

**Afonso Paiva Neto**

**Uma abordagem lagrangeana para simulação  
de escoamentos de fluidos viscoplásticos e  
multifásicos**

**Tese de Doutorado**

Tese apresentada ao Programa de Pós-graduação em  
Matemática Aplicada do Departamento de Matemática da PUC-  
Rio como requisito parcial para obtenção do título de Doutor em  
Matemática Aplicada

Orientador: Prof. Geovan Tavares dos Santos

Rio de Janeiro  
Agosto de 2007

# **Livros Grátis**

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

Milhares de livros grátis para download.



**Afonso Paiva Neto**

**Uma abordagem lagrangeana para simulação  
de escoamentos de fluidos viscoplásticos e  
multifásicos**

Tese apresentada ao Programa de Pós-graduação em Matemática Aplicada do Departamento de Matemática do Centro Técnico Científico da PUC-Rio como requisito parcial para obtenção do título de Doutor em Matemática Aplicada. Aprovada pela Comissão Examinadora abaixo assinada.

**Prof. Geovan Tavares dos Santos**

Orientador

Departamento de Matemática — PUC-Rio

**Prof. Antônio Castelo Filho**

Instituto de Ciências Matemáticas e da Computação — USP

**Prof. Carlos Eduardo Aguiar**

Instituto de Física — UFRJ

**Prof. Luiz Henrique de Figueiredo**

Instituto Nacional de Matemática Pura e Aplicada — IMPA

**Prof. Paulo Roberto de Souza Mendes**

Departamento de Engenharia Mecânica — PUC-Rio

**Prof. Thomas Lewiner**

Departamento de Matemática — PUC-Rio

**Prof. Hélio Côrtes Vieira Lopes**

Departamento de Matemática — PUC-Rio

**Prof. José Eugenio Leal**

Coordenador Setorial do Centro Técnico Científico — PUC-Rio

Rio de Janeiro, 6 de Agosto de 2007

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

### **Afonso Paiva Neto**

Bacharel em Matemática pela Universidade Federal de Uberlândia (2000). Tornou-se Mestre em Matemática pelo Instituto Nacional de Matemática Pura e Aplicada (2003), onde a sua dissertação foi sobre métodos robustos de visualização de atratores estranhos. Durante o doutorado foi membro do laboratório Matmídia do Departamento de Matemática da PUC-Rio, desenvolvendo projetos científicos patrocinados pela Petrobras. Tem experiência na área de Matemática Aplicada, com ênfase em Computação Gráfica e atualmente suas principais áreas de interesse são: modelagem computacional de fenômenos físicos, modelagem geométrica, dinâmica dos fluidos computacional e métodos numéricos auto-validados.

#### Ficha Catalográfica

Paiva Neto, Afonso

Uma abordagem lagrangeana para simulação de escoamentos de fluidos viscoplásticos e multifásicos / Afonso Paiva Neto; orientador: Geovan Tavares dos Santos. — Rio de Janeiro : PUC-Rio, Departamento de Matemática, 2007.

v., 91 f: il. ; 29,7 cm

1. Tese (doutorado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Matemática.

Inclui referências bibliográficas.

1. Matemática – Tese. 2. Dinâmica de Fluidos Computacional. 3. Smoothed Particle Hydrodynamics. 4. Fluido Viscoplástico. 5. Fluido Multifásico. 6. Análise Numérica. 7. Animação Computacional. 8. Computação Gráfica. I. Santos, Geovan Tavares dos. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Matemática. III. Título.

## Agradecimentos

Aos meus pais, meu irmão, meus sobrinhos e minha família que souberam lidar com minha ausência e que sem o apoio deles esse sonho não seria realizado.

À Lucilly pela sua paciência, seu amor e também por ser a minha eterna fonte de inspiração.

Ao meu orientador Professor Geovan Tavares pelo apoio durante a realização desse trabalho.

Ao Professor Paulo Roberto Mendes que contribuiu com importantes sugestões que melhoraram a qualidade desse trabalho.

Aos Professores Hélio Lopes, Marcos Craizer, Sinésio Pesco e Thomas Lewiner, os quais compartilharam grandes idéias e boas gargalhadas durante esse tempo.

Aos meus amigos Etereldes (Manjator), Fabiano (Véia Murrinha), Thiago (MochI) e Renner (BiGodz) que ao longo desses anos se tornaram meus irmãos, a minha outra família.

Aos meus colegas da PUC-Rio e do IMPA os quais me fizeram sentir o Rio a minha segunda casa, em especial Adelailson, Alex, Aninha, André (Galeguim), André (Galão), Citeli, Fernando (Ferma), Francisco (Fio), LG, Luís (Lubalf), Marquinhos, Rodrigo (Gaúcho) e Zé Luiz.

Ao pessoal do Departamento de Matemática da PUC-Rio pela ajuda e pelo café de todos os dias, em particular à Ana Cristina, Creuza, Kátia, Orlando e Otávio.

Ao CNPq, à Petrobras e à PUC-Rio, pelos auxílios concedidos, sem os quais esse trabalho não teria sido concluído.

## Resumo

Paiva Neto, Afonso; Santos, Geovan Tavares dos. **Uma abordagem lagrangeana para simulação de escoamentos de fluidos viscoplásticos e multifásicos**. Rio de Janeiro, 2007. 91p. Tese de Doutorado — Departamento de Matemática, Pontifícia Universidade Católica do Rio de Janeiro.

Na última década em Computação Gráfica, foram desenvolvidas várias técnicas de simulação do comportamento de objetos como corpos rígidos, água, fumaça, cabelo e tecido. Enquanto essas técnicas se concentram em modelos físicos simplificados através de fluidos newtonianos monofásicos ou de sólidos ideais, propomos aqui novas técnicas de simulação de fluidos viscoplásticos e multifásicos baseadas numa abordagem lagrangeana da equação de quantidade de movimento. Essas técnicas consistem na discretização do fluido através de um sistema de partículas ao invés dos tradicionais métodos baseados em malhas. A simulação computacional da dinâmica de fluidos é feita utilizando o método numérico conhecido como SPH (Smoothed Particle Hydrodynamics). Nessa tese, representamos um material viscoplástico como um fluido não-newtoniano que varia entre os estados sólido e líquido, e de alta para baixa viscosidade dependendo de uma força externa aplicada ao material ou de sua própria temperatura. A simulação de fluidos multifásicos é realizada através de um método híbrido malha-partícula a fim de garantir a incompressibilidade no método SPH.

## Palavras-chave

Dinâmica de Fluidos Computacional. Smoothed Particle Hydrodynamics. Fluido Viscoplástico. Fluido Multifásico. Análise Numérica. Animação Computacional. Computação Gráfica.

## Abstract

Paiva Neto, Afonso; Santos, Geovan Tavares dos. **A lagrangian approach for simulating viscoplastic and multiphase fluids.**

Rio de Janeiro, 2007. 91p. PhD Thesis — Departament of Mathematics, Pontifícia Universidade Católica do Rio de Janeiro.

In the last decade in Computer Graphics, they were development many techniques to simulate the behavior of objects such as rigid bodies, water, smoke, hair and cloth. While these techniques focus in simplified physical models through monophasic fluids or ideal solids, we propose new simulation techniques of viscoplastic and multiphase fluids based on a lagrangian approach of momentum equation. These techniques consist in the fluid discretization through particles system instead of traditional grid-based methods. The computational fluid dynamics is performed using the method called SPH (Smoothed Particle Hydrodynamics). In this thesis, we represent a viscoplastic material like a non-newtonian fluid that it varies between solid and liquid states, and from low to high viscosity depending on the external force applied on material or its own temperature. The simulation of multiphase fluids is made using a hybrid method grid-particle to guarantee the incompressibility of SPH method.

## Keywords

Computational Fluid Dynamics. Smoothed Particle Hydrodynamics. Viscoplastic Fluid. Multiphase Fluid. Numerical Analysis. Computer Animation. Computer Graphics.

## Sumário

1	Introdução	<b>13</b>
1.1	Trabalhos anteriores	14
1.2	Contribuições	17
2	Dinâmica dos fluidos clássica	<b>19</b>
2.1	Formulação lagrangeana	20
2.2	Fluidos não-newtonianos	21
3	Dinâmica dos fluidos computacional	<b>22</b>
3.1	Métodos eulerianos	22
3.2	Métodos lagrangeanos	24
3.3	Métodos híbridos	25
4	Dinâmica dos fluidos usando sistema de partículas	<b>27</b>
4.1	Smoothed Particle Hydrodynamics (SPH)	27
4.2	Equações de Navier–Stokes usando SPH	33
5	Implementação SPH	<b>36</b>
5.1	Busca de partículas vizinhas	36
5.2	Integração numérica temporal	39
5.3	Condição de fronteira	41
5.4	Representação implícita da superfície livre	42
5.5	Exemplo numérico	45
6	Fluidos viscoplásticos	<b>49</b>
6.1	Fluido Newtoniano Generalizado	50
6.2	Transição de fase	51
6.3	Condição de fronteira	54
6.4	Aspectos numéricos	59
6.5	Implementação	60
6.6	Resultados	62
7	Fluidos multifásicos incompressíveis	<b>70</b>
7.1	Formulação SPH para fluidos multifásicos	71
7.2	Incompressibilidade SPH através de um grid	71
7.3	Implementação	74
7.4	Resultados	79
8	Conclusão e trabalhos futuros	<b>83</b>
	Referências Bibliográficas	<b>85</b>



## Lista de figuras

1.1	Exemplos de materiais viscoplásticos que sofrem deformação plástica ou que derretem.	14
1.2	Fluido multifásico: hidrodinâmica de uma bolha de ar dentro da água. À esquerda, uma imagem de uma simulação real e a direita uma imagem de uma simulação numérica com o método híbrido grid-partícula proposto nessa tese.	15
2.1	Modelos de um volume de controle finito de fluido $dV$ : (a) Volume de controle fixo no espaço com o escoamento do fluido passando através dele. (b) Volume de controle movendo ao longo do escoamento com velocidade $v$ igual à velocidade local do fluxo naquele ponto.	20
3.1	(a) Discretização euleriana com campo de velocidade $v(x)$ nos vértices $x$ do grid. A região em azul representa o fluido. Representação lagrangeana: (b) discretização utilizando uma malha com campo de velocidade $v(x)$ nos vértices $x$ da malha. (c) discretização através de um sistema de partículas com campo de velocidade $v(p)$ nas partículas $p$ do sistema.	23
4.1	Função núcleo: partículas fora do raio de influência $\kappa h$ são descartadas. Em destaque, o gráfico do núcleo 1D e de suas derivadas de primeira e segunda ordem.	29
5.1	Uma visão geral do ciclo de simulação de um sistema SPH.	36
5.2	Busca das partículas vizinhas utilizando um grid uniforme bidimensional com o espaçamento sendo o raio de influência $\kappa h$ . A região amarela representa as células do grid onde será realizada a busca.	37
5.3	Estrutura de árvore utilizada no algoritmo de busca e a subdivisão hierárquica do espaço bidimensional. A busca é realizada através de um teste de intersecção do cubo envolvente da partícula $i$ (região em amarelo) com as células representadas pelos nós da árvore.	38
5.4	Esquema de integração leap-frog: posição $x_i$ e velocidade $v_i$ numa partícula $i$ são avaliadas de maneira intercalada em relação ao tempo $t$ .	40
5.5	As partículas fantasmas (vermelho) são utilizadas para simular as condições de fronteira sólidas, evitando que as partículas de fluido (azul) escapem do domínio do problema.	41
5.6	Tabela do algoritmo de Marching Cubes (MC) que codifica a triangulação de cada cubo do grid dependendo da configuração dos sinais em seus vértices.	43
5.7	Superfície implícita de duas esferas próximas uma da outra. À esquerda a superfície é gerada através do algoritmo de MC original. À direita a superfície é gerada utilizando o algoritmo de MC com garantias topológicas.	44

5.8	Visualização da superfície implícita extraída do modelo Stanford Bunny constituído de partículas.	44
5.9	Triangulação adaptativa da isosuperfície do torus utilizando um algoritmo robusto de DMC. As regiões em verde representam as regiões da superfície que possuem garantias topológicas e geométricas, enquanto as regiões com ambigüidades (vermelho) são resolvidas com um número pequeno de refinamentos.	45
5.10	Configuração inicial do problema.	46
5.11	Simulação SPH da quebra de uma barragem utilizando $10^4$ partículas.	48
6.1	Modelo de Fluido Newtoniano Generalizado: o mapa de cores representa a viscosidade de cada partícula, variando de baixa (azul) para alta viscosidade (vermelho). Note o salto da viscosidade criado pela força aplicada pela mão, quanto maior a força menor será a viscosidade do fluido.	50
6.2	Comportamento viscoplástico do fluido.	50
6.3	Variações da viscosidade de um fluido viscoplástico inicializado numa esfera com 1200 partículas. Acima, reduzimos a viscosidade do fluido usando $J = 15$ e abaixo aumentamos a viscosidade com $J = 150$ .	51
6.4	Temperatura das 10188 partículas do modelo Stanford Bunny: as regiões em azul escuro estão abaixo do ponto de fusão, e assim permanecem sólidas.	52
6.5	Escoamento de lava no Pão de Açúcar. A simulação da lava é feita utilizando 10137 partículas.	53
6.6	Resposta da colisão partícula $\times$ triângulo.	54
6.7	Simulação de um escoamento de lava após 1910 iterações com colisão suave (esquerda) e com colisão não-escorregadia (direita), com os mesmos parâmetros: 545 partículas e 9566 triângulos na fronteira. O mapa de cores representa a densidade de cada partícula. A colisão suave faz a lava deslizar mais rapidamente.	55
6.8	O teste de colisão realiza a busca de triângulos nas células do grid através do algoritmo de linhas de Bresenham.	56
6.9	Intersecção esfera $\times$ triângulo.	57
6.10	Intersecção esfera $\times$ segmento de reta.	58
6.11	Simulação de um escoamento de lava após 1910 iterações, sem XSPH (esquerda) e com a correção XSPH (direita), com a mesma configuração da Figura 6.7. A simulação explode sem XSPH, devido à distância arbitrariamente pequena entre as partículas.	59
6.12	Ciclo de simulação de um sistema SPH para fluidos não-newtonianos.	61
6.13	Derretimento do modelo Gargoyle totalmente líquido usando 6976 partículas: o mapa de cores representa a temperatura de cada partícula. A estabilidade do método preserva a forma do objeto sem uma representação explícita da malha mesmo após várias iterações.	63
6.14	Superfície chair modelada como um material viscoplástico com 7000 partículas, interagindo com um objeto sólido complexo representado pelo esqueleto de uma mão com 2351 triângulos.	65

- 6.15 Colisão de uma esfera metálica contra uma parede plástica (verde) constituída por 5000 partículas. A simulação é vista ao mesmo tempo em duas posições diferentes: vista lateral (esquerda) e vista frontal (direta). 66
- 6.16 Evolução da superfície livre da simulação de derretimento do modelo Stanford Bunny com 10188 partículas, iniciando frio na base e quente no topo do modelo. 67
- 6.17 Derretimento das letras "CGF 2007" usando 10773 partículas, iniciando frio na base e quente no topo do modelo. A adaptatividade do passo de tempo permite uma simulação mais precisa com poucas iterações. 68
- 6.18 Escoamento de lava num terreno virtual de 1547 triângulos, com 4900 partículas. 69
- 7.1 Os vetores associados à  $P$  e  $b$  seguem a ordenação natural do grid 2D. 73
- 7.2 Ciclo de simulação do método híbrido grid-partícula: em azul são as etapas que utilizam SPH e em verde as etapas que utilizam o grid da pressão. 74
- 7.3 Configuração do método híbrido grid-partícula. A dimensão do grid de pressão é a mesma que a do grid utilizado na busca de partículas vizinhas. 75
- 7.4 Condição de Neumann num grid  $(L+2) \times (M+2)$ : as setas indicam os valores de  $P$  copiados para os vértices da fronteira. 76
- 7.5 A pressão na partícula  $i$  é obtida através de uma interpolação dos vértices de  $P$  (verde). 77
- 7.6 Gráfico do núcleo quadrático e de sua derivada de primeira ordem. 78
- 7.7 Instabilidade de Rayleigh-Taylor formada entre dois fluidos estratificados utilizando 6000 partículas: um mais denso (azul) e um menos denso (cinza). Da esquerda para direita, as imagens acima mostram a evolução da instabilidade nos instantes: 0.0s, 0.8s, 1.2s, 1.7s e 2.1s. 79
- 7.8 Evolução da altura da instabilidade de Rayleigh-Taylor: comparação da simulação numérica com a solução analítica. 80
- 7.9 Campo vetorial da velocidade: (a) As cores representam a magnitude da velocidade do fluido, variando de baixa velocidade (azul) para alta (vermelho). (b) No campo de fase as cores representam a variação do coseno do ângulo formado pelo vetor velocidade e o eixo das abscissas indo de -1 (vermelho) até 1 (azul). (c) Os vórtices são melhores identificados através das linhas de fluxos. 81
- 7.10 Movimento hidrodinâmico de uma gota de um fluido mais leve (azul) no interior de um fluido mais pesado (cinza) utilizando 6000 partículas. 82
- 7.11 Evolução da altura de uma gota: comparação da simulação numérica com a solução analítica. 82

## Lista de tabelas

6.1	Atributos da partícula.	61
6.2	Tempo médio de iteração em cada simulação executadas num computador com processador Pentium 4 – 2.4 GHz e com 2 Gb de RAM.	64

*You have entered the Twilight Zone  
Beyond this world strange things are known  
Use the key, unlock the door  
See what your fate might have in store...  
Come explore your dreams' creation  
Enter this world of imagination...*

**Rush, *The Twilight Zone.***

# 1

## Introdução

Nos últimos anos em Computação Gráfica foram desenvolvidas várias técnicas de animação de objetos como corpos rígidos, água, fumaça, cabelo e tecido [5, 51, 65, 35, 19]. Essas técnicas se concentram em modelos físicos simplificados como fluidos newtonianos monofásicos ou de sólidos ideais — sólidos que não sofrem nenhuma deformação plástica. Nessa tese, propomos dois novos métodos de simulação de fluidos: um método de animação visualmente realístico que descreve o comportamento físico de materiais viscoplásticos e um método híbrido grid-partícula para simular fluidos multifásicos incompressíveis. Ao contrário dos tradicionais métodos com malha utilizados em simulações computacionais de dinâmica dos fluidos, discretizamos o fluido através de um sistema de partículas baseado num método numérico puramente lagrangeano conhecido como *Smoothed Particle Hydrodynamics* (SPH) [48].

