transporta lagrangeano de 100.000 pontos) para efeitos comparativos.

A Fig. 4.10 mostra um gráfico do número de pontos em função do tempo de simulação. Pode-se inferir que uma boa precisão é obtida com apenas 1.500 pontos ($h = 1/256$) onde a deformação é máxima. É importante observar que a simetria e a continuidade do gráfico mostram que o número de pontos é proporcional ao comprimento da interface. Por exemplo, para $h = 1/128$, o conjunto de pontos inicial contém 118 pontos e o final 128. Já no método *Lagrangian Particle Level Set* [23], o conjunto

de pontos inicial contém 1.200 pontos e o final 2.800 (Fig. 4.11).

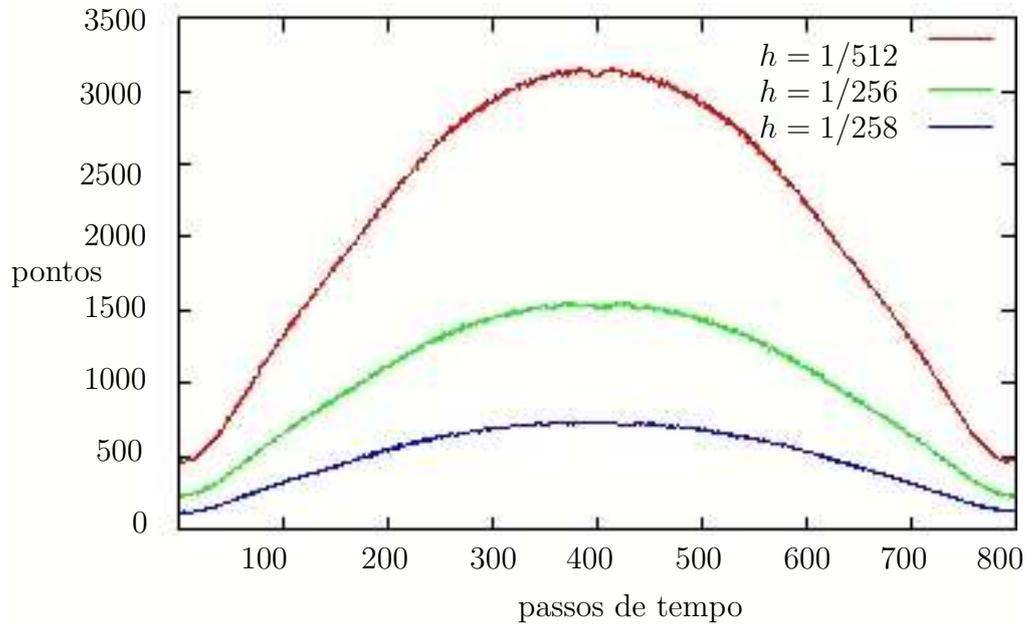


Figura 4.10: Evolução do número de pontos em $Q_h(t)$ ao decorrer da simulação modulada no tempo do escoamento de um vórtice.

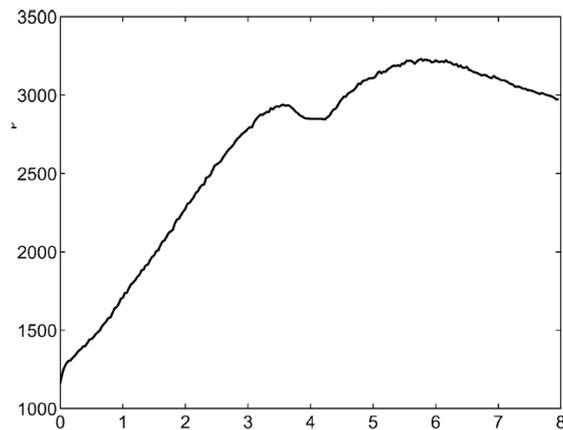


Figura 4.11: Evolução do número de partículas geradas pelo método LPSL [23] no decorrer da simulação modulada no tempo do escoamento de um vórtice para uma grade computacional com espaçamento 1/128.

4.2.3 Deformação Tridimensional

Como último caso de teste, consideramos uma deformação de uma superfície tridimensional. A superfície inicial $\mathcal{S}(0)$ é uma esfera de raio 0,15 centrada em $(0,35, 0,35, 0,35)$.

O campo de velocidades é dado por

$$\mathbf{v} = \begin{pmatrix} 2 \sin^2(\pi x_1) \sin(2\pi x_2) \sin(2\pi x_3) \\ -\sin(2\pi x_1) \sin^2(\pi x_2) \sin(2\pi x_3) \\ -\sin(2\pi x_1) \sin(2\pi x_2) \sin^2(\pi x_3) \end{pmatrix}.$$

No instante $t = T/2 = 1$, o campo de velocidade é multiplicado por -1 para testar a capacidade do método de preservar a geometria, massa e topologia da esfera. A Fig. 4.12 ilustra os resultados da simulação utilizando $h = 1/512$, $\Delta t = 0,0064$ e 628 passos de tempo.

Hieber and Koumoutsakos [23] apresentam resultados semelhantes (ver Fig. 4.18). O método proposto, todavia, parece preservar melhor a topologia em situações críticas de deformação. A geometria final é bem próxima à esfera inicial, porém uma pequena silhueta pode ser observada (ver Fig. 4.13).

Para facilitar a comparação com os resultados obtidos por Enright et al [18] e Du et al [17], também realizamos o mesmo teste para o mesmo campo de velocidade modulado no tempo, ou seja:

$$\mathbf{v} = \begin{pmatrix} 2 \sin^2(\pi x_1) \sin(2\pi x_2) \sin(2\pi x_3) \\ -\sin(2\pi x_1) \sin^2(\pi x_2) \sin(2\pi x_3) \\ -\sin(2\pi x_1) \sin(2\pi x_2) \sin^2(\pi x_3) \end{pmatrix} \cos\left(\frac{\pi t}{T}\right).$$

Após $t = T = 3$ unidades de tempo a interface exata volta a ser uma esfera. Consideramos $h = 1/512$, $\Delta t = 0,02$ e 150 passos de tempo. A Figura 4.14 ilustra os resultados obtidos. Novamente uma pequena silhueta pode ser observada na superfície

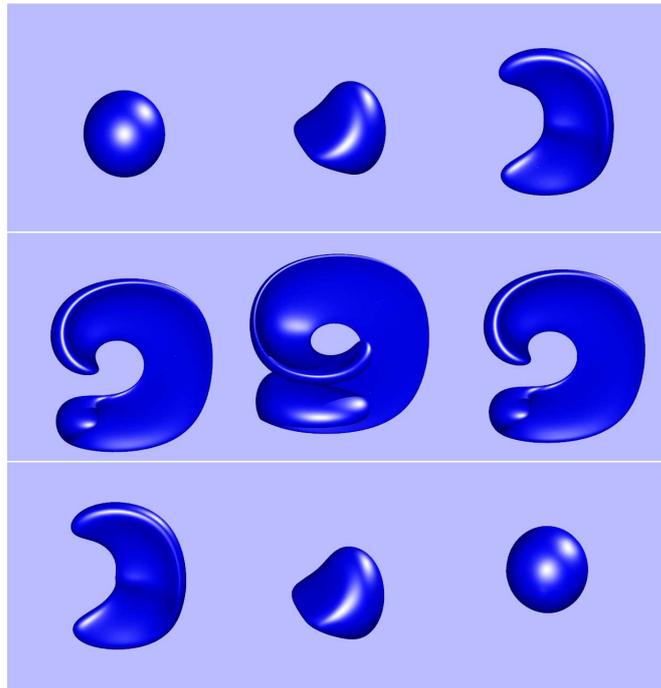


Figura 4.12: Deformação tridimensional ($h = 1/512$) obtida numericamente pelo método proposto em diferentes instantes de tempo (espaçados igualmente, com o campo de velocidades sendo invertido em $t = 2$). O número de pontos que representam a superfície são (da esquerda para direita e de cima para baixo), respectivamente, 65.000, 69.314, 109.728, 187.571, 277.021, 193.238, 114.898, 72.774, 68.031.

final, como mostra a Figura 4.15.