Os fenômenos de deformação plástica e de derretimento de materiais sólidos ocorrem com frequência na natureza (veja Figura 1.1). Com o objetivo de simular fisicamente o comportamento complexo desses fenômenos, propomos um método utilizando SPH na discretização de fluidos viscoplásticos através do modelo físico de *Fluido Newtoniano Generalizado* proposto por Mendes et al. [45]. A generalidade desse modelo reproduz eficientemente os efeitos viscosos desses fenômenos, onde a transição de fase é dada por uma simples relação entre a viscosidade e a temperatura do fluido.

O SPH é uma técnica originalmente criada para simular a dinâmica de fluidos compressíveis. Posteriormente, Monaghan representou um fluido incompressível através de um fluido quasi-compressível com alta velocidade do som [47], onde a pressão é calculada através de uma equação de estado. Os resultados utilizando essa abordagem são aceitáveis para simulações de superfície livre e escoamentos com baixo número de Reynolds [50]. Porém, para capturar a vorticidade que surge devido à diferença de densidade em simulações de fluidos multifásicos (Figura 1.2), essa abordagem requer um passo de tempo de integração muito pequeno. Com intuito de criar um método mais eficiente para simular fluidos multifásicos, propomos um novo método híbrido grid-partícula baseado no *método da projeção* [62]. No método proposto, a advecção

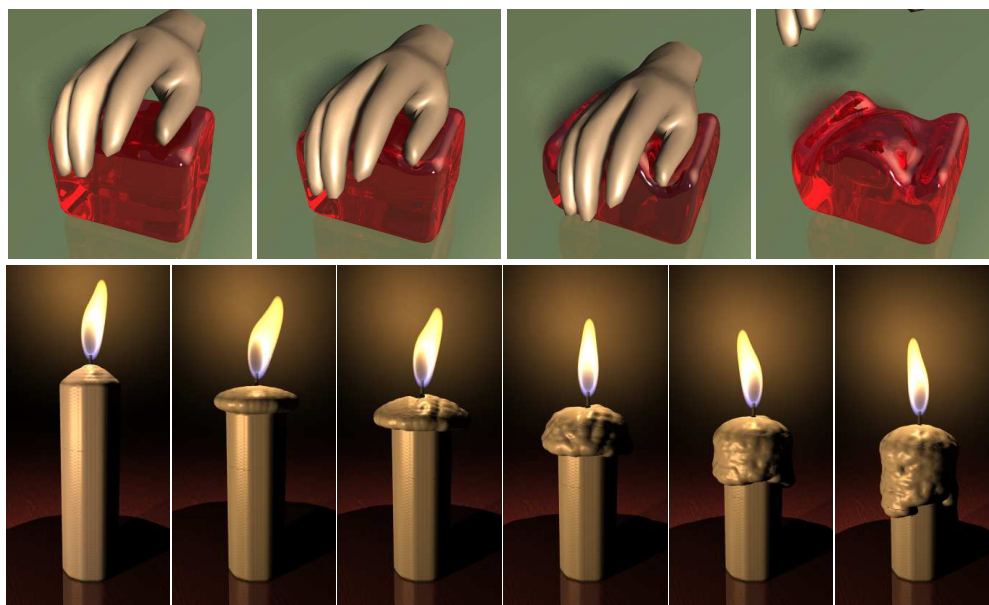


Figura 1.1: Exemplos de materiais viscoplásticos que sofrem deformação plástica ou que derretem.

do fluido é feita através do SPH enquanto que a incompressibilidade é dada implicitamente pelo cálculo da pressão através de uma equação de Poisson discretizada num grid fixo no espaço.

Essa tese é dividida nos seguintes capítulos: No Capítulo 2, apresentaremos as equações físicas que modelam a dinâmica dos fluidos. Enquanto que no Capítulo 3, mostraremos as classes de métodos numéricos utilizados para discretizar o modelo físico. No Capítulo 4, apresentaremos uma abordagem detalhada do método SPH. No Capítulo 5, discutiremos detalhes básicos de implementação computacional de um sistema SPH. Nos Capítulos 6 e 7, apresentaremos uma nova formulação SPH para fluidos não-newtonianos e um novo método híbrido grid-partícula para simular fluidos multifásicos, respectivamente. E finalmente no Capítulo 8, apresentamos as conclusões dessa tese e sugerimos trabalhos futuros.

O material complementar e os vídeos das animações mostradas nessa tese podem ser encontrados em <http://www.mat.puc-rio.br/~apneto/thesis>.

## 1.1 Trabalhos anteriores

Nessa seção, resumiremos os trabalhos de maior relevância relacionados à simulação de materiais viscoplásticos e fluidos multifásicos em quatro tópicos.

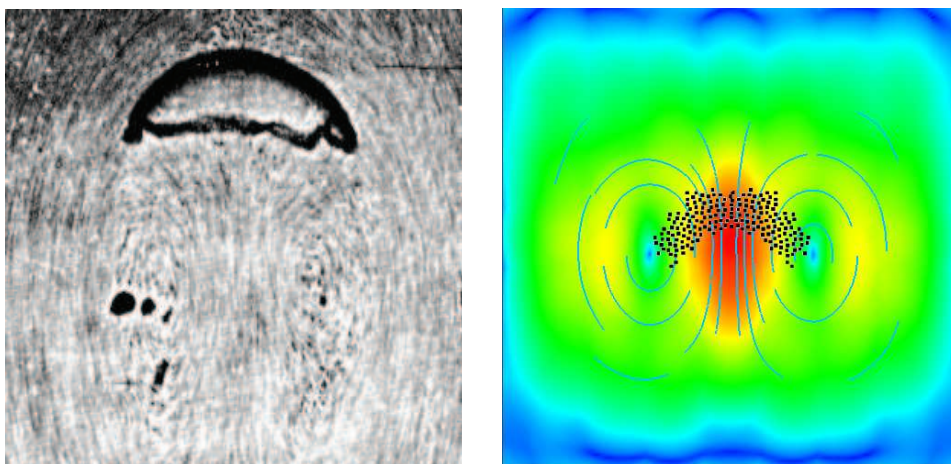


Figura 1.2: Fluido multifásico: hidrodinâmica de uma bolha de ar dentro da água. À esquerda, uma imagem de uma simulação real e a direita uma imagem de uma simulação numérica com o método híbrido grid-partícula proposto nessa tese.

**Fluidos não-newtonianos.** Existem poucos trabalhos na literatura em Computação Gráfica utilizando fluidos não-newtonianos. Goktekin et al. [25] propuseram um método utilizando um grid euleriano para simular fluidos viscoelásticos. Eles utilizaram um modelo linear de Maxwell com condição de tensão limite de von Mises. Clavet et al. [12] usaram SPH para simular fluidos viscoelásticos através de uma combinação linear de molas elásticas entre as partículas obedecendo à lei de Hooke. Outro método usando SPH foi proposto por Mao & Yang [44], onde o tensor de tensões viscosas é deduzido a partir de uma aproximação SPH do modelo co-rotacional de Maxwell.

**Deformação plástica.** A simulação de materiais sólidos que sofrem deformação plástica é feita usualmente através da formulação lagrangeana de elementos finitos utilizando uma malha tetraedral combinada com o critério de von Mises, condição que controla o início da deformação. O'Brien et al. [54] utilizaram essa formulação para simular uma ampla variedade de materiais sólidos que sofrem uma deformação significativa antes de se fraturarem. Infelizmente, uma alta deformação compromete a razão de aspecto dos tetraedros da malha gerando uma simulação numericamente instável. A fim de evitar essas instabilidades numéricas, Bargteil et al. [6] propuseram uma técnica de remalhamento baseada numa triangulação de Delaunay com o objetivo de garantir que as matrizes base associadas aos elementos da malha sejam bem-condicionadas. Entretanto, essa técnica de remalhamento é restrita a materiais que não sofrem modificações em sua topologia. Finalmente, Müller et al. [52] criaram um método pontual para simular materiais elastoplásticos baseados num modelo



físico de mecânica dos sólidos através de uma lei de Hooke generalizada onde a rigidez do material é determinada através do módulo de Young e a compressibilidade do material pela razão de Poisson. Eles utilizaram o tensor de deformação de Green calculando o tensor campo de velocidade com *Moving Least-Squares* (MLS). Finalmente, Keiser *et al.* [30] introduziram o método SPH tradicional no método anterior. Além disso, eles utilizaram uma técnica pontual conhecida como *splatting* para visualizar a superfície livre do fluido.

**Derretimento de objetos sólidos e escoamento de lava.** A idéia de simular o derretimento de objetos sólidos através do acoplamento da viscosidade com a temperatura aparece em diferentes contextos na literatura. Carlson *et al.* [10] propuseram uma modificação do método baseado num grid euleriano conhecido como *Marker-and-Cell* (MAC) [26], utilizando um integrador implícito para garantir a estabilidade numérica da simulação de fluidos com alta viscosidade. Wei *et al.* [69] criaram um autômato celular com o objetivo de substituir as equações de Navier-Stokes por um conjunto de regras simples em cada autômato. Stora *et al.* [66] simularam um escoamento de lava descendo sobre um terreno. Eles utilizaram uma versão simplificada de SPH, representado a viscosidade da lava por um modelo numérico de viscosidade artificial ao invés de um modelo físico.

**Fluidos multifásicos.** A simulação de fluidos multifásicos é feita geralmente utilizando métodos com malha, como por exemplo, o método MAC [26]. No MAC além do cálculo da pressão ser feito de forma implícita, a interface entre os dois fluidos é detectada graças a uma função que identifica se uma célula do grid contém a interface formada pelos fluidos. Sousa *et al.* [16] criaram um método robusto de simulação de fluidos multifásicos através de uma versão generalizada do MAC combinada com o método de mínimos quadrados para realizar o cálculo da tensão superficial. Assim como o MAC, os métodos sem malha *Moving Particle Semi-implicit* (MPS) [33] e *Projection Smoothed Particle Hydrodynamics* (PSPH) [15], resolvem a pressão implicitamente através do método da projeção e a interface é dada naturalmente pelas partículas. Losasso *et al.* [41] combinaram o método da projeção com o método de level-sets para simular a fronteira livre de fluidos multifásicos que sofrem reações químicas. Tartakovsky & Meakin [67] utilizaram o método SPH quasi-compressível para simular fluidos multifásicos miscíveis, onde a troca de massa entre as fases é dada por um processo de difusão. Liu *et al.* [39] criaram um método híbrido grid-partícula para simular fluidos multifásicos. No método deles, eles representam o fluido mais denso por partículas MPS enquanto que

o fluido menos denso é discretizado através de um grid fixo. A dinâmica dos fluidos é realizada no grid através do método *Volume-of-Fluid* (VOF) [27], enquanto a tensão superficial entre os fluidos é calculada utilizando MPS.

## 1.2 Contribuições

As contribuições dessa tese são divididas em duas seções. Na primeira seção, apresentaremos as contribuições relacionadas à simulação de fluidos viscoplásticos. Na segunda seção, discutiremos as contribuições do método híbrido proposto para simular fluidos multifásicos incompressíveis.

### 1.2.1 Fluidos viscoplásticos

No Capítulo 6, introduzimos uma nova formulação SPH para fluidos não-newtonianos baseada no modelo físico de Fluido Newtoniano Generalizado. Essa formulação descreve de maneira concisa a física de fluidos viscoplásticos utilizando uma simples função para o cálculo da viscosidade (veja Seção 6.1).

Diferentemente dos trabalhos anteriores, os quais simulam deformações plásticas de sólidos através de modelos físicos baseados em mecânica dos sólidos, propomos um novo paradigma na simulação desse fenômeno onde o sólido é representado como um fluido não-newtoniano de alta viscosidade.

Além disso, nas simulações de derretimento de objetos sólidos a viscosidade é controlada pela temperatura, logo é necessária uma boa aproximação por partículas da equação do calor. Por essa razão, introduzimos na Computação Gráfica uma nova e eficiente aproximação por partículas do operador laplaciano que envolve apenas derivadas de primeira ordem, ao invés de derivadas de segunda ordem (veja Seção 6.4).

Ao contrário das simulações SPH usuais, onde a condição de fronteira é realizada através de partículas fantasmas (Seção 5.3), propomos um novo tratamento para a condição de fronteira baseado num critério puramente geométrico. Nessa nova abordagem, as fronteiras do domínio são representadas por malhas triangulares e a interação da fronteira com as partículas é feita através de um simples teste geométrico de colisão (veja Seção 6.3). O teste de colisão além de ser eficiente, pois reduz consideravelmente o número de partículas no sistema, permite simular a influência da topografia de um terreno inclinado num escoamento de lava.

Com intuito de melhorar a estabilidade numérica do sistema, introduzimos algumas ferramentas numéricas no método. Primeiramente, apresentamos uma nova variante do método SPH conhecida como XSPH [46]. A finalidade do

uso XSPH é o de evitar a formação de aglomerados instáveis de partículas (veja Seção 6.4.2). Depois, utilizamos a equação da continuidade para o cálculo da densidade (equação (4-37)) a fim de evitar a deficiência de partículas na fronteira do fluido. Finalmente, introduzimos novamente o conceito de viscosidade artificial nas simulações SPH e um passo de tempo de integração adaptativo (Seções 6.4.1 e 6.4.3).

O Capítulo 6 da tese é uma extensão dos trabalhos de Paiva *et al.* em [59, 60], onde foram adicionados outros efeitos viscoplásticos tais como deformação plástica e escoamento de lava.

### 1.2.2

#### Fluidos multifásicos

No Capítulo 7, introduzimos um novo e eficiente método híbrido para simular fluidos multifásicos incompressíveis através do acoplamento do SPH com o método de diferenças finitas. Ao contrário do SPH, onde a pressão é obtida através de uma equação de estado, a incompressibilidade do método proposto é garantida devido ao cálculo implícito da pressão. Do ponto de vista dos métodos semi-implícitos como MPS e PSPH, que calculam a pressão fazendo uma aproximação por partículas da equação de Poisson, o nosso método não depende da posição e nem do número de partículas do sistema graças à discretização da equação de Poisson através de um grid fixo no espaço. Essa independência produz sistemas lineares menores, mais estáveis e mais eficientes.

Além disso, o uso do grid nos permite impor facilmente condições de fronteiras essenciais como a condição de Neumann para a pressão, tarefa geralmente difícil de ser realizada em métodos sem malha [23].

## 2

### Dinâmica dos fluidos clássica

Para simular o comportamento de um fluido, precisamos ter uma representação matemática do estado de um fluido num instante de tempo. A quantidade física mais importante a ser representada é a velocidade do fluido, pois a velocidade determina não só como o próprio fluido se move, mas também como a densidade do fluido varia.

O modelo matemático que descreve o comportamento de um fluido é dado por um conjunto de equações diferenciais parciais conhecido como *equações de Navier–Stokes*. Há duas abordagens para as equações de Navier–Stokes: a *forma euleriana* e a *forma lagrangeana*. Na forma euleriana o volume de controle finito do fluido está fixo no espaço, enquanto na forma lagrangeana o volume de controle finito se move junto com o escoamento (veja Figura 2.1). Tradicionalmente as equações de governo de fluidos são dadas na forma euleriana:

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (\rho \mathbf{v}) \quad (2-1)$$

$$\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} = -\frac{1}{\rho} \nabla p + \frac{1}{\rho} \nabla \cdot \mathbf{S} + \mathbf{g}, \quad (2-2)$$

onde  $t$  é o tempo,  $\mathbf{v}$  o vetor velocidade,  $\rho$  a densidade,  $p$  a pressão,  $\mathbf{g}$  o vetor de aceleração da gravidade,  $\mathbf{S}$  tensor extra-tensão e  $\nabla$  o operador gradiente. Por sua vez, o tensor  $\mathbf{S}$  é determinado a partir do coeficiente de viscosidade do fluido  $\mu$  e pelo tensor taxa de deformação  $\mathbf{D}$ :

$$\mathbf{S} = 2\mu \left( \mathbf{D} - \frac{1}{3} \text{traço}(\mathbf{D}) \mathbf{I} \right), \quad \text{com } \mathbf{D} = \frac{1}{2} (\nabla \mathbf{v} + \nabla \mathbf{v}^T). \quad (2-3)$$

Denotamos  $\nabla \mathbf{v}^T$  como sendo a transposta de  $\nabla \mathbf{v}$  e  $\mathbf{I}$  como sendo o tensor identidade. Note que,  $\text{traço}(\mathbf{D}) = \nabla \cdot \mathbf{v}$ .

As equações de Navier–Stokes são deduzidas a partir das leis físicas de conservação<sup>1</sup> [3]. A equação (2-2) é uma aplicação direta da segunda Lei de Newton e diz respeito à conservação de momento. A equação (2-1) obedece a lei de conservação de massa e é conhecida como *equação da continuidade*, a qual é deduzida a partir do teorema de Gauss. Se o fluido possuir densidade constante, isto é  $\frac{\partial \rho}{\partial t} = 0$ , podemos simplificar a equação (2-1) como sendo

<sup>1</sup>Nessa tese não iremos abordar a lei de conservação de energia, apenas massa e momento.

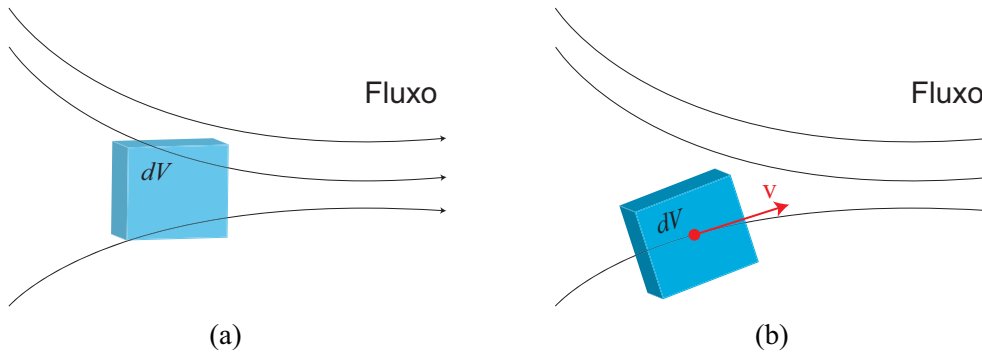


Figura 2.1: Modelos de um volume de controle finito de fluido  $dV$ : (a) Volume de controle fixo no espaço com o escoamento do fluido passando através dele. (b) Volume de controle movendo ao longo do escoamento com velocidade  $\mathbf{v}$  igual à velocidade local do fluxo naquele ponto.

apenas a restrição

$$\nabla \cdot \mathbf{v} = 0. \quad (2-4)$$

O fluido que escoar com o campo velocidade cuja divergência é livre é chamado de *fluido incompressível*.

O termo viscoso  $\nabla \cdot \mathbf{S}$  pode conter várias variáveis a serem calculadas; assim a equação (2-2) na forma geral não pode ser aplicada em qualquer problema. Por essa razão, devemos especificar o tensor  $\mathbf{S}$  de acordo com o comportamento do fluido baseando-se em observações naturais e aplicá-las com o intuito de representá-lo em termos de variáveis conhecidas, tais como a velocidade. Por exemplo, se o fluido for incompressível e possuir viscosidade constante, o termo  $\nabla \cdot \mathbf{S}$  pode ser reescrito na forma  $\mu \nabla^2 \mathbf{v}$ , onde  $\mu$  é a viscosidade do fluido. Assim podemos reescrever a equação (2-2) na forma usual:

$$\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} = -\frac{1}{\rho} \nabla p + \frac{\mu}{\rho} \nabla^2 \mathbf{v} + \mathbf{g}. \quad (2-5)$$

A discretização numérica das equações de Navier–Stokes na sua forma euleriana é feita através de métodos numéricos que usam uma grade fixa no espaço para estimar as derivadas espaciais. Detalharemos mais sobre esses métodos na Seção 3.1. O leitor interessado poderá encontrar com detalhes as deduções acima no livro de Anderson [3].

## 2.1 Formulação lagrangeana

A formulação lagrangeana das equações de Navier–Stokes descreve o comportamento de um fluido no ponto de vista de uma partícula que se move junto com o fluxo do fluido. Definimos a *derivada material* como

$$\frac{d(\cdot)}{dt} \equiv \frac{\partial(\cdot)}{\partial t} + \mathbf{v} \cdot \nabla(\cdot). \quad (2-6)$$

Fisicamente a derivada material representa a variação instantânea de uma grandeza física em relação ao tempo de um elemento de fluido que se move no espaço. Substituindo no lado esquerdo das equações (2-1) – (2-2) pela derivada material, as equações de governo na forma lagrangeana são dadas como:

$$\frac{d\rho}{dt} = -\rho \nabla \cdot \mathbf{v} \quad (2-7)$$

$$\frac{d\mathbf{v}}{dt} = -\frac{1}{\rho} \nabla p + \frac{1}{\rho} \nabla \cdot \mathbf{S} + \mathbf{g}. \quad (2-8)$$

Note que cada partícula possui uma advecção própria, logo a equação (2-8) não requer o cálculo explícito do termo advectivo

$$\mathbf{v} \cdot \nabla \mathbf{v}. \quad (2-9)$$

Assim podemos também reescrever a equação (2-5) na sua forma lagrangeana

$$\frac{d\mathbf{v}}{dt} = -\frac{1}{\rho} \nabla p + \frac{\mu}{\rho} \nabla^2 \mathbf{v} + \mathbf{g}. \quad (2-10)$$

Portanto, se discretizarmos um fluido como um sistema de partículas, a equação de momento na forma euleriana (2-2), que antes era descrita por uma equação diferencial parcial, se torna na forma lagrangeana um conjunto de equações diferenciais ordinárias as quais podem ser resolvidas via integração numérica em relação à variável temporal.

Nessa tese, o modelo físico que vamos adotar é dado pelas equações de Navier–Stokes em sua forma lagrangiana. No Capítulo 4 veremos com detalhes a discretização numérica dessas equações através de sistema de partículas.

## 2.2

### Fluidos não–newtonianos

Um *fluido newtoniano*, água por exemplo, é um fluido no qual seu tensor extra-tensão  $\mathbf{S}$  possui uma dependência linear do tensor taxa de deformação  $\mathbf{D}$ . Caso contrário, o fluido é chamado de *fluido não–newtoniano*.

Entre os fluidos não–newtonianos, o comportamento dos fluidos viscoplásticos é caracterizado quando uma tensão significativa é aplicada no material antes que ele comece a fluir como um líquido (efeito pasta de dente), essa tensão crítica é conhecida como *tensão limite do escoamento* (*yield stress*). Se a tensão aplicada é menor do que a tensão correspondente a plasticidade limite então o material se comporta como um sólido. Uma vez que a plasticidade limite é excedida, o material pode fluir como um fluido newtoniano.

Fluidos não–newtonianos são estudados extensivamente em dinâmica dos fluidos computacional, e o estudo deles é conhecido como *Reologia* [57].

## 3 Dinâmica dos fluidos computacional

Existem vários métodos numéricos que foram desenvolvidos para resolver equações diferenciais parciais que modelam matematicamente um sistema dinâmico. Esses métodos se diferem de acordo com a discretização das derivadas espaciais das equações diferenciais, e são tradicionalmente classificados em *métodos eulerianos*, *métodos lagrangeanos* e *métodos híbridos*. Nesse capítulo, apresentaremos uma visão geral de cada uma dessas classes de métodos numéricos, discutindo as diferenças e as vantagens de cada classe.

### 3.1 Métodos eulerianos

Métodos eulerianos fornecem tipicamente uma representação numérica das equações de Navier–Stokes para fluidos incompressíveis (2-2) – (2-4). A representação do fluido é feita no interior das células de um reticulado retangular fixo no espaço conhecido como *grid* (Figura 3.1(a)).

O grid, além de ser o domínio computacional da simulação, auxilia na estimativa das derivadas espaciais dessas equações usando o método de diferenças finitas [3]. Tipicamente a distribuição de massa do fluido é calculada através das células enquanto o fluxo de velocidade é calculado nos vértices do grid. A forma e o volume das células assim como a posição dos vértices não se alteram durante todo processo computacional.

O campo de velocidade  $\mathbf{v}$  é atualizado em cada instante de tempo resolvendo separadamente cada termo da equação (2-2) por etapas. Uma das etapas mais importantes consiste na projeção da pressão [62]. Os métodos de projeção são baseados no teorema<sup>1</sup> abaixo:

**Teorema 3.1.1 (Decomposição de Helmholtz–Hodge)** *Um campo vetorial  $\mathbf{w}$  num domínio  $D$  pode ser decomposto unicamente na forma*

$$\mathbf{w} = \mathbf{u} + \nabla p, \quad (3-1)$$

onde  $\nabla \cdot \mathbf{u} = 0$  e  $\mathbf{u} \cdot \mathbf{n} = 0$ ,  $\mathbf{n}$  é um vetor normal a  $\partial D$ .

<sup>1</sup>A demonstração do teorema pode ser encontrada em [11].

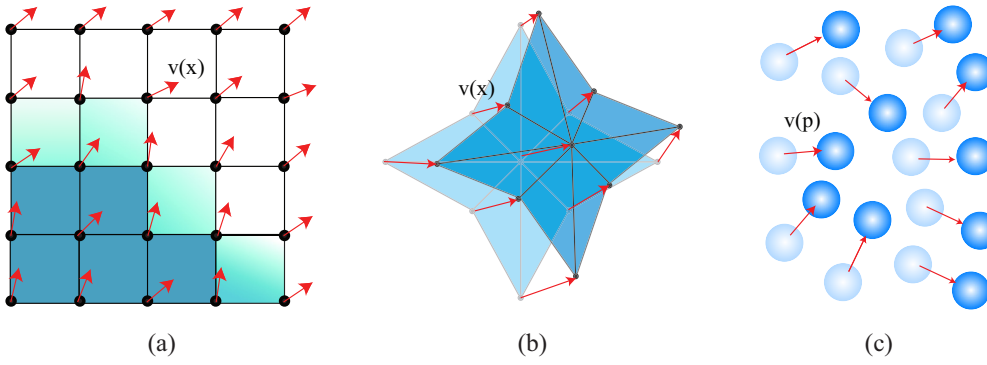


Figura 3.1: (a) Discretização euleriana com campo de velocidade  $\mathbf{v}(\mathbf{x})$  nos vértices  $\mathbf{x}$  do grid. A região em azul representa o fluido. Representação lagrangeana: (b) discretização utilizando uma malha com campo de velocidade  $\mathbf{v}(\mathbf{x})$  nos vértices  $\mathbf{x}$  da malha. (c) discretização através de um sistema de partículas com campo de velocidade  $\mathbf{v}(\mathbf{p})$  nas partículas  $\mathbf{p}$  do sistema.

A decomposição de Helmholtz–Hodge nos permite calcular implicitamente a pressão  $p^{n+1}$  num passo  $n + 1$  através da seguinte equação de Poisson:

$$\nabla^2 p^{n+1} = \frac{\rho}{\Delta t} \nabla \cdot \mathbf{v}^*. \quad (3-2)$$

com

$$\mathbf{v}^* = \mathbf{v}^n + \Delta t \left( \frac{\mu}{\rho} \nabla^2 \mathbf{v}^n + \mathbf{g} \right). \quad (3-3)$$

A pressão é a solução do sistema linear obtido da discretização dos operadores diferenciais da equação (3-2) pelo método de diferenças finitas. Finalmente, a velocidade com divergência livre  $\mathbf{v}^{n+1}$  é obtida da seguinte maneira

$$\mathbf{v}^{n+1} = \mathbf{v}^* - \frac{\Delta t}{\rho} \nabla p^{n+1}. \quad (3-4)$$

Os métodos eulerianos são rápidos, estáveis, tratam de extremas deformações e mudanças topológicas do fluido sem alterar a sua discretização. As maiores desvantagens são: a dificuldade de representar domínios com geometria complexa numa grade regular, problema de conservação de massa devido à difusão numérica, dificuldade de capturar a superfície livre de fluidos que possuem uma interface bem definida e alto armazenamento de memória, pois o domínio computacional abrange até mesmo regiões onde não há fluido.

Simulação de fluidos utilizando métodos eulerianos tornou-se bastante popular em Computação Gráfica com o trabalho de Foster & Metaxas [22]. Stam em [65], propôs um método estável na solução do termo advectivo (2-9) através de uma técnica semi-lagrangeana. Para introduzir novamente as pequenas escalas de vorticidade dissipadas numericamente em simulações de fumaça, Fedkiw *et al.* [20] adicionaram uma força artificial de *confinamento de vórtice*.



## 3.2

### Métodos lagrangeanos

A discretização numérica da formulação lagrangeana das equações de Navier–Stokes (2-7) – (2-8), se dividem em duas categorias: *métodos com malha* e *métodos sem malha*.

**Métodos com malha.** Nessa categoria, o domínio computacional é constituído por um conjunto de células irregulares (ou elementos) conectadas através de um mapa topológico formando uma malha não-estruturada. Diferentemente dos métodos eulerianos, a malha não-estruturada se movimenta dinamicamente com o fluido (Figura 3.1(b)). Conseqüentemente, se o fluido sofrer alguma mudança topológica a malha também sofrerá tal mudança. Um dos métodos lagrangeanos com malha mais conhecido e amplamente utilizado é o *método de elementos finitos* [72].

As principais vantagens dos métodos lagrangeanos com malha são: não requer nenhum esforço computacional extra para calcular o termo advectivo já que a malha não está fixa no espaço, modela facilmente domínios com geometrias complexas e irregulares graças à adaptatividade de sua malha não-estruturada, a malha envolve apenas o domínio computacional onde existe fluido.

Porém, devido ao fato da malha estar diretamente vinculada ao fluido, esses métodos apresentam algumas dificuldades, como a de simular um fluido que sofra uma grande deformação. Grandes deformações resultam em grandes distorções nos elementos da malha que acabam afetando numericamente a solução do sistema. Outra dificuldade é a captura da superfície livre do fluido, principalmente quando a superfície sofre mudanças topológicas. Uma técnica para resolver esse problema é conhecida como *remalhamento* [68].

Existem várias aplicações do método de elementos finitos em Computação Gráfica. O’Brien *et al.* [54] simularam fratura de objetos sólidos. Etmuss *et al.* [19] utilizaram elementos finitos para modelar a dinâmica de tecidos. Klingner *et al.* [31] apresentaram um método de simulação de fluidos usando uma malha tetraedral com remalhamento dinâmico.

**Métodos sem malha.** Geralmente essa categoria se refere a uma classe de métodos que utilizam um conjunto finito de partículas para discretizar o estado e a dinâmica de um sistema. Nos problemas de dinâmica dos fluidos, cada partícula está diretamente associada a atributos físicos do fluido e sua evolução é determinada através de leis de conservação de massa e momento (Figura 3.1(c)).

As vantagens desses métodos em relação aos tradicionais métodos com malha são: o fluido é representado por um sistema de partículas sem conectividade fixa, assim o tratamento de grandes deformações é relativamente mais fácil. Cada partícula representa um elemento de fluido, logo a massa do sistema é trivialmente conservada. Discretiza facilmente geometrias complexas, pois só requer a discretização inicial do domínio computacional. Refinamento de um sistema de partículas é uma tarefa computacionalmente mais simples de ser executada do que refinar uma malha. Captura facilmente superfície livres e suas mudanças topológicas.

Nessa tese, utilizamos um método sem malha conhecido como Smoothed Particle Hydrodynamics (SPH). O método SPH foi criado na década de 70 por Monaghan & Gingold [24] e Lucy [42] para simular fenômenos envolvendo fluidos compressíveis em astrofísica, tais como colisão de estrelas, formação de galáxias e supernovas. Desbrun & Cani em [17], introduziram o SPH na comunidade de Computação Gráfica com o objetivo de simular corpos deformáveis. Müller *et al.* [51] utilizaram o SPH para simular fluidos com tensão superficial em tempo de execução interativo. Recentemente, simulações SPH se tornaram bastante populares na indústria cinematográfica de efeitos especiais, onde foi utilizado em simulações de água nos filmes *Poseidon* e *300* (<http://www.nextlimit.com>). Além de aplicações em astrofísica e Computação Gráfica, o SPH é aplicado com sucesso em simulações mais complexas de hidrodinâmica relativística e colisões de íons pesados em altas energias [1, 32]. No Capítulo 4 mostraremos com detalhes o método SPH. Variações e extensões desse método podem ser encontradas em [37, 38].

### 3.3

#### Métodos híbridos

Apesar de possuírem características distintas e complementares os métodos eulerianos e lagrangeanos podem ser combinados de forma simbiótica a fim de evitar as desvantagens de cada método. Através dessa idéia, foram desenvolvidos dois métodos híbridos que utilizam tanto a abordagem euleriana quanto a lagrangeana: o *Coupled Eulerian Lagrangian* (CEL) [43] e o *Arbitrary Lagrange Eulerian* (ALE) [8]. Esses métodos são usados com frequência em simulações que envolvem a interação fluido–estrutura.

O método CEL combina os métodos euleriano e lagrangeano em regiões separadas do domínio computacional. A prática mais comum é utilizar a abordagem lagrangeana pra discretizar sólidos (estruturas) e a abordagem euleriana na discretização de fluidos. A região lagrangeana e a região euleriana interagem continuamente trocando informação computacional através de um

mapeamento especial na interface formada nessas duas regiões no grid.