Se realizamos a mesma simulação com o parâmetro $h = 1/256$, ainda obtemos resultados razoáveis, como mostra a Figura 4.16. Embora a forma da superfície quando a distorção é máxima ($t = 1,5$) seja bem reproduzida, artefatos numéricos no fim da simulação são bem maiores que anteriormente, quando o valor de h era igual a $1/512$. O método de Du et al. [17] (obtidos pela biblioteca FrontTier), para a mesma simulação (ver Figura 4.17), recupera perfeitamente a esfera inicial. Esta é uma consequência da falta de informação de conectividade do nosso método. Quando a superfície “estica”, duas “folhas” da superfície começam a interferir uma no resultado da outra. É claro que

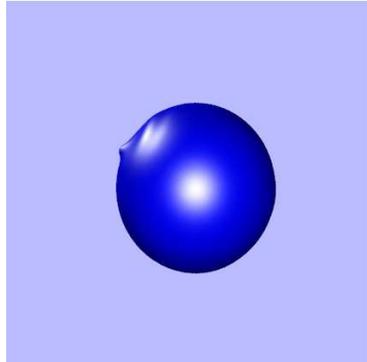


Figura 4.13: A superfície do canto inferior direito da Fig. 4.12 vista de um outro ângulo. Uma pequena silhueta pode ser observada na superfície.

isto não acontece com o método de Du et al. [17], já que este mantém uma estrutura de conectividade em que lados opostos da superfície estão topologicamente desconectados. Por outro lado, a atualização e manutenção de conectividade da malha que seja topologicamente consistente são procedimentos computacionalmente bem caros. Neste exemplo, o tempo de CPU e a memória necessária pelo código FronTier foram 4 e 29 vezes maior, respectivamente, que o tempo de CPU e memória necessários pelo nosso código.

Por fim, a Figura 4.18 ilustra resultados obtidos pelos métodos (grade computacional com espaçamento de $1/100$): (a) LS [36] (b) PLS [18] e (c) LPLS [23].

4.3 Testes com Mudanças Topológicas

Para avaliar a capacidade do método proposto para lidar com mudanças na topologia da interface, moveremos a superfície de um “anel aberto” (ver Figura 4.19(a)) de acordo com o seguinte campo de velocidades: $\mathbf{v}(\mathbf{x}, t) = \mathbf{n}(\mathbf{x})$ para todo $\mathbf{x} \in \mathcal{S}(t)$, para cada t . O raio do anel é 0,3 e sua largura inicial é 0,02. O parâmetro h é fixo e igual a $h = 1/64$ e o passo de tempo é $\Delta t = 0,01$.

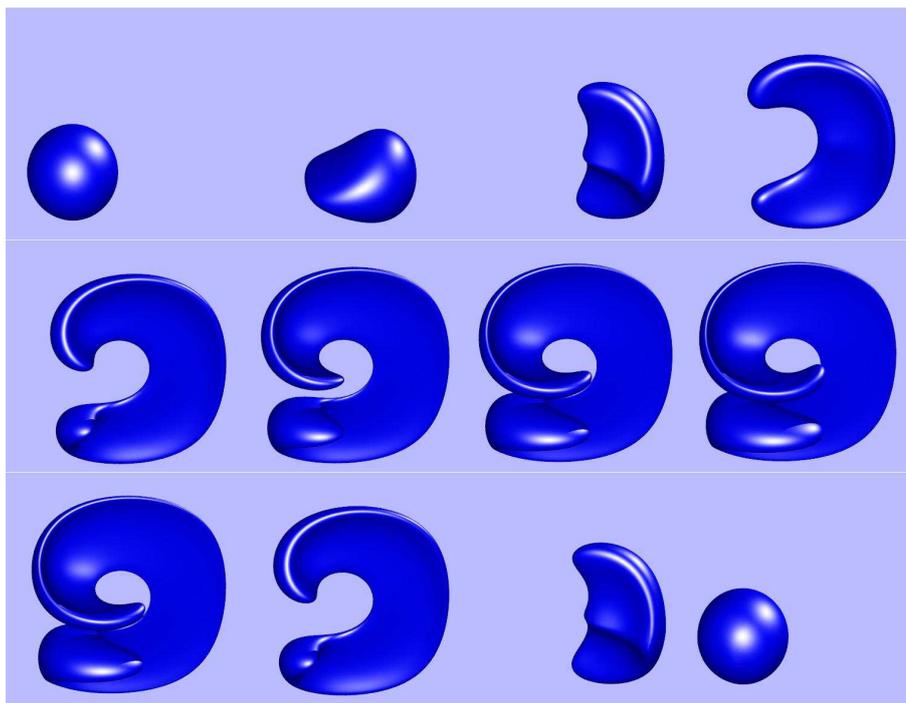


Figura 4.14: Deformação tridimensional ($h = 1/512$) modulada no tempo. Da esquerda para a direita e de cima para baixo: obtidas nos instantes $t = 0, 10\Delta t, 20\Delta t, 30\Delta t, 40\Delta t, 50\Delta t, 60\Delta t, 70\Delta t, 90\Delta t, 110\Delta t, 130\Delta t, 150\Delta t$, onde $\Delta t = 3/150$. O número de pontos que representam a superfície são: 65.000, 73.790, 88.263, 136.299, 183.720, 228.456, 263.896, 280.663, 263.853, 183.932, 88.545 e 69.295

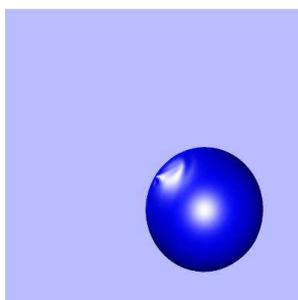


Figura 4.15: A superfície do canto inferior direito da Fig. 4.14 vista de um outro ângulo. Uma pequena silhueta pode ser observada na superfície.

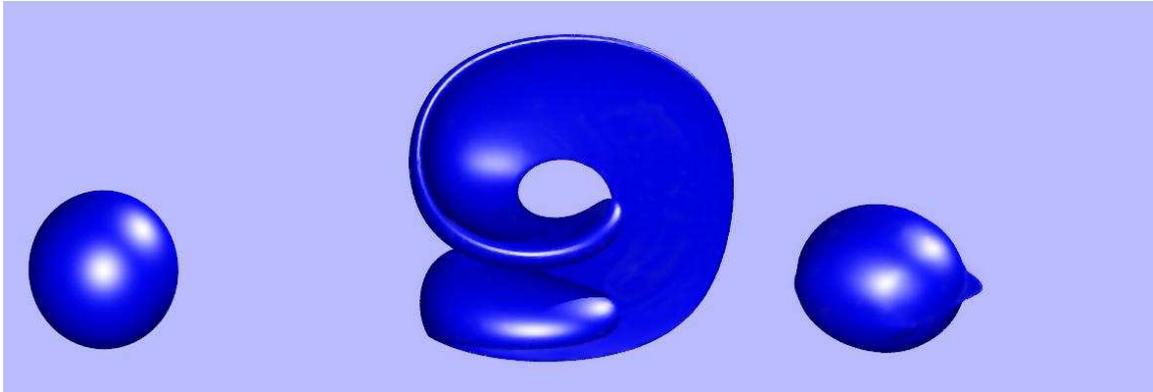


Figura 4.16: Deformação tridimensional de uma esfera ($h = 1/256$). Resultados do método RAMLS nos instantes $t = 0$, $t = 1,5$ e $t = 3$.