O método **ALE** é semelhante a uma técnica de deformação de uma malha lagrangeana de elementos finitos. O **ALE** é obtido introduzindo uma malha computacional de referência onde os seus vértices podem se deslocar através do movimento lagrangeano natural, ou podem estar fixos no espaço como nos métodos eulerianos, ou podem alguns vértices se movimentarem arbitrariamente e independentemente enquanto outros ficam fixos. O livre movimento dos vértices no método **ALE** tem como finalidade minimizar a distorção dos elementos da malha.

## 4

### Dinâmica dos fluidos usando sistema de partículas

Nesse capítulo, introduziremos o método SPH e sua aplicação em dinâmica dos fluidos. Na Seção 4.1, apresentaremos os conceitos básicos de SPH na sua forma contínua e discreta, em seguida definiremos regras para os operadores diferenciais discretizados através do método SPH. Em seguida, na Seção 4.2 mostraremos a aproximação SPH das equações de Navier–Stokes.

#### 4.1

##### Smoothed Particle Hydrodynamics (SPH)

O método SPH consiste num método de interpolação das propriedades físicas de um sistema hidrodinâmico e de aproximação das derivadas espaciais de campos contínuos utilizando um conjunto discreto de partículas. A principal idéia do método SPH é baseada no conceito da representação integral de uma função contínua  $f : \mathbb{R}^d \rightarrow \mathbb{R}$  definida num domínio aberto  $\Omega \subset \mathbb{R}^d$  tal que  $\mathbf{x} \in \Omega$ . Essa representação é feita através da convolução de  $f$  com a *distribuição delta de Dirac*, a qual é definida como sendo a distribuição que possui a seguinte propriedade fundamental:

$$f(\mathbf{x}) = \int_{\Omega} f(\mathbf{x}') \delta(\mathbf{x} - \mathbf{x}') d\mathbf{x}' \quad (4-1)$$

e também satisfaz

$$\delta(\mathbf{x} - \mathbf{x}') = \begin{cases} \infty, & \mathbf{x} = \mathbf{x}' \\ 0, & \mathbf{x} \neq \mathbf{x}' \end{cases} . \quad (4-2)$$

Note que em particular tomando  $f \equiv 1$  em (4-1), temos a condição:

$$\int_{\Omega} \delta(\mathbf{x} - \mathbf{x}') d\mathbf{x}' = 1. \quad (4-3)$$

Portanto, em outras palavras, a distribuição delta de Dirac é nula em qualquer ponto exceto em  $\mathbf{x}'$ , onde seu valor é infinito o suficiente para que a área sob seu gráfico seja exatamente 1.

Embora não conseguimos definir uma função que satisfaça as propriedades da distribuição delta de Dirac, a Análise Funcional nos permite definir a distribuição delta de Dirac como sendo o limite de funções suaves [28]. Portanto, podemos definir o operador de aproximação SPH (denotado por  $\langle \rangle$ )

da função  $f(\mathbf{x})$  simplesmente substituindo a distribuição delta de Dirac na equação (4-1) por uma função suave  $W$ , logo a equação (4-1) é reescrita na forma:

$$\langle f(\mathbf{x}) \rangle = \int_{\Omega} f(\mathbf{x}') W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}', \quad (4-4)$$

onde  $h$  é chamado de *comprimento suave* e que por sua vez define o *raio de influência*  $\kappa h$  da função  $W$ , sendo  $\kappa$  o fator de escala de  $W$ . A função  $W$  é conhecida como *função núcleo suave* ou simplesmente *núcleo*.

O núcleo é geralmente escolhido como sendo uma função par devido à sua forma simétrica e tipicamente satisfaz algumas propriedades:

- Não-negativo:  $W(\mathbf{x}, h) \geq 0$ .
- Suavidade:  $W(\mathbf{x}, h) \in C^k(\Omega)$ , com  $k > 1$ .
- Partição da unidade:  $\int_{\Omega} W(\mathbf{x}, h) d\mathbf{x} = 1$ .
- Suporte compacto:  $W(\mathbf{x} - \mathbf{x}', h) = 0$ , quando  $\|\mathbf{x} - \mathbf{x}'\| > \kappa h$ .
- Convergência:  $W(\mathbf{x}, h) \rightarrow \delta$ , quando  $h \rightarrow 0$ .

Na literatura há várias funções núcleo [38], as mais comuns são a função Gaussiana e as funções splines. Note que apesar da Gaussiana não possuir suporte compacto ela decai rapidamente para zero, logo possui numericamente a propriedade de suporte compacto. Nessa tese, utilizamos uma função spline de quinta ordem (Figura 4.1),

$$W(\mathbf{x} - \mathbf{x}_j, h) = \alpha_d \cdot w\left(\frac{\|\mathbf{x} - \mathbf{x}_j\|}{h}\right) \quad \text{com} \quad (4-5)$$

$$w(q) = \begin{cases} (3-q)^5 - 6(2-q)^5 + 15(1-q)^5; & 0 \leq q < 1 \\ (3-q)^5 - 6(2-q)^5; & 1 \leq q < 2 \\ (3-q)^5; & 2 \leq q \leq 3 \\ 0; & q > 3 \end{cases}$$

a constante de normalização do núcleo  $\alpha_d$  é  $120/h$ ,  $7/478\pi h^2$  e  $3/359\pi h^3$  em espaços 1D, 2D e 3D, respectivamente.

Quando a função  $f(\mathbf{x})$  é representada numericamente, a aproximação  $\langle f(\mathbf{x}) \rangle$  deve ser mais próxima possível de  $f(\mathbf{x})$ . Independentemente da escolha do núcleo simétrico  $W$ , o método SPH possui precisão de segunda ordem e é facilmente verificada usando uma expansão em séries de Taylor da função  $f(\mathbf{x}')$  em torno de  $\mathbf{x}$ . Supondo  $f \in C^3(\Omega)$  e usando a equação (4-4) temos

$$\begin{aligned} \langle f(\mathbf{x}) \rangle &= \int_{\Omega} \left[ f(\mathbf{x}) + f'(\mathbf{x})(\mathbf{x}' - \mathbf{x}) + r((\mathbf{x} - \mathbf{x}')^2) \right] W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' \\ &= f(\mathbf{x}) \int_{\Omega} W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' \\ &+ f'(\mathbf{x}) \int_{\Omega} (\mathbf{x}' - \mathbf{x}) W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' + r(h^2), \end{aligned} \quad (4-6)$$

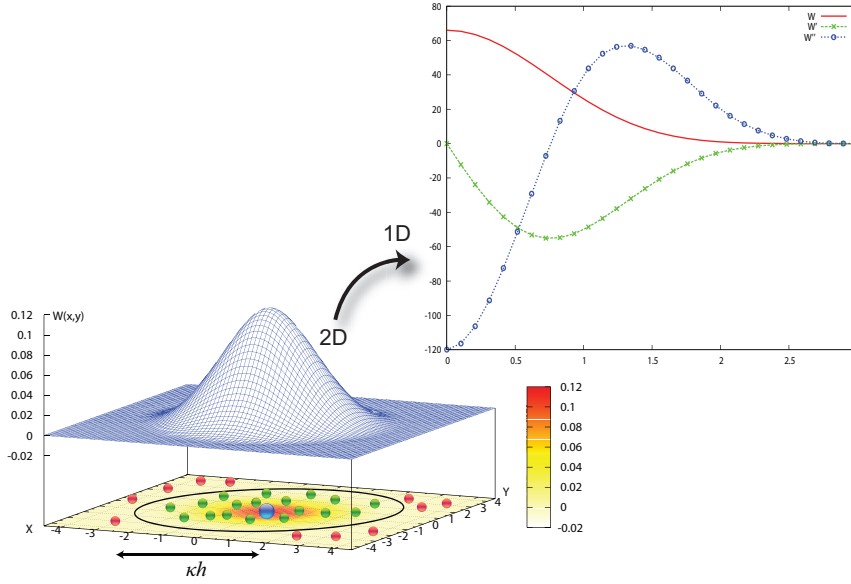


Figura 4.1: Função núcleo: partículas fora do raio de influência  $\kappa h$  são descartadas. Em destaque, o gráfico do núcleo 1D e de suas derivadas de primeira e segunda ordem.

onde  $r$  é o resíduo. Perceba que  $W$  é uma função par em relação à variável  $\mathbf{x}$  então  $(\mathbf{x}' - \mathbf{x}) W(\mathbf{x} - \mathbf{x}', h)$  será uma função ímpar, logo teremos

$$\int_{\Omega} (\mathbf{x}' - \mathbf{x}) W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' = 0. \quad (4-7)$$

Usando o fato de  $W$  ser uma partição da unidade e a equação (4-7), a equação (4-6) se torna

$$\langle f(\mathbf{x}) \rangle = f(x) + r(h^2). \quad (4-8)$$

A ordem de precisão de uma aproximação SPH é chamada de *ordem de consistência*. Dizemos que a aproximação é de ordem  $n$  se satisfaz as seguintes condições:

$$\begin{cases} M_0 = \int_{\Omega} W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' = 1 \\ M_1 = \int_{\Omega} (\mathbf{x}' - \mathbf{x}) W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' = 0 \\ M_2 = \int_{\Omega} (\mathbf{x}' - \mathbf{x})^2 W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' = 0 \\ \vdots \\ M_n = \int_{\Omega} (\mathbf{x}' - \mathbf{x})^n W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' = 0 \end{cases} \quad (4-9)$$

O método SPH é um método puramente lagrangeano; logo todo sistema é representado por um conjunto finito de partículas que possuem atributos individuais. Conseqüentemente, a representação integral contínua da aproximação SPH pode ser discretizada através de um somatório sobre todas as partículas que estão no interior de uma região delimitada pelo comprimento suave (Figura 4.1). Assim, o elemento de volume infinitesimal  $d\mathbf{x}'$  na equação (4-4) é

representado numa partícula  $j$  pelo volume  $\Delta V_j$  e que por sua vez é determinado através de atributos físicos da própria partícula como

$$\Delta V_j = \frac{m_j}{\rho_j} \quad (4-10)$$

sendo  $m_j$  a massa da partícula e  $\rho_j$  a densidade da partícula. A forma discreta da aproximação SPH é dada pela equação:

$$\begin{aligned} \langle f(\mathbf{x}) \rangle &= \sum_{j \in N(\mathbf{x})} f(\mathbf{x}_j) W(\mathbf{x} - \mathbf{x}_j, h) \Delta V_j \\ &= \sum_{j \in N(\mathbf{x})} f(\mathbf{x}_j) W(\mathbf{x} - \mathbf{x}_j, h) \frac{m_j}{\rho_j}, \end{aligned} \quad (4-11)$$

onde  $\mathbf{x}_j$  é a posição da partícula  $j$  e o conjunto  $N(\mathbf{x})$  contém todas as partículas cuja distância a partir de  $\mathbf{x}$  é menor que o raio de influência do núcleo  $\kappa h$ . A aproximação para uma função avaliada numa partícula  $i$  pode ser escrita diretamente como

$$\langle f(\mathbf{x}_i) \rangle = \sum_{j \in N(\mathbf{x}_i)} f(\mathbf{x}_j) W_{ij} \frac{m_j}{\rho_j} \quad (4-12)$$

com  $W_{ij} = W(\mathbf{x}_i - \mathbf{x}_j, h)$ .

A partir da definição do operador SPH, dadas duas funções arbitrárias  $f_1$  e  $f_2$  foram criadas regras que são bastante convenientes para problemas hidrodinâmicos mais complexos. A primeira regra diz que a aproximação SPH da soma de funções é a soma da aproximação de cada função

$$\langle f_1 + f_2 \rangle = \langle f_1 \rangle + \langle f_2 \rangle. \quad (4-13)$$

A segunda regra diz que a aproximação SPH do produto de duas funções pode ser escrita como o produto da aproximação de cada função

$$\langle f_1 f_2 \rangle \approx \langle f_1 \rangle \langle f_2 \rangle. \quad (4-14)$$

Particularmente, supondo a função  $f_2$  como uma função constante  $c$  na equação (4-14), vale a igualdade

$$\langle c f_1 \rangle = c \langle f_1 \rangle. \quad (4-15)$$

#### 4.1.1

##### Aproximação SPH das derivadas espaciais

Desde que a abordagem SPH é usada para aproximar equações diferenciais parciais através de uma formulação integral, o ingrediente principal para capturar a dinâmica de um sistema é saber como representar e discretizar os operadores diferenciais através do método SPH. A aproximação SPH do gradiente da função escalar  $f(\mathbf{x})$  é inicialmente obtida substituindo  $f(\mathbf{x})$  por  $\nabla f(\mathbf{x})$

na equação (4-4):

$$\langle \nabla f(\mathbf{x}) \rangle = \int_{\Omega} \nabla_{\mathbf{x}'} f(\mathbf{x}') W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}', \quad (4-16)$$

onde  $\nabla_{\mathbf{x}'}$  denota o gradiente da função  $f$  em relação à variável  $\mathbf{x}'$ .

Fazendo uma integração por partes na equação (4-16) temos:

$$\begin{aligned} \langle \nabla f(\mathbf{x}) \rangle &= \int_{\Omega} \nabla_{\mathbf{x}'} [f(\mathbf{x}') W(\mathbf{x} - \mathbf{x}', h)] d\mathbf{x}' \\ &- \int_{\Omega} f(\mathbf{x}') \nabla_{\mathbf{x}'} W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}'. \end{aligned} \quad (4-17)$$

Podemos utilizar o teorema da divergência de Gauss para representar a primeira integral do lado direito da equação (4-17) como uma integral sobre a superfície de bordo  $S = \partial\Omega$  do domínio de integração  $\Omega$ :

$$\begin{aligned} \langle \nabla f(\mathbf{x}) \rangle &= \int_S f(\mathbf{x}') W(\mathbf{x} - \mathbf{x}', h) \cdot \mathbf{n} dS \\ &- \int_{\Omega} f(\mathbf{x}') \nabla_{\mathbf{x}'} W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}', \end{aligned} \quad (4-18)$$

com  $\mathbf{n}$  sendo o vetor unitário normal a superfície  $S$ . Se o suporte de  $W$  estiver contido no interior de  $\Omega$ , temos que a integral de superfície do lado direito de (4-18) é nula. Caso contrário, se o suporte de  $W$  intersectar  $S$ , o suporte de  $W$  é truncado na superfície de bordo  $S$  para que a integral de superfície também se anule. Portanto, sob tais circunstâncias podemos simplificar a equação (4-18) da seguinte forma:

$$\langle \nabla f(\mathbf{x}) \rangle = - \int_{\Omega} f(\mathbf{x}') \nabla_{\mathbf{x}'} W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}'. \quad (4-19)$$

Através da regra da cadeia e pelo fato de  $W$  ser simétrico temos:

$$\nabla_{\mathbf{x}} W(\mathbf{x} - \mathbf{x}', h) = -\nabla_{\mathbf{x}'} W(\mathbf{x} - \mathbf{x}', h). \quad (4-20)$$

Portanto, substituindo (4-20) em (4-19) segue que

$$\langle \nabla f(\mathbf{x}) \rangle = \int_{\Omega} f(\mathbf{x}') \nabla_{\mathbf{x}} W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}'. \quad (4-21)$$

Resumindo, para calcular a aproximação SPH do gradiente de uma função  $f(\mathbf{x})$  basta apenas calcular o gradiente da aproximação da função, ou seja,

$$\langle \nabla f(\mathbf{x}) \rangle = \nabla \langle f(\mathbf{x}) \rangle. \quad (4-22)$$

A versão discreta do gradiente segue diretamente da equação (4-21)

$$\langle \nabla f(\mathbf{x}) \rangle = \sum_{j \in N(\mathbf{x})} f(\mathbf{x}_j) \nabla W(\mathbf{x} - \mathbf{x}_j, h) \frac{m_j}{\rho_j}. \quad (4-23)$$



Finalmente, a aproximação do gradiente de uma função avaliada numa partícula  $i$  pode ser escrita como

$$\langle \nabla f(\mathbf{x}_i) \rangle = \sum_{j \in N(\mathbf{x}_i)} f(\mathbf{x}_j) \nabla_i W_{ij} \frac{m_j}{\rho_j}, \quad (4-24)$$

onde pela regra da cadeia segue que

$$\nabla_i W_{ij} = \frac{\mathbf{x}_{ij}}{r_{ij}} \frac{\partial W_{ij}}{\partial r_{ij}}, \quad (4-25)$$

com  $\mathbf{x}_{ij} = \mathbf{x}_i - \mathbf{x}_j$  e  $r_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$ . Uma propriedade bastante útil decorrente dessa aproximação é o fato de

$$\langle \nabla W \rangle \approx \nabla \langle W \rangle = \nabla \sum_{j \in N(\mathbf{x})} W(\mathbf{x} - \mathbf{x}_j, h) \frac{m_j}{\rho_j} \approx \nabla(1) = 0. \quad (4-26)$$

A aproximação (4-24) nem sempre é precisa, pois a terceira lei de Newton (lei da ação e reação) é violada e conseqüentemente essa imprecisão destrói as propriedades de conservação nos sistemas discretizados através do SPH. Porém, quando a aproximação é combinada com um termo adicional que contém uma expressão nula  $\langle \nabla W \rangle = 0$ , isso pode produzir resultados melhores. Assim como no método de diferenças finitas onde as derivadas espaciais são discretizadas utilizando uma malha fixa no espaço, iremos definir regras para o cálculo discreto dos operadores diferenciais através da formulação SPH.