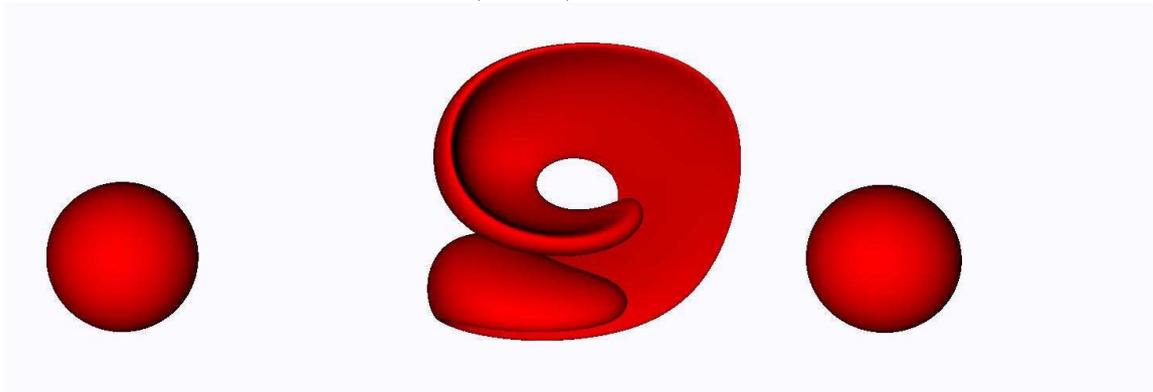
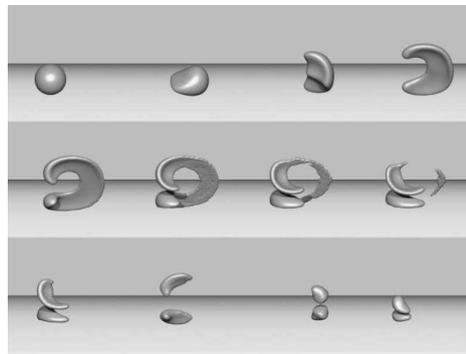


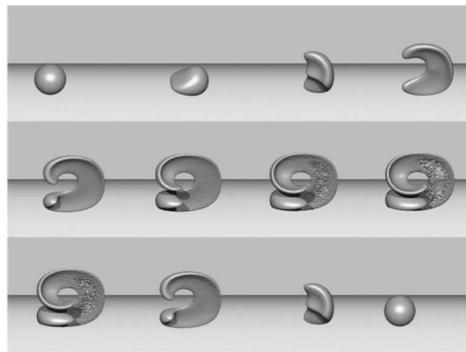
Figura 4.17: Deformação tridimensional de uma esfera ($h = 1/256$). Resultados do método proposto por Du et al. [17] nos instantes $t = 0$, $t = 1,5$ e $t = 3$.

A Figura 4.19 mostra os resultados da simulação descrita acima. A mudança de topologia da superfície (coalescência) ocorre sem problemas, como acontece nos métodos eulerianos. No caso de métodos lagrangeanos, o encontro das duas “folhas” da superfície teria levado a um colapso da simulação (ou seria necessário um procedimento para conectar as duas malhas, como o procedimento de reconstrução utilizado no código FronTier [17]).

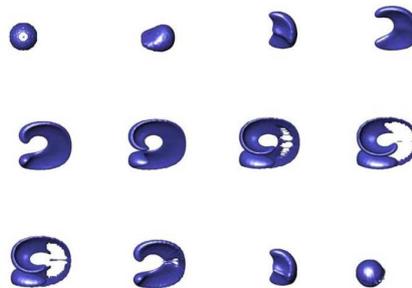
Entretanto, se o mesmo teste é simulado com um passo de tempo duas vezes maior,



(a)



(b)



(c)

Figura 4.18: Solução do caso de deformação tridimensional pelos métodos (grade computacional com espaçamento de $1/100$): (a) LS [36](b) PLS [18] e (c) LPLS [23].

artefatos numéricos podem ser vistos na solução obtida. A título de ilustração, considere o mesmo caso de teste do “anel aberto” em uma versão bidimensional. A Figura 4.20

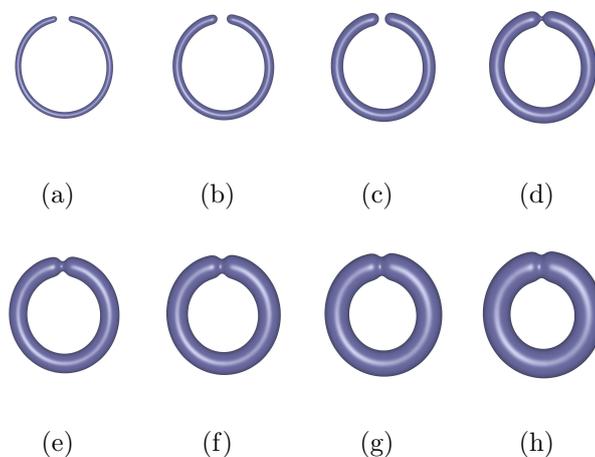


Figura 4.19: Resultados do teste tridimensional do anel aberto nos instantes: (a) 0, (b) 0,2, (c) 0,4, (d) 0,6, (e) 0,8, (f) 1,0, (g) 1,2 e (h) 1,4. A mudança de topologia ocorre no instante $t = 0,6$

ilustra o resultado do teste.

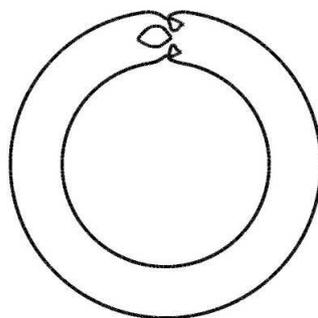


Figura 4.20: Resultado do teste bidimensional do “anel aberto”: artefatos numéricos podem ser observados na solução obtida ($t = 1,0$).

Com o objetivo de solucionar este problema, desenvolvemos uma heurística para efetuar mudanças topológicas de uma maneira simples e funcional na maioria dos casos (embora possa falhar quando diversas superfícies se aproximam simultaneamente). A

heurística é baseada na constatação de que os outros conjuntos de níveis da função definida pela Equação 2.17 podem ser utilizados para prever mudanças topológicas na superfície definida pelo conjunto de nível zero da mesma função. Como podemos observar na Figura 4.21, uma coalescência ocorre no conjunto de nível ϵ , enquanto uma ruptura da superfície ocorre no conjunto de nível $-\epsilon$.

A heurística proposta pode ser dividida em três passos, a saber: **(i)** identificação de “folhas” opostas próximas umas em relação às outras; **(ii)** classificação da operação topológica (coalescência ou ruptura da superfície); **(iii)** execução da operação topológica (detalhada a seguir). Com o intuito de facilitar a notação, denotaremos por \mathcal{P} o conjunto $\mathcal{P}_h(t)$ para valores fixos de t e h . A seguir, detalharemos os passos da heurística proposta:

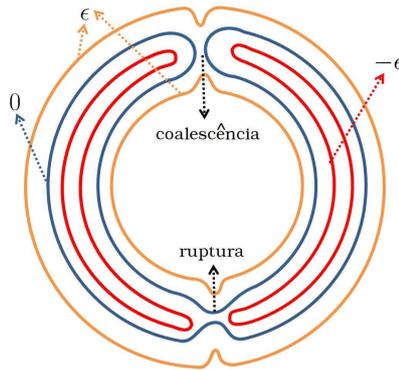


Figura 4.21: Ilustração dos conjuntos de nível utilizados para executar as operações de coalescência e ruptura da superfície: uma ruptura da superfície no conjunto de nível $-\epsilon$ (em vermelho) (em comparação com o conjunto de nível zero (em azul)) pode ser observada, enquanto que no nível ϵ (em laranja) uma coalescência pode ser observada.

Passo (i) O objetivo do passo de identificação é detectar se uma mudança topológica está prestes a ocorrer. A detecção utilizada é baseada no fato de que as normais em pontos vizinhos não mudam abruptamente. Em outras palavras, considere um ponto $\mathbf{p}_i \in \mathcal{P}$ e seja V_i o conjunto de pontos em \mathcal{P} tal que $\|\mathbf{p}_i - \mathbf{p}_j\| < \Delta$. Se há um

ponto em V_i para o qual o ângulo entre as normais \mathbf{N}_i e \mathbf{N}_j é maior do que 150° (este valor foi escolhido após diversos testes numéricos) e $\|\mathbf{p}_i - \mathbf{p}_j\| < 2\epsilon$, então afirmamos que duas “folhas” da superfície estão suficientemente próximas para que uma mudança topológica possa acontecer na vizinhança de \mathbf{p}_i . O parâmetro ϵ ($0 < \epsilon < \frac{\Delta}{2}$) permite controlar a distância máxima entre duas “folhas” da superfície, ou seja, se duas “folhas” estiverem a menos de 2ϵ de distância uma da outra, executamos uma mudança topológica.