**Regra SPH I para o gradiente.** Seja  $A(\mathbf{x})$  um campo escalar, segue que

$$\langle \nabla A(\mathbf{x}_i) \rangle = \sum_{j \in N(\mathbf{x}_i)} (A(\mathbf{x}_j) - A(\mathbf{x}_i)) \nabla_i W_{ij} \frac{m_j}{\rho_j}. \quad (4-27)$$

Note que o termo  $\sum_{j \in N(\mathbf{x}_i)} A(\mathbf{x}_i) \nabla_i W_{ij} \frac{m_j}{\rho_j} = A(\mathbf{x}_i) \nabla_i \langle W_{ij} \rangle = 0$ .

**Regra SPH I para o jacobiano.** Seja  $\mathbf{A}(\mathbf{x})$  um campo vetorial, segue que

$$\langle \nabla \mathbf{A}(\mathbf{x}_i) \rangle = \sum_{j \in N(\mathbf{x}_i)} (\mathbf{A}(\mathbf{x}_j) - \mathbf{A}(\mathbf{x}_i)) \otimes \nabla_i W_{ij} \frac{m_j}{\rho_j}. \quad (4-28)$$

**Regra SPH I para o divergente.** Seja  $\mathbf{A}(\mathbf{x})$  um campo vetorial, segue que

$$\langle \nabla \cdot \mathbf{A}(\mathbf{x}_i) \rangle = \sum_{j \in N(\mathbf{x}_i)} (\mathbf{A}(\mathbf{x}_j) - \mathbf{A}(\mathbf{x}_i)) \cdot \nabla_i W_{ij} \frac{m_j}{\rho_j}. \quad (4-29)$$

Apesar das aproximações (4-27) e (4-29) serem aceitáveis, elas possuem a deficiência: não são simétricas em relação aos índices  $i$  e  $j$ . Para obtermos aproximações simétricas do gradiente de um campo escalar, necessitamos

definir um novo conjunto de regras. Consideremos a identidade abaixo

$$\frac{\nabla A(\mathbf{x})}{\rho} = \frac{A(\mathbf{x})}{\rho^\alpha} \nabla \left( \frac{1}{\rho^{1-\alpha}} \right) + \frac{1}{\rho^{2-\alpha}} \nabla \left( \frac{A(\mathbf{x})}{\rho^{\alpha-1}} \right), \text{ com } \alpha \in \mathbb{Z}. \quad (4-30)$$

Através da identidade (4-30), podemos estabelecer uma aproximação SPH da seguinte forma:

$$\left\langle \frac{\nabla A(\mathbf{x}_i)}{\rho_i} \right\rangle = \sum_{j \in N(\mathbf{x}_i)} \left( \frac{A(\mathbf{x}_j)}{\rho_i^{2-\alpha} \rho_j^\alpha} + \frac{A(\mathbf{x}_i)}{\rho_i^\alpha \rho_j^{2-\alpha}} \right) \nabla_i W_{ij} m_j. \quad (4-31)$$

E a partir da aproximação (4-31), podemos criar um novo conjunto de regras SPH para a aproximação do gradiente de campos escalares e divergente de campos vetoriais.

**Regra SPH II para o gradiente.** Seja  $A(\mathbf{x})$  um campo escalar e  $\alpha = 1$ :

$$\langle \nabla A(\mathbf{x}_i) \rangle = \sum_{j \in N(\mathbf{x}_i)} (A(\mathbf{x}_i) + A(\mathbf{x}_j)) \nabla_i W_{ij} \frac{m_j}{\rho_j}. \quad (4-32)$$

**Regra SPH II para o divergente.** Seja  $\mathbf{A}(\mathbf{x})$  um campo vetorial e  $\alpha = 1$ :

$$\langle \nabla \cdot \mathbf{A}(\mathbf{x}_i) \rangle = \sum_{j \in N(\mathbf{x}_i)} (\mathbf{A}(\mathbf{x}_i) + \mathbf{A}(\mathbf{x}_j)) \cdot \nabla_i W_{ij} \frac{m_j}{\rho_j}. \quad (4-33)$$

**Regra SPH III para o gradiente.** Seja  $A(\mathbf{x})$  um campo escalar e  $\alpha = 2$ :

$$\left\langle \frac{\nabla A(\mathbf{x}_i)}{\rho_i} \right\rangle = \sum_{j \in N(\mathbf{x}_i)} \left( \frac{A(\mathbf{x}_i)}{\rho_i^2} + \frac{A(\mathbf{x}_j)}{\rho_j^2} \right) \nabla_i W_{ij} m_j. \quad (4-34)$$

**Regra SPH III para o divergente.** Seja  $\mathbf{A}(\mathbf{x})$  um campo vetorial e  $\alpha = 2$ :

$$\left\langle \frac{\nabla \cdot \mathbf{A}(\mathbf{x}_i)}{\rho_i} \right\rangle = \sum_{j \in N(\mathbf{x}_i)} \left( \frac{\mathbf{A}(\mathbf{x}_i)}{\rho_i^2} + \frac{\mathbf{A}(\mathbf{x}_j)}{\rho_j^2} \right) \cdot \nabla_i W_{ij} m_j. \quad (4-35)$$

## 4.2

### Equações de Navier–Stokes usando SPH

Nessa seção mostraremos com detalhes a aplicação do método SPH em dinâmica dos fluidos. A abordagem SPH consiste em discretizar um fluido através de um sistema de partículas onde as derivadas espaciais das equações de Navier–Stokes na forma lagrangeana (2-7) – (2-8) são calculadas em cada partícula utilizando as regras definidas na Seção 4.1. Além das partículas amostradas serem o centro de interpolação, cada partícula representa um

elemento de fluido, ou seja, possuem atributos do fluido como por exemplo, massa, posição, velocidade, viscosidade, pressão, etc.

#### 4.2.1

##### Aproximação SPH da equação da continuidade

Geralmente em sistemas SPH a aproximação da densidade numa partícula  $i$  é feita utilizando a equação (4-12):

$$\rho_i = \sum_{j \in N(\mathbf{x}_i)} \rho_j W_{ij} \frac{m_j}{\rho_j} = \sum_{j \in N(\mathbf{x}_i)} m_j W_{ij}. \quad (4-36)$$

Entretanto, a aproximação (4-36) requer um ciclo extra de computação, pois a densidade é calculada antes dos outros parâmetros, tornando assim a simulação mais lenta no ponto de vista computacional. Outra opção para o cálculo da densidade é através da equação da continuidade (2-7), simplesmente aplicando a aproximação (4-29) em (2-7). Seja  $\mathbf{v}_i$  a velocidade de uma partícula  $i$ , assim a versão SPH da equação de continuidade é da forma:

$$\frac{d\rho_i}{dt} = -\rho_i \sum_{j \in N(\mathbf{x}_i)} (\mathbf{v}_j - \mathbf{v}_i) \cdot \nabla_i W_{ij} \frac{m_j}{\rho_j}. \quad (4-37)$$

A densidade da partícula  $i$  é obtida integrando (4-37) em relação ao tempo.

#### 4.2.2

##### Aproximação SPH da equação de momento

A aceleração em cada partícula  $i$  é dada pela equação de momento (2-8)

$$\frac{d\mathbf{v}_i}{dt} = -\frac{1}{\rho_i} \nabla p_i + \frac{1}{\rho_i} \nabla \cdot \mathbf{S}_i + \mathbf{g}, \quad (4-38)$$

semelhante a aproximação da equação da continuidade (4-37), iremos aplicar as regras de aproximação SPH em cada termo da equação (4-38).

**Pressão.** No método SPH ao contrário dos métodos com malha, onde a pressão de um fluido incompressível é a solução implícita de uma equação de Poisson (3-2), a pressão é uma função explícita da densidade local do fluido.

Apesar do método SPH ser um método criado para simular fluidos compressíveis, podemos aproximar um fluido incompressível através de um fluido quasi-compressível utilizando uma equação de estado [47]. Assim a pressão  $p_i$  pode ser calculada por uma equação de estado sugerida por Batchelor [7] e tem a forma

$$p_i = B \left( \left( \frac{\rho_i}{\rho_0} \right)^\gamma - 1 \right), \quad (4-39)$$

com  $\gamma = 7$  e  $\rho_0$  é a densidade (ou massa específica) de referência. O parâmetro  $B$  é o termo relacionado às flutuações de densidade do fluido

e é estimado por

$$B = \frac{c^2 \rho_0}{\gamma}, \quad (4-40)$$

onde  $c$  é a velocidade do som. A velocidade do som representa a velocidade pontual mais rápida de propagação de onda naquele meio, nas simulações SPH ela é escolhida como sendo aproximadamente dez vezes a maior velocidade esperada no escoamento do fluido, isto é, um número de Mach em torno de 0.1. A equação (4-39) é conhecida como *equação de Tait*. Morris *et al.* [50] simularam fluidos incompressíveis com baixo número de Reynolds através da seguinte equação de estado

$$p_i = c^2 (\rho_i - \rho_0). \quad (4-41)$$

A equação (4-41) é muito semelhante à equação de estado de um gás ideal usada por Desbrun & Cani para simular objetos deformáveis [17]. O cálculo da pressão ainda continua sendo um ponto delicado nas simulações de fluidos incompressíveis usando SPH, pois há uma dificuldade de manter a incompressibilidade do fluido devido à falta de um controle explícito da densidade global.

Após atualizar a pressão em todas as partículas usando uma das equações de estado acima, podemos avaliar o gradiente da pressão na equação (4-38) em cada partícula. Segue pela aproximação (4-34) que:

$$-\frac{1}{\rho_i} \nabla p_i = - \sum_{j \in N(\mathbf{x}_i)} m_j \left( \frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \nabla_i W_{ij}. \quad (4-42)$$

**Forças viscosas.** A fim de calcular as forças viscosas, precisamos primeiro calcular o tensor campo de velocidade  $\nabla \mathbf{v}_i$  em cada partícula  $i$ , pela aproximação (4-28) temos:

$$\nabla \mathbf{v}_i = \sum_{j \in N(\mathbf{x}_i)} \frac{m_j}{\rho_j} (\mathbf{v}_j - \mathbf{v}_i) \otimes \nabla_i W_{ij}. \quad (4-43)$$

Após o cálculo da velocidade, o tensor extra-tensão é calculado da seguinte forma

$$\mathbf{S}_i = \mu_i \mathbf{D}_i, \quad (4-44)$$

onde  $\mu_i$  é a viscosidade e  $\mathbf{D}_i$  é o tensor taxa de deformação com

$$\mathbf{D}_i = \nabla \mathbf{v}_i + (\nabla \mathbf{v}_i)^T. \quad (4-45)$$

Finalmente, após atualizar o tensor  $\mathbf{S}_i$ , o termo viscoso da equação (4-38) pode ser aproximado de acordo com a regra (4-33):

$$\frac{1}{\rho_i} \nabla \cdot \mathbf{S}_i = \sum_{j \in N(\mathbf{x}_i)} \frac{m_j}{\rho_i \rho_j} (\mathbf{S}_i + \mathbf{S}_j) \cdot \nabla_i W_{ij}. \quad (4-46)$$

## 5 Implementação SPH

No Capítulo 4, mostramos com detalhes a aproximação SPH das equações de Navier–Stokes. Nesse capítulo, discutiremos os aspectos computacionais de implementação de um simulador de fluidos baseado no método SPH, onde o ciclo de simulação de um sistema SPH segue cada uma das etapas mostradas na Figura 5.1. Essas etapas serão apresentadas nas seções desse capítulo.

Na Seção 5.1, apresentaremos algumas estruturas de dados de busca de pares de interação entre as partículas. Na Seção 5.2, abordaremos os esquemas de integração numérica para resolver as equações que descrevem a dinâmica do sistema. Nas Seções 5.3 e 5.4, mostraremos a forma em que as condições de fronteira do sistema são usualmente modeladas e a representação da superfície livre do fluido, respectivamente. E finalmente, apresentaremos na Seção 5.5 um exemplo de uma simulação SPH.

### 5.1 Busca de partículas vizinhas

No método SPH como todas as aproximações são realizadas através de convoluções discretas com núcleos que possuem suporte compacto de raio  $\kappa h$ , somente um número finito de partículas estarão dentro da área de influência de uma partícula dada. Essas partículas são conhecidas como sendo *partículas vizinhas* de uma determinada partícula. Ao contrário dos métodos eulerianos,

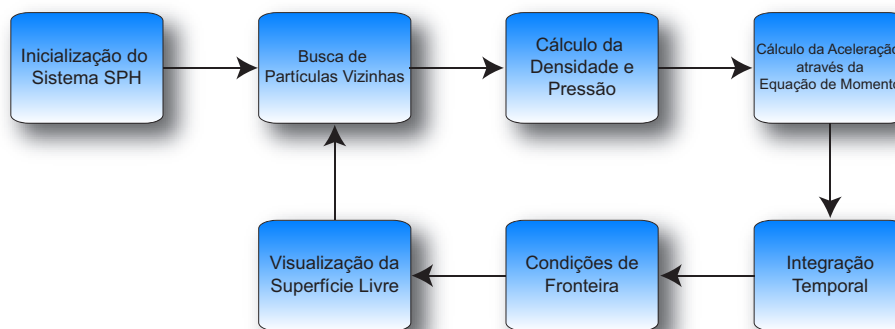


Figura 5.1: Uma visão geral do ciclo de simulação de um sistema SPH.

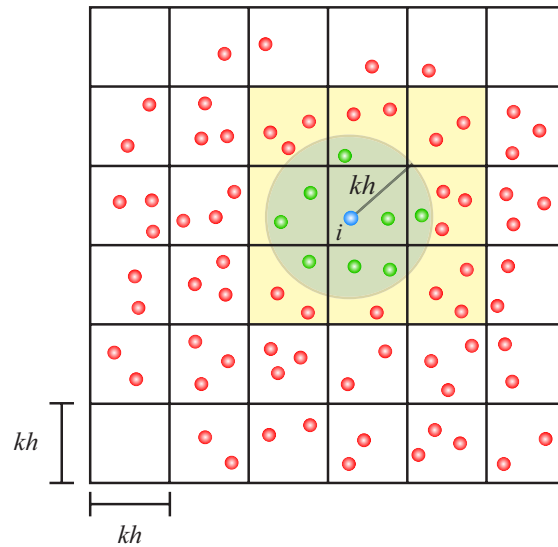


Figura 5.2: Busca das partículas vizinhas utilizando um grid uniforme bidimensional com o espaçamento sendo o raio de influência  $\kappa h$ . A região amarela representa as células do grid onde será realizada a busca.

onde as posições das células vizinhas do grid são bem definidas, no SPH as partículas vizinhas geralmente variaram com o tempo.

A busca de partículas vizinhas em SPH é uma tarefa árdua e requer estruturas de dados eficientes que possam realizar consultas de partículas que distam entre si uma distância menor do que o raio de influência do núcleo. Exemplos dessas estruturas de dados são: *grids uniformes*, *tabela hash*, *octrees*, *kd-trees* e *bsp-trees* [34]. Essas estruturas de dados se diferem na complexidade computacional de sua construção ou atualização, como também na consulta de vizinhança de uma partícula e no consumo de memória. Estruturas de dados não hierárquicas tais como grids podem ser construídas e atualizadas eficientemente, enquanto estruturas de dados hierárquicas como octrees, são geralmente mais caras de construir e de atualizar, mas podem ser bastante eficientes quando o comprimento suave do núcleo é variável.

### 5.1.1 Algoritmo baseado em grid

Esse algoritmo é recomendado para casos em que o núcleo possui comprimento suave constante. Nessa implementação um grid uniforme (células alinhadas com os eixos) é construído sobre o domínio do problema (Figura 5.2). Para núcleos que possuem raio de influência  $\kappa h$ , o espaçamento do grid também será  $\kappa h$  e uma célula pode ser identificada através do seu vértice mínimo, ou seja, o vértice que possui os valores mínimos nas três coordenadas. Cada célula do grid está associada a uma lista simplesmente encadeada contendo todas

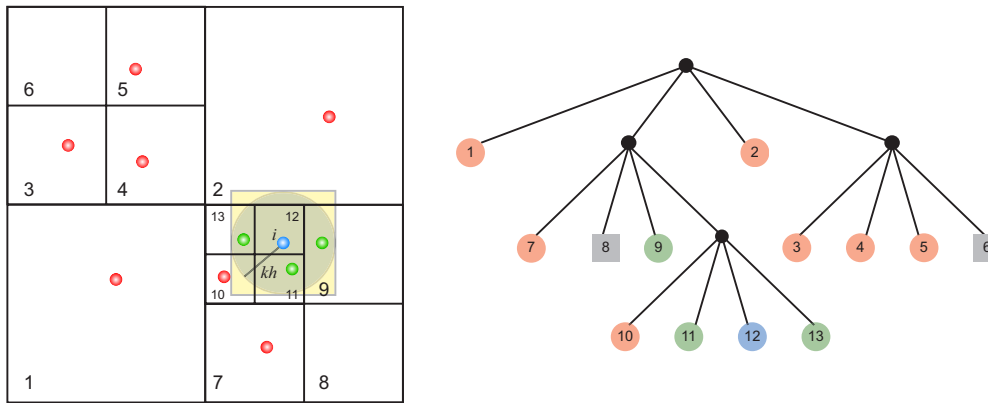


Figura 5.3: Estrutura de árvore utilizada no algoritmo de busca e a subdivisão hierárquica do espaço bidimensional. A busca é realizada através de um teste de intersecção do cubo envolvente da partícula  $i$  (região em amarelo) com as células representadas pelos nós da árvore.

as partículas que ocupam a porção de espaço delimitada pela própria célula. Dado um grid 3D, um ponto  $(x, y, z) \in \mathbb{R}^3$  é mapeado numa célula do grid com índice  $(I, J, K)$  da seguinte maneira

$$(I, J, K) = \left( \left\lfloor \frac{x}{kh} \right\rfloor, \left\lfloor \frac{y}{kh} \right\rfloor, \left\lfloor \frac{z}{kh} \right\rfloor \right), \quad (5-1)$$

onde  $\lfloor \cdot \rfloor$  representa a menor parte inteira. Através do mapeamento (5-1) podemos armazenar as partículas nas listas associadas às células do grid. Perceba que a construção do grid é muito eficiente, pois as partículas são inseridas nas listas em tempo constante  $\mathcal{O}(1)$ , resultando numa complexidade de construção e atualização de  $\mathcal{O}(N)$  sendo  $N$  o número de partículas do sistema.

Após a construção do grid, dada uma partícula  $i$  as suas partículas vizinhas só podem estar na mesma célula ocupada pela partícula  $i$  ou em suas células diretamente adjacentes. Assim, a busca por partículas que possuem uma distância menor do que  $\kappa h$  a partir da partícula  $i$  é restrita a  $3^d$  células, onde  $d$  é a dimensão do espaço. Se a média do número de partículas por células for suficientemente pequena, a complexidade do algoritmo baseado em grid uniforme é  $\mathcal{O}(N)$ . O algoritmo se torna menos eficiente quando o comprimento suave varia em relação ao tempo, logo o espaçamento do grid pode não ser ótimo para todas as partículas.

### 5.1.2

#### Algoritmo baseado em octree

Esse algoritmo funciona muito bem em simulações onde o comprimento suave do núcleo varia entre cada partícula do sistema. A estratégia do algoritmo é a de criar uma árvore octária (octree) de acordo com a posição das partículas. A octree subdivide recursivamente o domínio do problema em octantes que contenham pelo menos uma partículas. Cada octante corresponde a um nó da árvore, a recursão termina quando cada folha da árvore possuir apenas uma partícula (Figura 5.3). Se a árvore estiver balanceada, a complexidade da construção e atualização da octree é na média  $\mathcal{O}(N \log N)$  e no pior caso  $\mathcal{O}(N^2)$ , sendo  $N$  o número de partículas do sistema. Após a octree ser gerada, podemos usar essa estrutura para realizar a busca de partículas vizinhas.

Dada uma partícula  $i$ , a busca por suas partículas vizinhas é realizada através de um cubo centrado na posição da partícula  $i$  de lado  $2\kappa h_i$  e que envolve a própria partícula. Em cada nível da árvore é feito o teste de quais células da octree são intersectadas pelo cubo envolvente da partícula. Se caso o cubo não intersectar uma determinada célula da octree, a descida ao próximo nível da octree no caminho determinado por essa célula é interrompida. Se o cubo intersectar a célula, o processo de descida ao próximo nível da octree continua recursivamente até encontrar uma folha da árvore que contenha apenas uma partícula. Finalmente, é verificada se essa partícula está dentro do suporte compacto da partícula  $i$ . Caso esteja, ela é marcada como sendo uma partícula vizinha. A complexidade da busca baseada em octree é na média  $\mathcal{O}(N \log N)$  e  $\mathcal{O}(N^2)$  no pior caso.

## 5.2

### Integração numérica temporal

Após o cálculo da versão SPH das equações de Navier–Stokes (Seção 4.2), para atualizar a posição, velocidade e outras propriedades das partículas num certo instante de tempo, precisamos integrar a equação de momento (4-38) em relação ao tempo. Existem vários métodos de integração de equações diferenciais ordinárias, veja o livro de Eberly [18]. Dependendo de como são calculadas as variáveis desconhecidas do problema, os esquemas de integração podem ser classificados em *métodos implícitos* e *métodos explícitos*. Nos métodos implícitos as variáveis desconhecidas são dadas implicitamente como a solução de um sistema de equações envolvendo o estado atual e o desconhecido do sistema, enquanto que nos métodos explícitos elas são calculadas considerando apenas o estado atual. As vantagens dos métodos explícitos são: podem ser implementados facilmente, podem ser calculados de maneira rápida e eficiente.



Porém a estabilidade do método depende de como é feita a escolha do passo de tempo  $\Delta t$ , para que haja a convergência o passo de tempo  $\Delta t$  deve obedecer ao critério de estabilidade numérica conhecido como *condição Courant–Friedrichs–Lewy* (CFL) [14]. Já os métodos implícitos são estáveis em passos de tempo arbitrários, mas para isso requer resolver sistemas algébricos que são na sua maioria não-lineares. Nessa tese, optamos em usar métodos explícitos como *método de Euler* e o *integrador leap–frog* em nossa implementação.

### 5.2.1

#### Método de Euler

O método explícito mais simples é conhecido como método de Euler. Nesse método, a velocidade  $\mathbf{v}_i$  e a posição  $\mathbf{x}_i$  são atualizadas paralelamente da seguinte maneira:

$$\begin{aligned} \mathbf{v}_i(t + \Delta t) &= \mathbf{v}_i(t) + \Delta t \mathbf{a}_i(t) \\ \mathbf{x}_i(t + \Delta t) &= \mathbf{x}_i(t) + \Delta t \mathbf{v}_i(t + \Delta t), \end{aligned} \quad (5-2)$$

onde  $\mathbf{a}_i$  é aceleração da partícula num instante de tempo  $t$ , obtida a partir da equação (4-38).

Infelizmente, o método de Euler possui uma precisão de primeira ordem e pode se tornar bastante instável em certas circunstâncias – uma propriedade indesejável para qualquer integrador. Essa instabilidade pode ser remediada tomando um passo de tempo  $\Delta t$  muito pequeno.

### 5.2.2

#### Integrador leap–frog

Um integrador mais atraente e tão simples quanto o método de Euler, mas com uma precisão de segunda ordem, é o integrador leap–frog [18]. Nesse esquema de integração a velocidade de cada partícula é calculada nos pontos médio dos intervalos de tempo (Figura 5.4). Assim, posição e velocidade são avaliadas intercaladamente da seguinte forma:

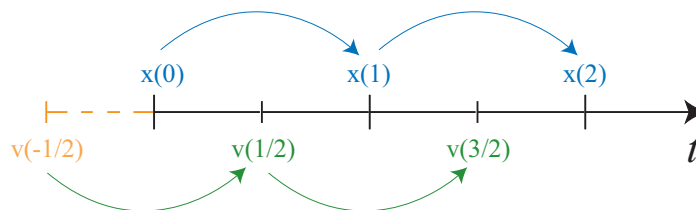


Figura 5.4: Esquema de integração leap–frog: posição  $\mathbf{x}_i$  e velocidade  $\mathbf{v}_i$  numa partícula  $i$  são avaliadas de maneira intercalada em relação ao tempo  $t$ .

$$\begin{aligned} \mathbf{v}_i \left( t + \frac{1}{2} \Delta t \right) &= \mathbf{v}_i \left( t - \frac{1}{2} \Delta t \right) + \Delta t \mathbf{a}_i(t) \\ \mathbf{x}_i \left( t + \Delta t \right) &= \mathbf{x}_i \left( t \right) + \Delta t \mathbf{v}_i \left( t + \frac{1}{2} \Delta t \right). \end{aligned} \quad (5-3)$$

Note que a velocidade no tempo  $t$  não é dada de forma explícita, logo ela é calculada através da média entre a velocidade anterior e posterior

$$\mathbf{v}_i(t) = \frac{1}{2} \left[ \mathbf{v}_i \left( t + \frac{1}{2} \Delta t \right) + \mathbf{v}_i \left( t - \frac{1}{2} \Delta t \right) \right]. \quad (5-4)$$

Além disso, a inicialização do método requer uma etapa adicional no cálculo da velocidade  $\mathbf{v}_i \left( -\frac{1}{2} \right)$  através do método de Euler

$$\mathbf{v}_i \left( -\frac{1}{2} \right) = \mathbf{v}_i(0) - \frac{1}{2} \Delta t \mathbf{a}_i(0). \quad (5-5)$$

O integrador leap-frog é reversível no tempo devido à forma simétrica na qual ele é definido. Métodos explícitos clássicos como o método Euler, método do ponto-médio e os métodos de Runge-Kutta não possuem essa propriedade, pois todos eles avaliam as derivadas de maneira assimétrica. A reversibilidade no tempo é uma propriedade importante, pois ela garante conservação de energia do sistema.

### 5.3 Condição de fronteira

O tratamento de fronteiras sólidas no SPH foi proposto por Monaghan em [47], a sua idéia era a de representar as fronteiras do domínio através de partículas virtuais conhecidas como *partículas fantasmas* (veja Figura 5.5). O objetivo das partículas fantasmas é gerar forças altamente repulsivas e assim prevenir a interpenetração de partículas do fluido nas fronteiras sólidas do problema. Essa força de repulsão é calculada usando uma expressão matemática

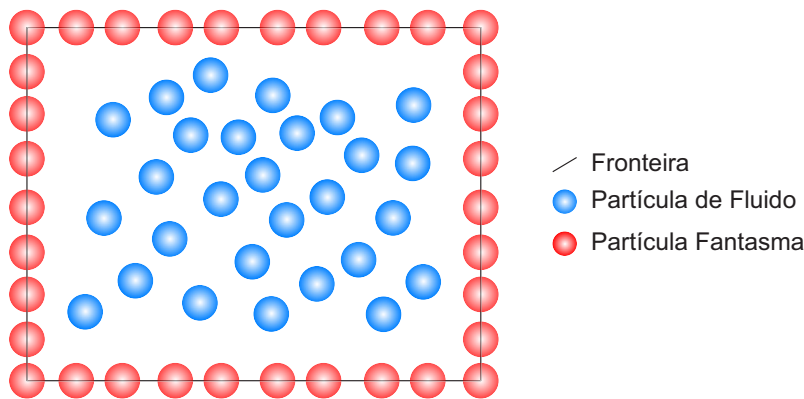


Figura 5.5: As partículas fantasmas (vermelho) são utilizadas para simular as condições de fronteira sólidas, evitando que as partículas de fluido (azul) escapem do domínio do problema.

semelhante à do potencial de Lennard–Jones utilizado em dinâmica molecular [4]. Portanto, a força de repulsão sobre uma partícula de fluido  $i$  que colide com uma partícula fantasma  $g$  é

$$\Gamma_{ig} = \begin{cases} D \left[ \left( \frac{r_0}{r_{ig}} \right)^{12} - \left( \frac{r_0}{r_{ig}} \right)^4 \right] \frac{\mathbf{x}_{ig}}{r_{ig}^2}, & \frac{r_0}{r_{ig}} \leq 1 \\ 0, & \frac{r_0}{r_{ig}} > 1 \end{cases}, \quad (5-6)$$

com  $\mathbf{x}_{ig} = \mathbf{x}_i - \mathbf{x}_g$ ,  $r_{ig} = \|\mathbf{x}_i - \mathbf{x}_g\|$ . A constante  $D$  depende do problema e deve ser da mesma ordem de grandeza do quadrado da maior velocidade esperada no escoamento. O raio de interação  $r_0$  é importante na simulação. Pois se for escolhido um valor muito alto para  $r_0$ , a força de repulsão pode causar uma grande perturbação no estado inicial das partículas e conseqüentemente arruinar toda a simulação. Se for escolhido um valor pequeno, a força de repulsão não será suficiente para evitar a interpenetração de partículas na fronteira. Na maioria dos casos, é recomendado  $r_0$  ser um valor próximo da distância inicial das partículas do sistema.

Uma das maiores dificuldades da abordagem acima é a de modelar fronteiras que possuem geometria complexa através de partículas fantasmas. Para simular a interação entre corpos deformáveis com fluidos utilizando SPH, Müller *et al.* em [53] utilizaram quadratura Gaussiana para gerar partículas fantasmas sobre a malha triangular de um objeto deformável. No Capítulo 6, mostraremos uma nova abordagem no tratamento de fronteiras sólidas de natureza puramente geométrica.

#### 5.4 Representação implícita da superfície livre

Uma superfície implícita de um *isovalor*  $c$  é dada como  $S = f^{-1}(c) = \{\mathbf{x} \in \mathbb{R}^3 : f(\mathbf{x}) = c\}$  onde  $f : \mathbb{R}^3 \rightarrow \mathbb{R}$  é uma função escalar. A função  $f$  é definida tal que um ponto  $\mathbf{x}$  está no interior da superfície  $S$  temos  $f(\mathbf{x}) - c < 0$  e quando  $\mathbf{x}$  está no exterior de  $S$  temos  $f(\mathbf{x}) - c > 0$ . Logo, para verificar se um ponto  $\mathbf{x}$  está dentro ou fora de uma superfície  $S$  basta observar o sinal de  $f(\mathbf{x})$ . Por outro lado, fazer amostragem de pontos sobre uma superfície é um processo laborioso, pois se torna um problema de achar raízes de uma função arbitrária. Assim a visualização de superfícies implícitas se torna mais difícil do que a visualização de superfícies paramétricas.

Geralmente, a função  $f : \mathbb{R}^3 \rightarrow \mathbb{R}$  é definida como sendo a *função distância com sinal* da superfície  $S$ , daí a amostragem de um campo distância se torna uma tarefa fácil. A visualização de superfícies implícitas é feita tradicionalmente com malhas triangulares obtidas através do algoritmo de *marching cubes* (MC) [40]. No método MC a função implícita é armazenada

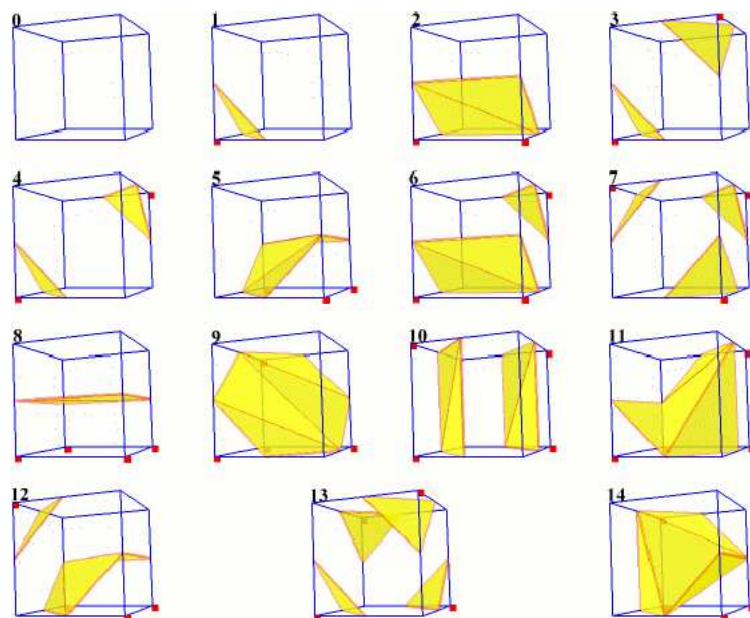


Figura 5.6: Tabela do algoritmo de Marching Cubes (MC) que codifica a triangulação de cada cubo do grid dependendo da configuração dos sinais em seus vértices.

num campo escalar amostrado num grid tridimensional. O MC classifica os vértices do grid como positivo ou negativo, de acordo com o isovalor dado. Após cada vértice ser classificado, o algoritmo faz uma varredura nos cubos do grid verificando a configuração dos sinais dos vértices de cada cubo. Se um cubo possuir mudança de sinal, parte da triangulação da superfície está no seu interior e é dada através de uma tabela pré-definida de acordo com sua configuração de sinais, veja a Figura 5.6. Finalmente, as posições dos vértices dos triângulos no interior do cubo são obtidas através de uma interpolação tri-linear.

Entretanto, visualizar superfícies com singularidades como pontas e quinas, requer informações adicionais tais como o gradiente da superfície (normal da superfície). O algoritmo *dual marching cubes* (DMC) [64] é uma extensão do algoritmo de MC devido ao uso da informação do gradiente para detectar e reconstruir singularidades através de um grid adaptativo. A vantagem das superfícies implícitas é a sua consistência no ponto de vista de não possuir auto-intersecções e a de lidar intrinsecamente com as mudanças topológicas. Entretanto, o algoritmo de MC possui ambigüidades topológicas que podem aparecer nas faces ou no interior do cubo. Essas ambigüidades podem gerar mudanças topológicas incorretas, por exemplo, duas componentes conexas de uma superfície podem se unir pelo fato de estarem próximas e formar apenas uma única componente (Figura 5.7).