Passo (ii) Para cada ponto \mathbf{p}_i identificado no **Passo (i)** como um ponto localizado em uma vizinhança na qual pode ocorrer uma mudança topológica da superfície, decidimos se o ponto pertence a uma região de coalescência ou ruptura. Esta decisão também é feita baseada na orientação das normais. Especificamente, se existe um ponto $\mathbf{p}_j \in V_i$ tal que o vetor $(\mathbf{p}_j - \mathbf{p}_i)$ e a normal \mathbf{N}_i possuem a mesma orientação, isto é, que o produto interno $(\mathbf{p}_j - \mathbf{p}_i) \cdot \mathbf{N}_i$ é positivo, então \mathbf{p}_i é rotulado como um ponto de coalescência. Caso contrário, \mathbf{p}_i é rotulado como um ponto de ruptura (estamos assumindo que as normais estão orientadas consistentemente para fora da superfície).

Passo (iii) Para efetuar mudanças topológicas detectadas nos passos anteriores, o conjunto \mathcal{P} é substituído por um novo conjunto de pontos obtido por operadores de união e intersecção de pontos de três diferentes níveis de pontos: (i) \mathcal{P} (o conjunto de pontos atual), (ii) \mathcal{P}_c (o conjunto de pontos amostrados do conjunto de nível ϵ e transladados de ϵ na direção oposta ao gradiente da função 2.17) e (iii) \mathcal{P}_b (o conjunto de pontos amostrados do conjunto de nível $-\epsilon$ e transladados de ϵ na direção do gradiente da função 2.17). Sejam C e B os conjuntos definidos por:

$$C(X) = \{\mathbf{p} \in X \mid \exists \mathbf{p}_i \text{ rotulado como ponto de coalescência} \in \mathcal{P}, \|\mathbf{p} - \mathbf{p}_i\| < \Delta\}.$$

$$B(X) = \{\mathbf{p} \in X \mid \exists \mathbf{p}_i \text{ rotulado como ponto de ruptura} \in \mathcal{P}, \|\mathbf{p} - \mathbf{p}_i\| < \Delta\}.$$

Finalmente, o novo conjunto de pontos $\mathcal{P}_{\mathcal{N}}$, de acordo com as mudanças to-

pológicas escolhidas, pode ser obtido pela seguinte expressão:

$$\mathcal{P}_{\mathcal{N}} = (\mathcal{P} \cup \mathcal{P}_c \cup \mathcal{P}_b) - C(\mathcal{P}) \cup C(\mathcal{P}_b) - B(\mathcal{P}) \cup B(\mathcal{P}_c). \quad (4.5)$$

Note que, se no **Passo (ii)** os pontos foram rotulados apenas como pontos de coalescência, o conjunto de pontos \mathcal{P}_b pode ser desconsiderado e a Equação 4.5 pode ser redefinida como $\mathcal{P}_{\mathcal{N}} = (\mathcal{P} \cup \mathcal{P}_c) - C(\mathcal{P})$. No caso de haver apenas pontos de ruptura, a Equação 4.5 pode ser redefinida como $\mathcal{P}_{\mathcal{N}} = (\mathcal{P} \cup \mathcal{P}_b) - B(\mathcal{P})$.

Há dois aspectos da heurística descrita acima que precisam ser analisados: o motivo da escolha da curva de nível ϵ para se efetuar a mudança topológica e a translação dos pontos. Ambos são baseados no fato de que a função implícita gerada pelo método RAMLS aproxima a função distância (ver Seção 2.4). De fato, quando duas “folhas” da superfície estão próximas, a função implícita gerada pelo método RAMLS subestima a função distância (como pode ser observado na Figura 4.22). Usando esta propriedade podemos utilizar as curvas de nível ϵ e $-\epsilon$ para obter uma superfície com a topologia desejada. Infelizmente não conseguimos provar matematicamente que as curvas de nível ϵ e $-\epsilon$ têm a topologia desejada, e deixaremos isto para um trabalho futuro. Não obstante, os resultados numéricos obtidos mostram que, se a distância entre duas folhas é menor que 2ϵ , a curva de nível *epsilon* possui topologia diferente da curva de nível zero.

Por fim, como $\|\nabla f\| \approx 1$ (pelo menos em regiões em que duas “folhas” da superfície não estejam próximas), um ponto $\tilde{\mathbf{p}}_i$ localizado no conjunto de nível ϵ pode ser transladado para o conjunto de nível zero simplesmente fazendo-se $\mathbf{p}_i = \tilde{\mathbf{p}}_i - s\nabla f(\tilde{\mathbf{p}}_i)$, onde s é $-\epsilon$ ou ϵ , dependendo se é o caso de uma ruptura ou de uma coalescência. A Figura 4.23 ilustra o processo de translação dos pontos para o caso de uma coalescência da superfície.

A Figura 4.24 ilustra a simulação bidimensional do “anel aberto”, descrita anteriormente, porém agora utilizando a heurística proposta. A mudança de topologia ocorre

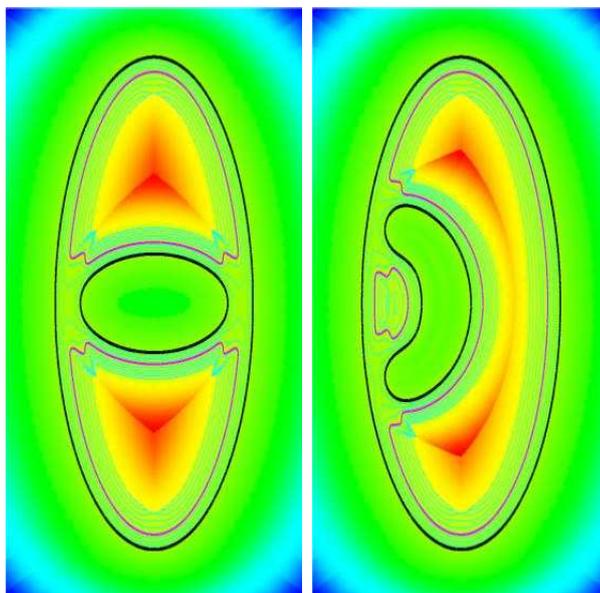


Figura 4.22: A função implícita gerada pelo método RAMLS subestima a função distância quando duas “folhas” da superfície estão próximas. Em preto, temos o seu conjunto de nível zero; em rosa, temos o conjunto de nível $\vartheta/2$, onde ϑ é a distância entre as duas “folhas” da superfície; em azul temos diversas outras curvas de nível.

satisfatoriamente, sem a produção de superfícies espúrias na região de coalescência.

Com o intuito de validar a heurística proposta para casos em que os processos de coalescência e ruptura da superfície ocorrem simultaneamente, amostramos um conjunto de pontos de uma superfície similar à ilustrada pela Figura 4.21. A Figura 4.25 ilustra o resultado da aplicação da heurística proposta acima para este conjunto de pontos. Pode-se notar que os processos de coalescência e ruptura da superfície são obtidos de forma satisfatória.

A Figura 4.26 ilustra uma simulação em que a mudança de topologia ocorre de acordo com a escolha do parâmetro ϵ , ou seja, ao especificar o valor de ϵ , podemos forçar a mudança de topologia da superfície. Na Figura 4.26(a), escolhendo $\epsilon = 0,5h$ a coalescência entre os braços dos bonecos Gingerbread não ocorre (neste exemplo, fixou-

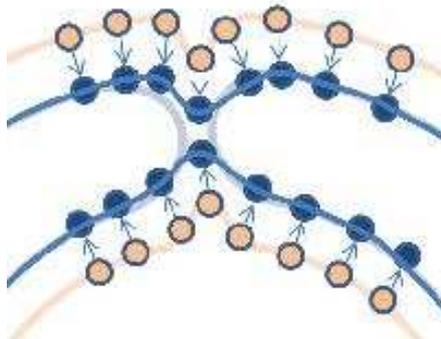


Figura 4.23: Ilustração da operação de coalescência: primeiramente, os pontos em laranja são obtidos pela interseção entre o grid R_h e o conjunto de nível ϵ . Em seguida os pontos em laranja são transportados na direção do gradiente da função, definindo os pontos em azul. Finalmente, os pontos em azul são utilizados para executar corretamente a mudança topológica, definindo uma superfície com diferente topologia (curva azul-marinho).