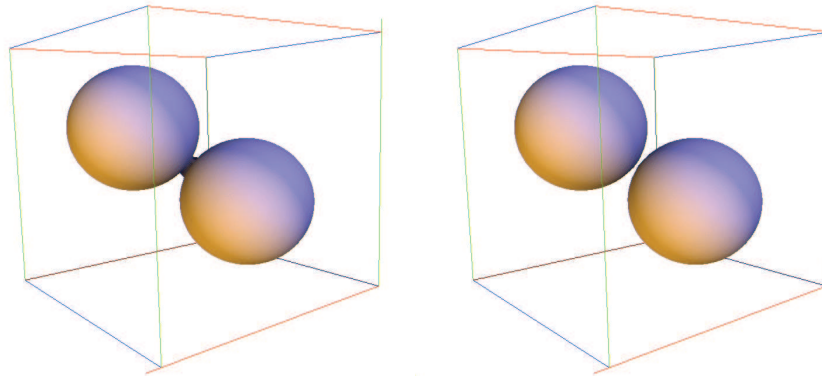


Figura 5.7: Superfície implícita de duas esferas próximas uma da outra. À esquerda a superfície é gerada através do algoritmo de MC original. À direita a superfície é gerada utilizando o algoritmo de MC com garantias topológicas.

A captura da superfície livre do fluido é dada implicitamente através da aproximação SPH da função característica dada por Morris em [49]:

$$\chi(\mathbf{x}) = \sum_{j \in N(\mathbf{x})} \frac{m_j}{\rho_j} W(\mathbf{x} - \mathbf{x}_j, h), \quad (5-7)$$

o isovalor é tomado no intervalo  $[0, 1]$ , isso se deve ao fato do núcleo  $W$  possuir suporte compacto e ser partição da unidade. A geração da malha triangular da superfície livre do fluido é feita através de uma implementação eficiente do MC e com garantias topológicas [36]. Usamos a mesma estrutura de dados utilizada na busca de partículas vizinhas (Seção 5.1) para acelerar a avaliação da função

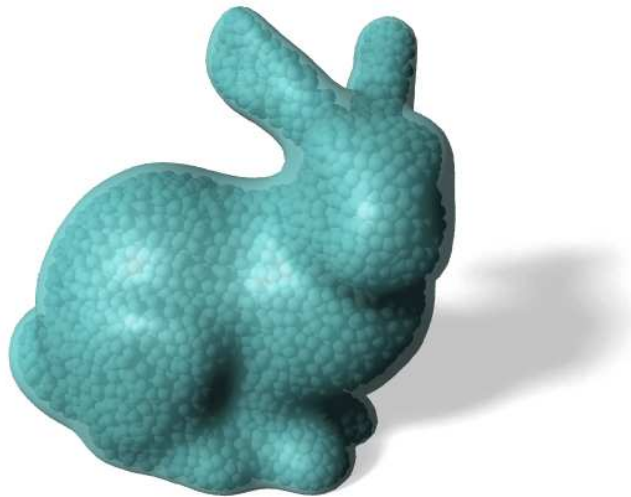


Figura 5.8: Visualização da superfície implícita extraída do modelo Stanford Bunny constituído de partículas.

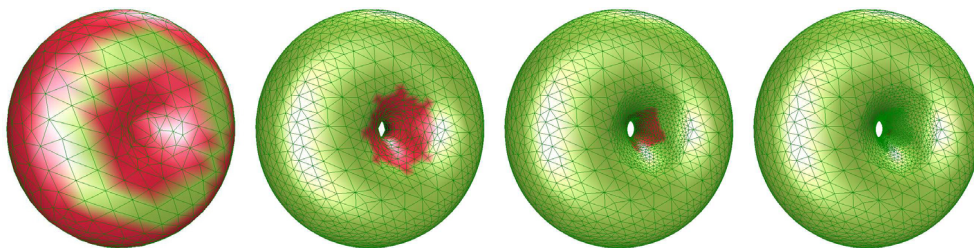


Figura 5.9: Triangulação adaptativa da isosuperfície do torus utilizando um algoritmo robusto de DMC. As regiões em verde representam as regiões da superfície que possuem garantias topológicas e geométricas, enquanto as regiões com ambigüidades (vermelho) são resolvidas com um número pequeno de refinamentos.

característica (5-7) no grid. A Figura 5.8 mostra a superfície livre do fluido gerada a partir de modelo volumétrico formado por partículas. Uma classe especial de superfícies implícitas usadas na representação da superfície livre de fluidos são as *level-sets* [56]. A superfície level-set  $F(\mathbf{x}, t)$  acompanha a evolução da superfície livre do fluido ao longo de sua normal de acordo com um campo escalar de velocidade  $\mathbf{v}(\mathbf{x}, t)$  e é representada através de uma equação diferencial parcial

$$\frac{\partial F}{\partial t} = -\mathbf{v} \cdot \|\nabla F\|. \quad (5-8)$$

Paiva *et al.* [58] criaram um algoritmo robusto de DMC com garantias topológicas utilizando ferramentas numéricas auto-validadas como aritmética intervalar e diferenciação automática (Figura 5.9). Um grande desafio na visualização de superfície livre dada de forma implícita é a síntese de textura dinâmica, devido ao fato de geralmente a superfície não possuir uma parametrização explícita. Todas as animações apresentadas nessa tese foram geradas utilizando o *ray-tracer* de código aberto POV-Ray (<http://www.povray.org>).

## 5.5

### Exemplo numérico

Um exemplo clássico da aplicação do método SPH em superfície livre é a simulação da quebra de uma barragem [47]. O problema consiste num volume de água confinado numa região cúbica de um canal onde é instantaneamente retirada a barragem da represa. O método SPH é utilizado para simular o comportamento da queda da coluna de água através das equações de Navier-Stokes (4-37) – (4-38). O problema é esquematizado na Figura 5.10.

**Inicialização numérica do problema.** O sistema é iniciado com densidade inicial da água sendo  $\rho_0 = 1000\text{kg/m}^3$ , a viscosidade  $\mu = 10^{-3}\text{Ns/m}^2$  e a gravidade  $\mathbf{g} = 9.8\text{m/s}^2$ . A velocidade característica do escoamento  $\mathbf{v}_e$  é estimada

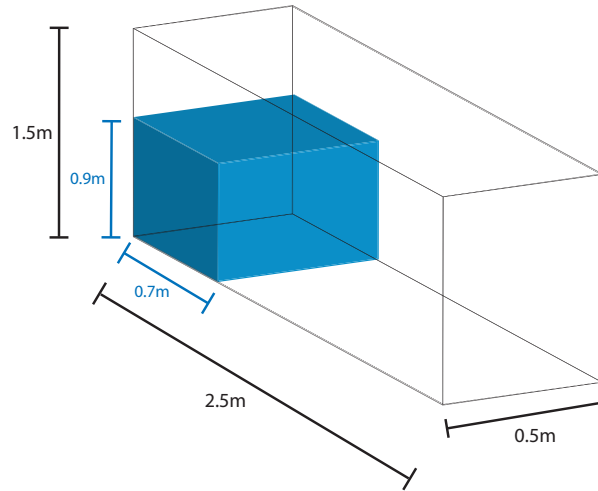


Figura 5.10: Configuração inicial do problema.

a partir da conservação de energia mecânica do sistema (transformação de energia potencial em cinética), segue que

$$\frac{m\mathbf{v}_e^2}{2} = mgH, \quad (5-9)$$

como a coluna de água possui altura  $H = 0.9\text{m}$ , logo  $\mathbf{v}_e = \sqrt{2gH} = 4.2\text{m/s}$ . Assim, podemos estimar a velocidade do som  $c \approx 42\text{m/s}$  e obter a constante  $B = 252\text{kPa}$  da equação (4-39).

Dado o volume total do fluido  $V$ , a massa de uma partícula  $i$  é determinada da forma

$$m_i = \rho_0 \frac{V}{N}, \quad (5-10)$$

sendo  $N$  o número total de partículas do sistema. Cada partícula do sistema é representada por uma esfera, logo o raio de uma partícula  $i$  é

$$r_i = \sqrt[3]{\frac{3}{4} \frac{m_i}{\rho_0 \pi}}. \quad (5-11)$$

Na simulação do problema utilizamos  $10^4$  partículas (Figura 5.11), portanto a massa de cada partícula vale  $m_i = 3.15 \times 10^{-2}\text{kg}$ . As partículas são inicialmente geradas num grid uniforme onde o raio de influência  $\kappa h$  é tomado como sendo 1.2 vezes o tamanho do espaçamento do grid, recomenda-se que em simulações SPH que o número de partículas vizinhas seja no mínimo 20 em simulações 2D e 56 em simulações 3D [38]. O critério de estabilidade numérica da integração temporal respeita a condição CFL

$$\Delta t = 0.1 \min_i \left\{ \frac{h}{\|\mathbf{v}_i\| + c} \right\}. \quad (5-12)$$

A atualização do estado de um sistema SPH convencional é realizada seguindo o Algoritmo 1. Em todas as simulações dessa tese, adotamos o

comprimento suave  $h$  constante, logo a busca de partículas vizinhas é feita eficientemente utilizando um grid uniforme (Seção 5.1.2).

---

**Algoritmo 1** Sistema SPH convencional

---

```
1: repeat
2:   Faça a busca das partículas vizinhas. (Seção 5.1.2)
3:   for cada partícula  $i$  do
4:     Atualize a pressão  $p_i$ . (equação (4-39))
5:   end for
6:   for cada partícula  $i$  do
7:     Calcule a derivada da densidade. (equação (4-37))
8:     Calcule a aceleração. (equação (4-38))
9:   end for
10:  for cada partícula  $i$  do
11:    Atualize  $\mathbf{v}_i$  e  $\rho_i$  com o integrador leap-frog. (Seção 5.2.2)
12:    Aplique a condição de fronteira. (Seção 5.3)
13:  end for
14:  Atualize  $\Delta t$  usando a condição CFL. (equação (5-12))
15:   $tempo = tempo + \Delta t$ 
16: until  $tempo < tempo_{total}$ 
```

---





Figura 5.11: Simulação SPH da quebra de uma barragem utilizando  $10^4$  partículas.

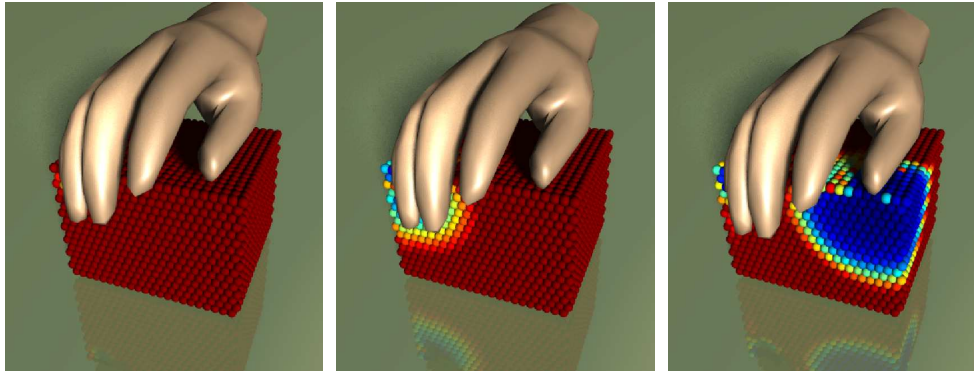
## 6 Fluidos viscoplásticos

Nesse capítulo, apresentaremos um novo método de animação visualmente realística de fenômenos físicos como deformações plásticas e mudanças de fase sólida-líquida que ocorrem em materiais viscoplásticos, tais como: metal, plástico, cera, polímero, argila e lava. Essa técnica consiste na representação desses materiais como um fluido não-newtoniano que varia entre o estado sólido e líquido, e de alta para baixa viscosidade dependendo de sua própria temperatura ou de uma força externa aplicada sobre o material.

O primeiro objetivo desse capítulo é o de simular com fidelidade os efeitos viscosos de um fluido viscoplástico, onde a viscosidade de um fluido descreve a maneira que a velocidade do fluido se dissipa. Assim, criamos um modelo SPH baseado na formulação física de Fluido Newtoniano Generalizado proposta por Mendes *et al.* [45] (veja Seção 6.1). Nessa formulação, a viscosidade do material é dada através de uma função não-linear em termos da intensidade de uma tensão de deformação aplicada sobre ele.

O próximo objetivo consiste na simulação de materiais que variam entre o estado sólido e líquido devido a variações térmicas. Para simular essas transições criamos uma simples dependência da viscosidade com a temperatura do material. Na Seção 6.2, mostraremos com detalhes como é feita essa transição, assim como uma nova aproximação SPH da equação do calor.

Mais adiante, na Seção 6.3, apresentaremos uma forma geométrica de impor as condições de fronteira do fluido. Em seguida, na Seção 6.4, discutiremos alguns aspectos numéricos a fim de garantir a estabilidade do sistema e apresentaremos uma nova variante do SPH conhecida como XSPH. E finalmente nas Seções 6.5 e 6.6, mostraremos a implementação e os resultados do nosso método, respectivamente.



6.1(a): 25 iterações

6.1(b): 890 iterações

6.1(c): 1225 iterações

Figura 6.1: Modelo de Fluido Newtoniano Generalizado: o mapa de cores representa a viscosidade de cada partícula, variando de baixa (azul) para alta viscosidade (vermelho). Note o salto da viscosidade criado pela força aplicada pela mão, quanto maior a força menor será a viscosidade do fluido.

### 6.1 Fluido Newtoniano Generalizado

No modelo físico de Fluido Newtoniano Generalizado, para cada partícula  $i$  do sistema, o tensor extra-tensão  $\mathbf{S}_i$  é representado matematicamente da seguinte forma:

$$\mathbf{S}_i = \eta(D_i) \mathbf{D}_i, \quad \text{com} \quad D_i = \sqrt{\frac{1}{2} \text{traço}(\mathbf{D}_i)^2} \quad \text{e}$$

$$\eta(D_i) = (1 - \exp[-(J + 1) D_i]) \left( D_i^{n-1} + \frac{1}{D_i} \right), \quad (6-1)$$

onde o tensor taxa de deformação  $\mathbf{D}_i$  é dado pela equação (4-45).

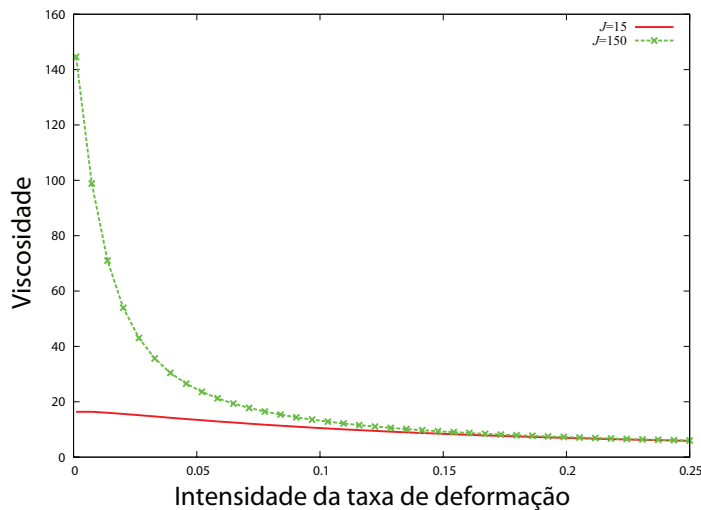


Figura 6.2: Comportamento viscoplástico do fluido.

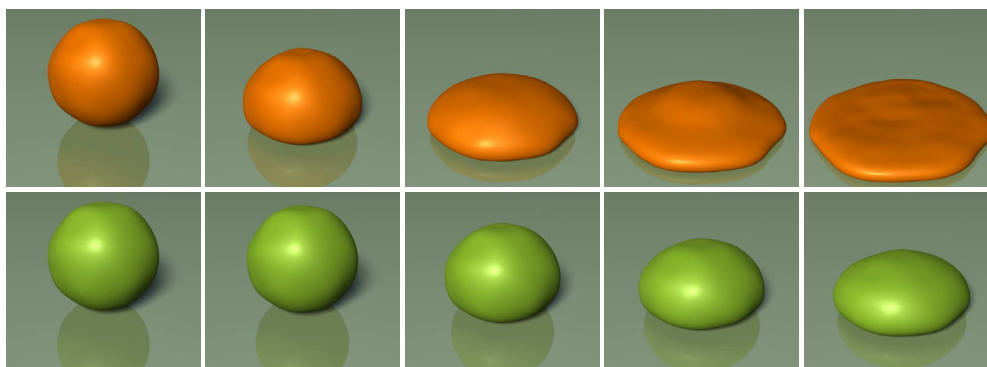


Figura 6.3: Variações da viscosidade de um fluido viscoplástico inicializado numa esfera com 1200 partículas. Acima, reduzimos a viscosidade do fluido usando  $J = 15$  e abaixo aumentamos a viscosidade com  $J = 150$ .

Essa formulação modela a viscosidade  $\eta$  como uma função não-linear em termos da intensidade da taxa de deformação  $D_i$  (veja Figura 6.2), sendo inversamente proporcional à tensão de deformação aplicada ao material (veja Figura 6.1). A função de viscosidade  $\eta$  depende apenas de dois parâmetros reológicos: o índice de comportamento do escoamento  $n$  e o termo  $J$ , conhecido como *jump number*.