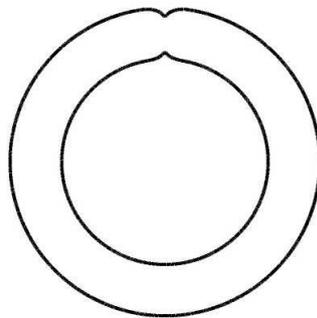


Figura 4.24: Resultado do teste bidimensional do “anel aberto” utilizando a heurística proposta.

se $h = 1/512$ e a simulação acontece dentro de um cubo unitário). Por outro lado, quanto escolhemos $\epsilon = 1,5h$, os braços dos bonecos se unem, produzindo uma única superfície conectada.

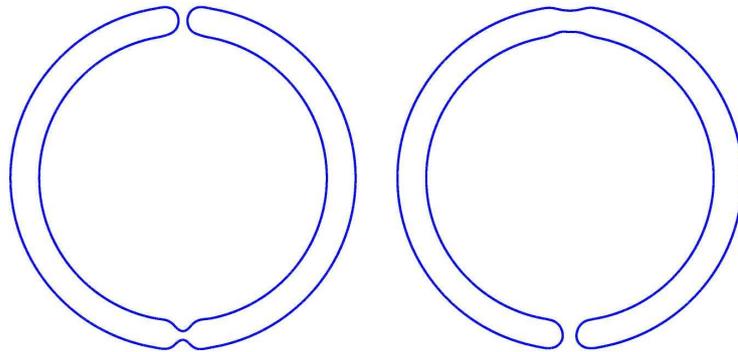


Figura 4.25: Simulação na qual coalescência e ruptura ocorrem no mesmo passo de tempo.

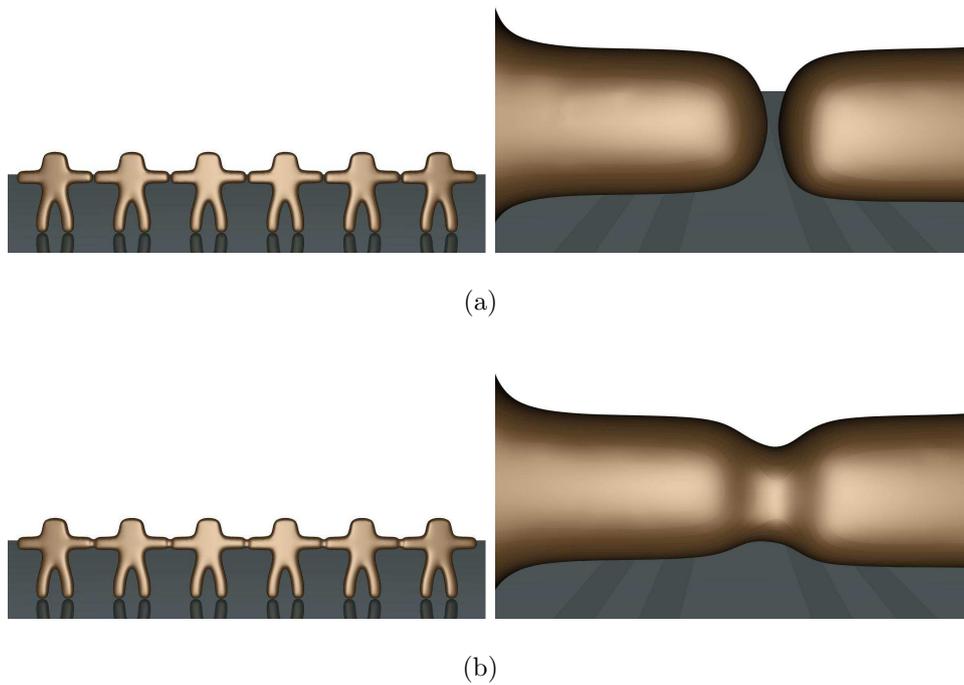


Figura 4.26: O parâmetro ϵ permite controlar a distância máxima entre duas superfícies opostas. (a) $\epsilon = 0,5h$: sem mudanças topológicas; (b) $\epsilon = 1,5h$, coalescência entre os braços dos bonecos.

A Figura 4.27 ilustra a capacidade da heurística no caso de ruptura de uma superfície tridimensional. Neste exemplo, a cabeça do bebê foi ajustada para que ocorresse uma ruptura da superfície. A Figura 4.27(a) ilustra a configuração inicial da superfície e as Figuras 4.27(b) e 4.27(c) ilustram a superfície após 4 passos de tempo, utilizando $\Delta t = 0,001$, $h = 1/512$ e o campo de velocidades dado por $\mathbf{v}(\mathbf{x}, t) = \mathbf{n}(\mathbf{x})$ para todo $\mathbf{x} \in \mathcal{S}(t)$, para cada t . Note que, se escolhermos $\epsilon = 0,4h$, a topologia da superfície não se altera. Em contrapartida, se escolhermos $\epsilon = 1,0h$, a superfície se rompe.

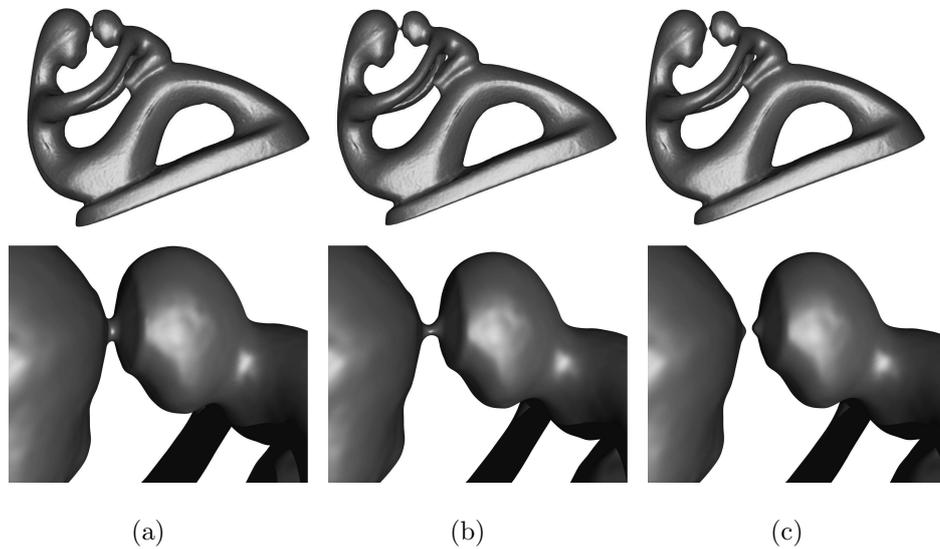


Figura 4.27: Simulação em que ocorre uma ruptura da superfície: a imagem à esquerda mostra a configuração inicial. Após quatro passos de tempo, a imagem do meio ilustra a ausência de mudanças topológicas (considerando $\epsilon = 0,4h$), enquanto que na imagem à direita pode-se observar uma ruptura da superfície (considerando $\epsilon = 1,0h$).

Como último exemplo, dois modelos tridimensionais do “Hommer Simpson” são movidos um em direção ao outro, provocando a união das duas superfícies de maneira satisfatória. A Figura 4.28 ilustra alguns passos da simulação.

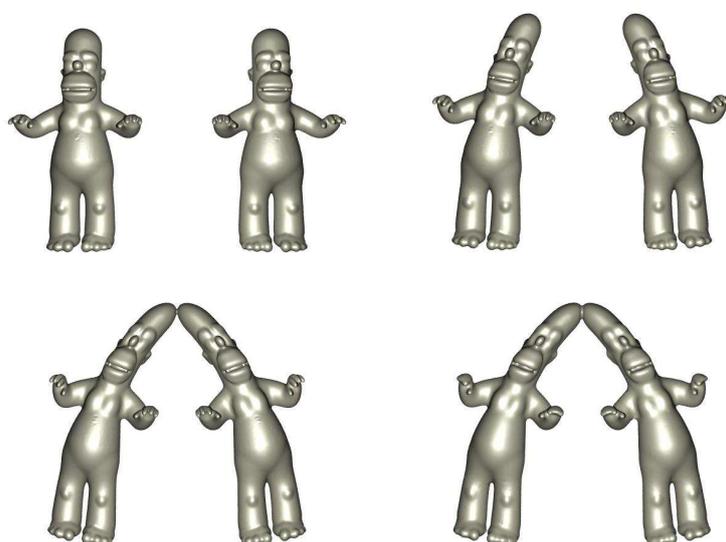


Figura 4.28: Coalescência entre dois modelos tridimensionais do “Hommer Simpson”.

CAPÍTULO 5

A Biblioteca RAMLS

Nesse Capítulo apresentamos a biblioteca elaborada no decorrer do projeto para representação de Superfícies RAMLS, bem como o *software* desenvolvido no decorrer do projeto que implementa o método de acompanhamento de fronteira proposto neste trabalho. Também ilustraremos outros testes realizados com o auxílio da biblioteca RAMLS.

5.1 Projeto da Biblioteca RAMLS

A biblioteca RAMLS é uma biblioteca destinada ao auxílio no desenvolvimento de aplicações nas quais há a necessidade de se obter uma representação implícita de uma nuvem de pontos pelo método RAMLS. A representação da superfície é feita por meio da técnica descrita na Seção 3.3.1. Desenvolvida na linguagem C++, ela utiliza a licença de software público GPL [1]. A Fig. 5.1 ilustra o diagrama de classes da biblioteca.

A classe `RAMLS_Surface` possui métodos para avaliar o valor da Função dada pela Eq. 2.17, aproximar esferas algébricas pelas soluções dos problemas da minimização das distâncias algébricas e da minimização dos desvios das normais, além de métodos para atualizar as normais (Passo 2.0.2 do método RAMLS proposto), encontrar a interseção da superfície com um grid euleriano (Passo 2.1 do método RAMLS proposto) e remover pontos duplicados (Passo 2.0.1 do método RAMLS Proposto).

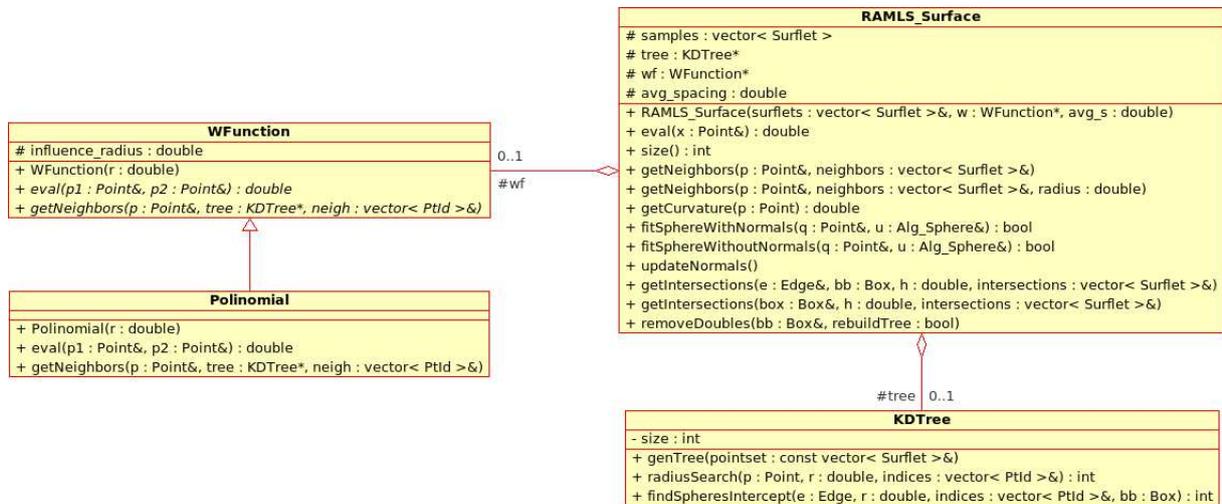


Figura 5.1: Diagrama de classes da biblioteca RAMLS.

A classe descrita no parágrafo acima também agrega duas outras classes:

- a classe abstrata `WFunction`, que representa a função núcleo (Eq. 2.1). Esta classe contém métodos abstratos puros para avaliar a função e decidir sobre a vizinhança de um dado ponto. Como classe concreta, temos a classe `Polinomial`, que implementa a função dada pela Eq. 2.4, a qual foi utilizada neste projeto.
- a classe `KDTree`, que representa a estrutura de dados para busca eficiente da vizinhança de um determinado ponto.

Para renderizar as imagens das superfícies RAMLS, a biblioteca desenvolvida foi

incorporada ao código-fonte do traçador-de-raios **PovRay** (*Persistence of Vision*) [2].

5.2 *Software* para Acompanhamento de Fronteira com Superfícies RAMLS

O *software* para acompanhamento de fronteira desenvolvido utiliza a biblioteca descrita na Seção 5.1 para implementar o método descrito na Seção 3.4. Foi implementada uma interface para o usuário carregar e salvar arquivos, mudar parâmetros e visualizar o estado atual da simulação. Este *software* foi desenvolvido na linguagem C++ e disponibilizado com a licença pública GPL [1]. A Fig. 5.2 ilustra o diagrama de classes do *software*.

A classe `SimControl` é a classe de controle dos eventos recebidos pela classe `Interface`, que por sua vez tem a função de renderizar a cena para o usuário. A classe `SimControl` possui métodos para salvar e carregar arquivos de simulação, além de métodos para dar passos no tempo, de acordo com o método descrito na Seção 3.4. Ela possui uma relação por composição com a classe `ODE_Solver`, que é responsável pelo transporte lagrangeano dos pontos segundo um campo de velocidades prescrito, representado pela classe `VField`.

A classe `RAMLS_DrawableSurface` é derivada por especialização das classes `RAMLS_Surface` e `SceneObject`. A classe `SceneObject` é uma classe abstrata que representa os objetos que compõem a cena que será renderizada pela classe `Interface`, segundo a posição do observador dada pela classe `Camera`.

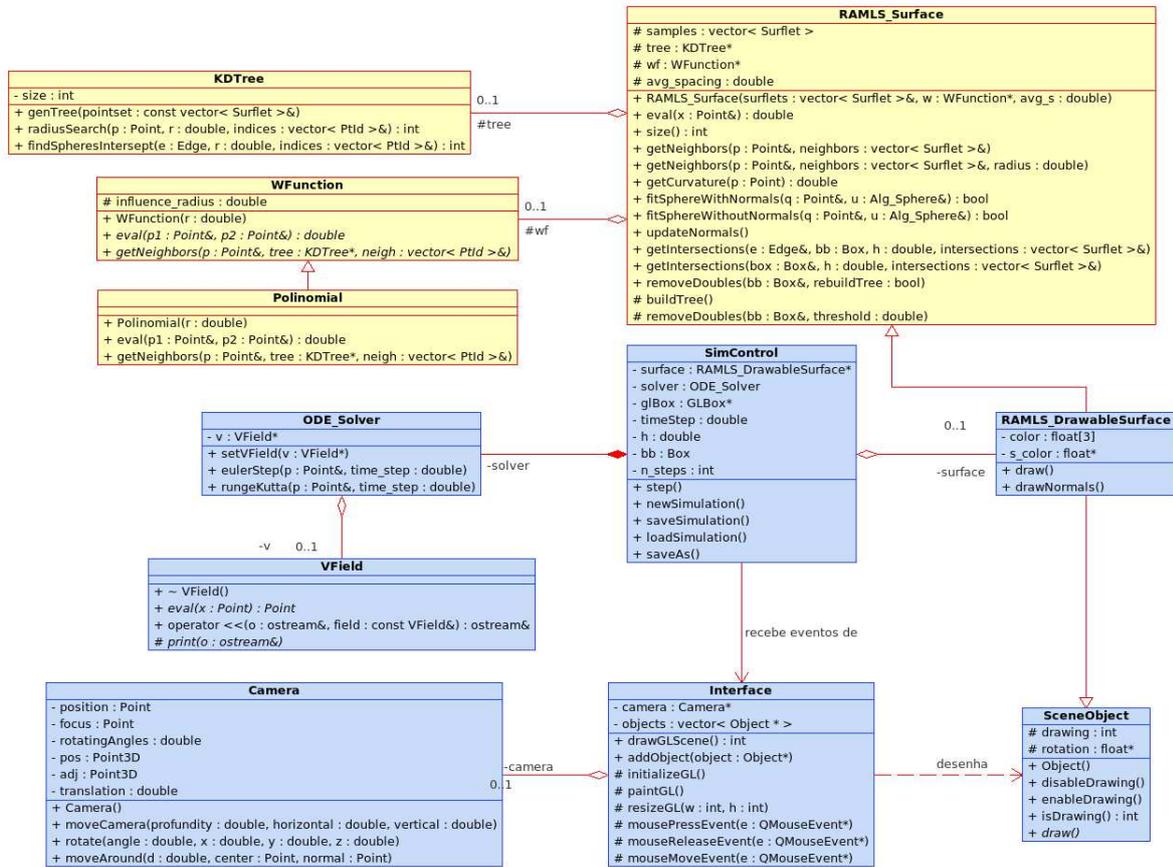


Figura 5.2: Diagrama de classes do *Software* para Acompanhamento da Frente com Superfícies RAMLS. Classes com bordo vermelho pertencem à biblioteca RAMLS, enquanto que as classes com bordo em azul são específicas do *software* desenvolvido.

5.3 Exemplos de Simulações Realizadas com o *Software*

Com o intuito de verificar e validar o *software* desenvolvido, e também produzir resultados adicionais para estudar a precisão do método proposto de um modo qualitativo,

simulamos o efeito de um campo de velocidades sobre superfícies dadas por conjuntos de pontos bem conhecidos na área de reconstrução de superfície em Computação Gráfica. A Figura 5.3 ilustra uma deformação do *Stanford Bunny*. Pode-se observar que as características da superfície são suavizadas no decorrer da simulação. A topologia da superfície, entretanto, é bem preservada.

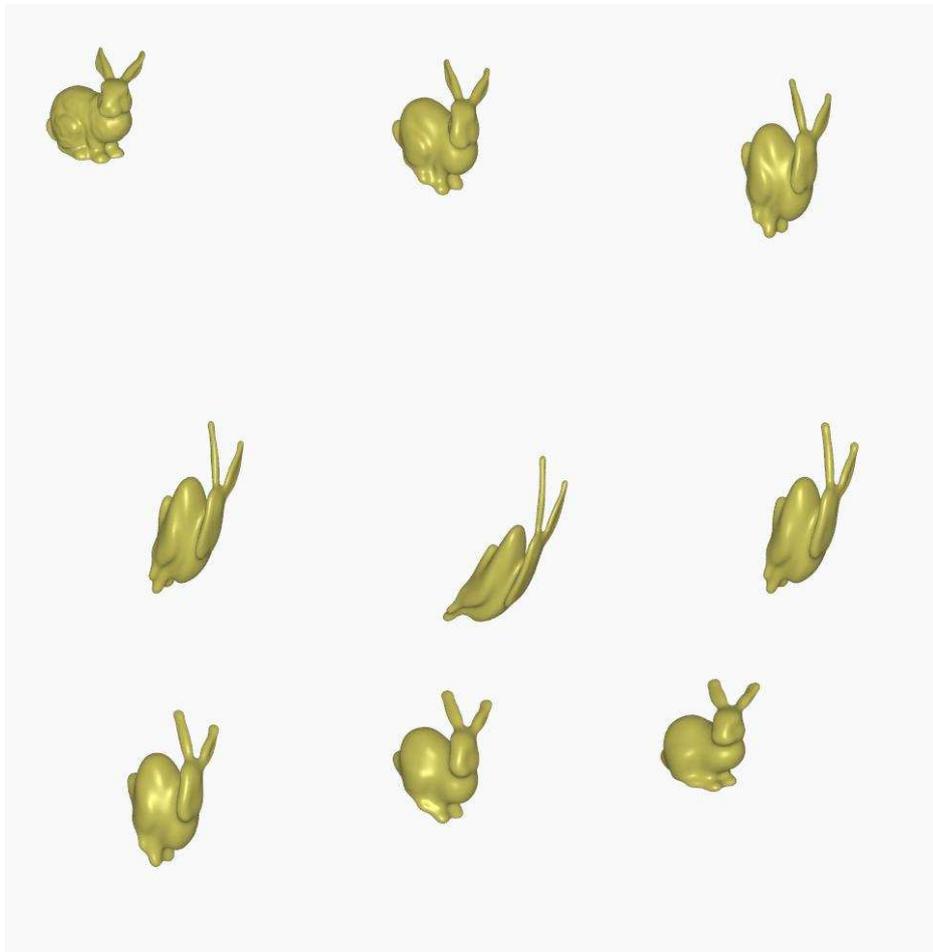


Figura 5.3: Deformação do Stanford Bunny ($h = 1/512$).

A Figura 5.4 ilustra uma deformação do *Hommer Simpson*. A superfície não é regenerada a cada passo neste caso, apenas atualiza-se as normais de acordo com o passo 2.0.2 do algoritmo descrito na Seção 3.4.2. Pode-se observar que as características da

superfície são mantidas, já que não há a re-geração dos pontos, apenas o transporte lagrangeano dos mesmos.

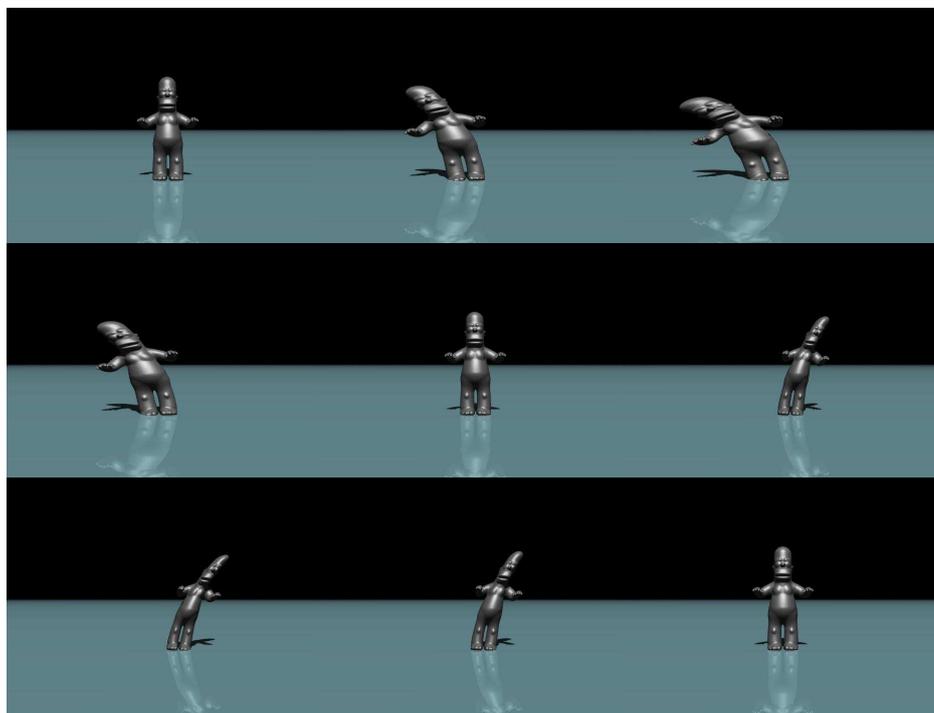


Figura 5.4: Deformação do Hommer Simpson.

CAPÍTULO 6

Conclusão

Técnicas de representação da interface entre dois fluidos foram amplamente estudadas e desenvolvidas por pesquisadores nas últimas décadas. Este fato demonstra a relevância científica da área de pesquisa e a dificuldade para encontrar uma solução ótima para o problema.

Nesse trabalho desenvolvemos um método híbrido de acompanhamento de fronteiras livre de malhas que utiliza aproximações algébricas de alta ordem, o qual denominamos de método RAMLS. Com base nos diversos testes documentados no Capítulo 4, podemos concluir que o método proposto é competitivo em termos de precisão e robustez com outros métodos propostos anteriormente na literatura, geralmente requerendo a utilização de menos partículas para obter precisão equivalente. O nosso método também é comparável em termos de precisão ao método de acompanhamento de fronteira introduzido por Du et al. [17], o qual mantém uma estrutura de conectividade entre as partículas marcadoras.

Também fizemos a análise, projeto e implementação de uma biblioteca que implementa o método proposto, com o intuito de facilitar o acesso a pesquisadores que se interessem pela utilização do método em simulações bifásicas.

Desenvolvemos ainda uma heurística para tratar mudanças de topologia de uma maneira simples e funcional. Não obstante, esta heurística precisa ser melhor estudada para que seja suficientemente robusta para uma simulação de fluidos, e seu aperfeiçoamento consiste em um dos desafios futuros. Outros trabalhos futuros incluem a elaboração de mecanismos no qual a re-geração de pontos da superfície está acoplada com a malha em que se realiza a simulação numérica do escoamento e a inclusão de anisotropia no método.

Referências Bibliográficas

- [1] Licença gpl - <http://www.gnu.org/copyleft/gpl.html> acessado em abril de 2008.
- [2] Persistence of vision - <http://www.porvay.org>. acessado em abril de 2008.
- [3] A. Adamson and M. Alexa. Approximating and intersecting surfaces from points. In *SGP '03: Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 230–239, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.
- [4] A. Adamson and M. Alexa. Ray tracing point set surfaces. In *Proceedings of the Shape Modeling International*, pages 272–279, 2003.
- [5] M. Alexa and A. Adamson. On normals and projection operators for surfaces defined by point sets. In *Proceedings of Eurographics Symposium on Point-based Graphics*,, pages 149–156. IEEE Computer Society, 2004.

- [6] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C.T. Silva. Point set surfaces. In *VIS '01: Proceedings of the conference on Visualization '01*, pages 21–28. IEEE Computer Society, 2001.
- [7] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C.T. Silva. Computing and rendering point set surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 9(1):3–15, 2003.
- [8] N. Amenta, M. Bern, and M. Kamvysselis. A new voronoi-based surface reconstruction algorithm. *Computer Graphics*, 32(Annual Conference Series):415–421, 1998.
- [9] N. Amenta and Y.J. Kil. Defining point-set surfaces. *ACM Trans. Graph.*, 23(3):264–270, 2004.
- [10] N. Amenta and Y.J. Kil. The domain of a point-set surface. In *Eurographics Workshop on Point-based Graphics*, pages 139–147, 2004.
- [11] J.B. Bell, P. Colella, and H.M. Glaz. A 2nd-order projection method for the incompressible navier-stokes equations. *Journal of Computational Physics*, 2(85):257–283, 1989.
- [12] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):349–359, 1999.
- [13] P. Carrica, R. Wilson, and F. Stern. Unsteady RANS simulation for the forward speed diffraction problem. *Computers and Fluids*, 35:545–570, 2006.
- [14] P. Carrica, R. Wilson, and F. Stern. An unsteady single-phase level set method for viscous free surface flows. *International Journal for Numerical Methods in Fluids*, 53:229–256, 2007.

- [15] M. deBerg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, 1997.
- [16] T. K. Dey and J. Sun. An adaptive mls surface for reconstruction with guarantees. In *Symposium on Geometry Processing*, pages 43–52, 2005.
- [17] J. Du, B. Fix, J. Glimm, X. Jia, X. Li, Y. Li, and L. Wu. A simple package for front tracking. *J. Comput. Phys.*, 213(2):613–628, 2006.
- [18] D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell. A Hybrid Particle Level Set Method for Improved Interface Capturing. *Journal of Computational Physics*, 183:83–116, 2002.
- [19] O. Gloth, D. Hänel, L. Tran, and R. Vilsmeier. A front tracking method on unstructured grids. *Computers & Fluids*, 32:547–570, 2003.
- [20] GSL Gsl gnu scientific library <http://www.gnu.org/software/gsl/>. novembro/2007.
- [21] M. Gopi, S. Krishnan, and C. T. Silva. Surface reconstruction based on lower dimensional localized delaunay triangulation. In M. Gross and F. R. A. Hopgood, editors, *Computer Graphics Forum (Eurographics 2000)*, volume 19(3), 2000.
- [22] G. Guennebaud and M. Gross. Algebraic point set surfaces. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, page 23, New York, NY, USA, 2007. ACM.
- [23] S. Hieber and P. Koumoutsakos. A Lagrangian particle level set method. *Journal of Computational Physics*, 210:342–367, 2005.
- [24] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. *Computer Graphics*, 26(2):71–78, 1992.
- [25] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. *Symposium on Geometry Processing*, pages 61–70, 2006.

- [26] R. Kolluri. Provably good moving least squares. In *SODA '05: Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1008–1017, Philadelphia, PA, USA, 2005. Society for Industrial and Applied Mathematics.
- [27] B. Lafaurie, C. Nardone, R. Scardovelli, S. Zaleski, and G. Zanetti. Modelling Merging and Fragmentation in Multiphase Flows with SURFER. *Journal of Computational Physics*, 113:134–147, 1994.
- [28] D. Lakehal, M. Meier, and M. Fulgosi. Interface tracking towards the direct simulation of heat and mass transfer in multiphase flows. *International Journal of Heat and Fluid Flow*, 23:242–257, 2002.
- [29] D. Levin. The approximation power of moving least-squares. *Mathematics of Computation*, 67(224):1517–1531, 1998.
- [30] D. Levin. Mesh-independent surface interpolation. In Brunnett, Hamann, and Mueller, editors, *Geometric Modeling for Scientific Visualization*, pages 37–49. Springer-Verlag, 2003.
- [31] F. Losasso, R. Fedkiw, and S. Osher. Spatially adaptive techniques for level set methods and incompressible flow. *Computers and Fluids*, 35:995–1010, 2006.
- [32] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*, volume 153 of *Applied Mathematical Sciences*. Springer, 2003.
- [33] S.J. Osher and R.P. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer, 2002.
- [34] V. Pratt. Direct least-squares fitting of algebraic surfaces. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 145–152, New York, NY, USA, 1987. ACM Press.

- [35] R. Scardovelli and S. Zaleski. Direct numerical simulation of free-surface and interfacial flow. *Annu. Rev. Fluid Mech.*, 31:567–603, 1999.
- [36] J. A. Sethian. Fast Marching Methods and Level Set Methods for Propagation of Interfaces. In *von Karman Institute Lecture Series, Computational Fluid Mechanics*. 1998.
- [37] J. A. Sethian and P. Smereka. Level Set Methods for Fluids Interfaces. *Annual Rev. Fluid Mech.*, 35:341–372, 2003.
- [38] C. Shen, J. F. O’Brien, and J. R. Shewchuk. Interpolating and approximating implicit surfaces from polygon soup. In *SIGGRAPH ’04: ACM SIGGRAPH 2004 Papers*, pages 896–904, New York, NY, USA, 2004. ACM Press.
- [39] F.S. Sousa, N. Mangiavacchi, L.G. Nonato, A. Castelo, M. Tomé, V. Ferreira, J. Cuminato, and S. McKee. A front-tracking method for simulation of 3D multifluid flows with free surfaces. *Journal of Computational Physics*, 198:469–499, 2004.
- [40] M. Sussman. A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles. *Journal of Computational Physics*, 187:110–136, 2003.
- [41] M. Sussman and E. G. Puckett. A coupled level set and volume-of-fluid method for computing 3d and axisymmetric incompressible two-phase flows. *Journal of Computational Physics*, 162:301–337, 2000.
- [42] D. Torres and J. Brackbill. The point-set method: Front-tracking without connectivity. *Journal of Computational Physics*, 165:620–644, 2000.
- [43] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, and Y. J. Jan. A Front-Tracking Method for the Computations of Multiphase Flow. *Journal of Computational Physics*, 169:708–759, 2001.

- [44] S. Unverdi and G. Tryggvason. A front-tracking method for viscous, incompressible, multi-fluid flows. *Journal of Computational Physics*, 100:25–37, 1992.
- [45] I. Wald and H.P. Seidel. Interactive ray tracing of point-based models. In *Eurographics Symposium on Point-Based Graphics*, pages 1–8, 2005.
- [46] R. Wilson, P. Carrica, and F. Stern. Simulation of ship breaking bow waves and induced vortices and scars. *International Journal for Numerical Methods in Fluids*, in press.
- [47] M. Zwicker, M. Pauly, O. Knoll, and M. Gross. Pointshop 3d: an interactive system for point-based surface editing. *ACM Trans. Graph.*, 21(3):322–329, 2002.

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)