Para fluidos viscoplásticos, Mendes representou de forma concisa vários parâmetros reológicos, tais como a tensão limite do escoamento e o índice de consistência, através de  $J$ . A viscosidade do fluido é controlada diretamente pela constante  $J$ , ou seja, quanto maior for o valor de  $J$  maior será a viscosidade do fluido (veja Figura 6.3). Note que, através da Figura 6.2 quando  $J = 15$  o fluido possui um comportamento newtoniano. Para simular o comportamento viscoplástico de um fluido o valor de  $n$  deve ser entre 0 e 1, logo em nossas simulações fixamos  $n = \frac{1}{2}$ .

## 6.2

### Transição de fase

A simulação de objetos que derretem e solidificam é uma tarefa delicada, pois é necessária a variação da viscosidade durante a transição de fase, de acordo com as propriedades do material. Em particular, nesse capítulo concentraremos na transição induzida pela temperatura entre as fases sólida e líquida. Modelamos essa transição variando a viscosidade de acordo com a temperatura de cada partícula de fluido. A variação da temperatura em relação ao tempo é determinada pela *equação do calor*. Essa equação governa a difusão térmica de um material, transferindo energia térmica de uma região de alta temperatura para uma região de baixa temperatura. A equação do calor é descrita através da temperatura  $T$  e do coeficiente de difusão térmica  $k$ :

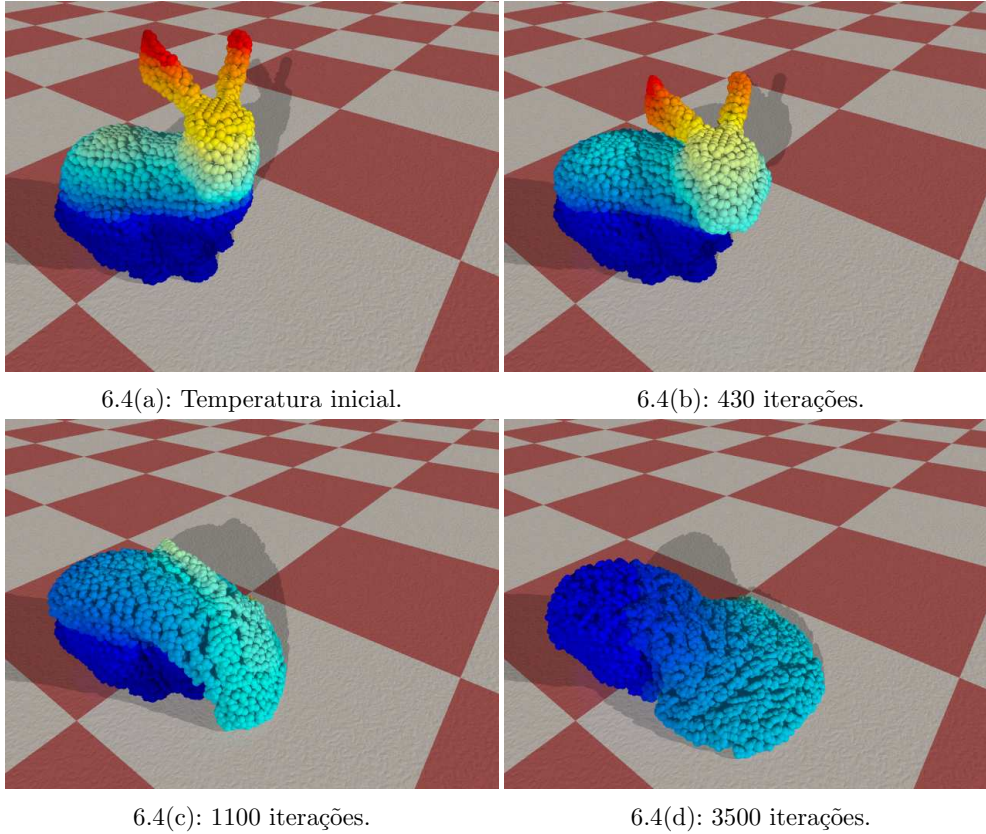


Figura 6.4: Temperatura das 10188 partículas do modelo Stanford Bunny: as regiões em azul escuro estão abaixo do ponto de fusão, e assim permanecem sólidas.

$$\frac{dT}{dt} = k\nabla^2 T. \quad (6-2)$$

No derretimento, a temperatura de algumas partes do objeto aumenta até alcançar o seu ponto de fusão no qual o objeto se torna líquido (veja Figura 6.4). Nos fluxos de lava, a temperatura pode decrescer abaixo do seu ponto de fusão fazendo com que a lava se solidifique e altere a topologia do terreno inicial (Figura 6.5). Em ambos os casos, o jump number  $J$  decresce com a temperatura. Aproximamos essa dependência através de uma combinação linear em termos da temperatura:

$$J(T) = (1 - u)J_{max} + uJ_{min}, \quad \text{com} \quad u = \frac{T - T_{min}}{T_{max} - T_{min}}. \quad (6-3)$$

Perceba que a função de viscosidade (6-1) decresce quando a temperatura aumenta e vice-versa. Em nossas simulações, assumimos que os objetos são homogêneos, isto é, possuem um coeficiente de difusão térmica constante.

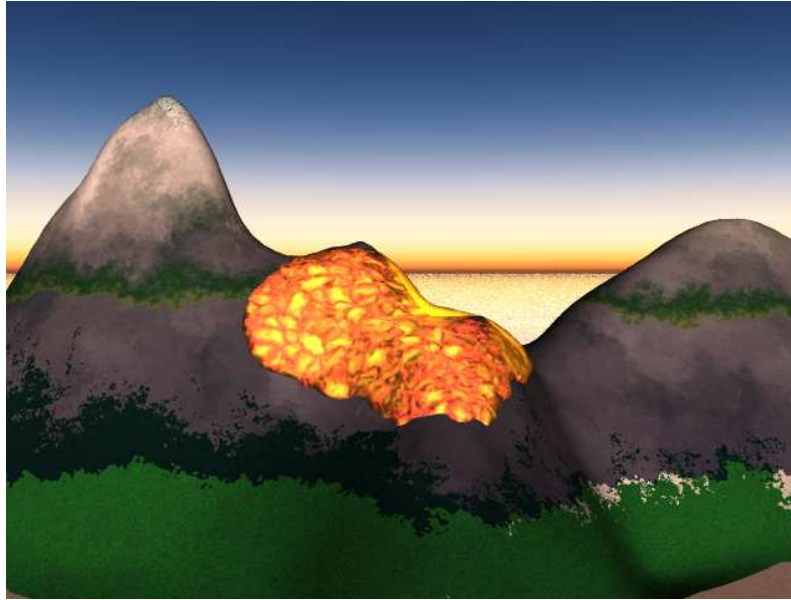


Figura 6.5: Escoamento de lava no Pão de Açúcar. A simulação da lava é feita utilizando 10137 partículas.

### 6.2.1 Aproximação SPH do operador laplaciano

A equação do calor (6-2), que governa a transição entre as fases sólida e líquida, requer uma aproximação por partículas do laplaciano da temperatura  $\nabla^2 T_i$ . As derivadas de segunda ordem podem ser aproximadas utilizando a convolução SPH usual através do laplaciano do núcleo para cada partícula  $i$ :

$$\nabla^2 T_i = \sum_{j \in N(\mathbf{x}_i)} \frac{m_j}{\rho_j} T_j \nabla_i^2 W_{ij}. \quad (6-4)$$

Porém, a equação (6-4) possui algumas desvantagens. Primeiro, essa aproximação é muito sensível à desordem de partículas. Segundo, a transferência de calor de uma partícula  $i$  para uma partícula  $j$  pode ser positiva ou negativa devido à mudança de sinal da derivada de segunda ordem do núcleo (veja Figura 4.1). Por outro lado, a física nos diz que uma partícula quente deve transferir calor para uma partícula fria sem se importar de que maneira é feita a separação delas. Outra desvantagem é que essa expressão não resulta em conservação de energia térmica num processo adiabático.

Para evitar esses problemas, Müller *et al.* [51] adicionaram em seu modelo um núcleo com derivada de segunda ordem positiva exclusivo para o operador laplaciano. Entretanto, além desse modelo ser errado do ponto de vista conceitual ele também introduz mais avaliações no sistema. Por essas razões, usamos uma aproximação do operador laplaciano proposta por Cleary & Monaghan [13] que envolve apenas derivadas de primeira ordem. Essa aproximação

é deduzida a partir da integral

$$I = \int (T(\mathbf{x}) - T(\mathbf{x}')) F(\mathbf{x} - \mathbf{x}') d\mathbf{x}', \quad (6-5)$$

tal que a função  $F$  é definida por

$$\mathbf{x}F(\mathbf{x}, h) = \nabla W(\mathbf{x}, h). \quad (6-6)$$

Expandindo o termo  $T(\mathbf{x}')$  no integrando de (6-5) em série de Taylor em torno de  $\mathbf{x}$ , até os termos de segunda ordem, temos

$$I = \nabla^2 T(\mathbf{x}) + r(h^2). \quad (6-7)$$

Portanto, tomando a forma SPH da integral (6-5) para uma partícula  $i$  e escolhendo

$$F(\mathbf{x}, h) = \frac{\mathbf{x} \cdot \nabla W(\mathbf{x}, h)}{\|\mathbf{x}\|^2}, \quad (6-8)$$

finalmente temos

$$\nabla^2 T_i = \sum_{j \in N(\mathbf{x}_i)} \frac{m_j}{\rho_j} (T_i - T_j) \frac{\mathbf{x}_{ij} \cdot \nabla_i W_{ij}}{|\mathbf{x}_{ij}|^2}. \quad (6-9)$$

Devido ao fato de  $\mathbf{x}_{ij} \cdot \nabla_i W_{ij} \leq 0$ , a expressão (6-9) tem a propriedade de que se  $T_i > T_j$ , então o fluxo de calor será realizado da partícula  $i$  para  $j$  e vice-versa.

### 6.3 Condição de fronteira

Em várias aplicações do método SPH, as fronteiras rígidas (terrenos ou paredes) são modeladas através do uso de partículas de fantasmas (Seção 5.3), as quais adicionam uma nova força interação nas partículas de fluido. Apesar de elegante, esse método aumenta consideravelmente o número de partículas do sistema, é computacionalmente caro, requer alto consumo de memória e a modelagem de geometrias complexas se torna difícil. Por essas razões,

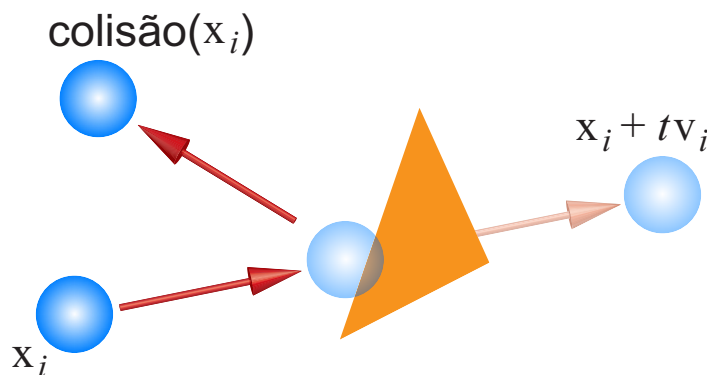
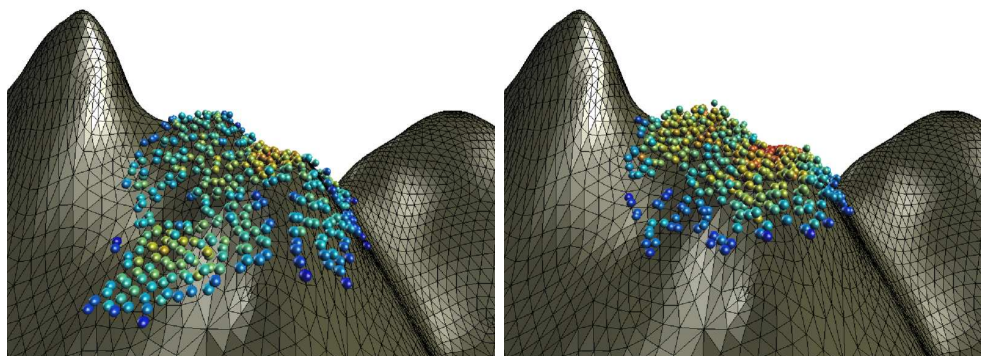


Figura 6.6: Resposta da colisão partícula  $\times$  triângulo.



6.7(a): Colisão suave.

6.7(b): Colisão não-eskorregadia.

Figura 6.7: Simulação de um escoamento de lava após 1910 iterações com colisão suave (esquerda) e com colisão não-eskorregadia (direita), com os mesmo parâmetros: 545 partículas e 9566 triângulos na fronteira. O mapa de cores representa a densidade de cada partícula. A colisão suave faz a lava deslizar mais rapidamente.

representamos explicitamente a fronteira do domínio utilizando uma malha triangular tal que as condições de fronteira são impostas através de um critério puramente geométrico, dado por um teste de colisão. O teste de colisão entre as partículas (esferas) e os triângulos orientados da fronteira é feito em duas etapas: detecção da colisão e a atualização do novo estado da partícula (resposta).

### 6.3.1

#### Resposta da colisão

A resposta do teste de colisão consiste no cálculo de uma nova posição e velocidade da partícula. Podemos inserir dissipação na velocidade de resposta simplesmente modificando a componente normal e tangencial da velocidade inspirada numa lei de Snell (Figura 6.6). Numa colisão perfeitamente elástica é invertida apenas a componente normal. Para colisões suaves, a componente tangencial e normal podem ser escaladas por coeficientes constantes entre 0 e 1, o coeficiente que dissipa a velocidade tangencial é chamado de *coeficiente de fricção* enquanto o que dissipa a velocidade normal é conhecido como *coeficiente de restituição*. Nos casos em que ambos os coeficientes são 0 a colisão é dita *não-eskorregadia* (Figura 6.7).

### 6.3.2

#### Detecção da colisão

Com objetivo de acelerar a detecção de uma eventual colisão entre as partículas do sistema com os triângulos da fronteira, armazenamos os triângulos no mesmo grid que utilizamos para fazer a busca de partículas



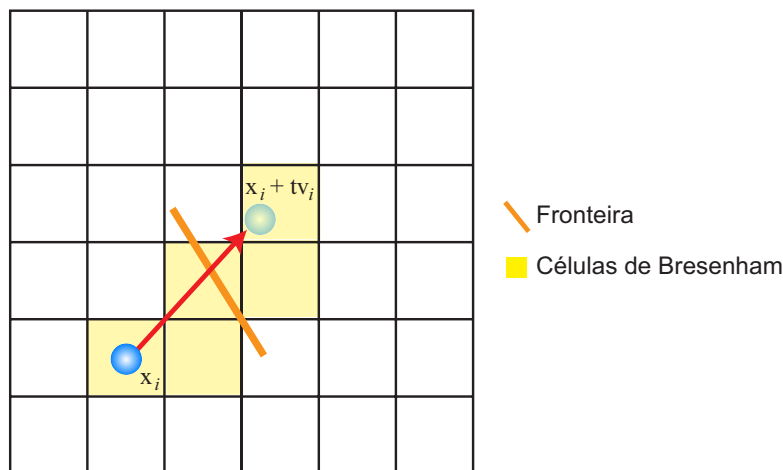


Figura 6.8: O teste de colisão realiza a busca de triângulos nas células do grid através do algoritmo de linhas de Bresenham.

vizinhas (Seção 5.1). Cada célula do grid de busca além de possuir uma lista de partículas vizinhas também possui uma lista de triângulos  $\mathcal{T}$ , para cada triângulo da fronteira fazemos o teste de intersecção com as células (cubos) do grid utilizando o algoritmo de Akenine–Möller [2]. Se o triângulo intersecar a célula logo ele é inserido na lista  $\mathcal{T}$  dessa célula. Apesar da possibilidade de um triângulo aparecer em várias células, esse armazenamento garante um tempo constante de busca para cada posição. Após a atualizar cada lista de triângulos  $\mathcal{T}$  nas células do grid, o teste de colisão é realizado de forma semelhante ao algoritmo de linhas de Bresenham [21] num grid 3D. O algoritmo de Bresenham seleciona incrementalmente algumas células do grid na tentativa de reproduzir a trajetória da partícula  $i$  através da linha que conecta os pontos  $\mathbf{x}_i$  e  $\mathbf{x}_i + \Delta t \mathbf{v}_i$ , o teste de colisão é feito somente nos triângulos das células selecionadas pelo algoritmo (Figura 6.8).

O teste de colisão entre partícula e triângulo é feito através de uma versão modificada de [29] e adaptada à nossa simulação. Para cada partícula  $i$  do sistema, utilizamos a informação da direção da velocidade  $\mathbf{v}_i$  para criar um eficiente teste de rejeição de triângulos através de uma simples verificação de sinal. Para cada triângulo  $T$  da lista  $\mathcal{T}$ , calculamos a sua normal  $\mathbf{N}_T$  através do produto vetorial de suas arestas e em seguida fazemos o teste  $\mathbf{v}_i \cdot \mathbf{N}_T$ . Se caso  $\mathbf{v}_i \cdot \mathbf{N}_T \geq 0$ , o triângulo  $T$  é descartado do teste de colisão. Caso contrário, a próxima etapa do teste de colisão consiste no cálculo do tempo de colisão  $t_I$  da partícula  $i$  com o plano determinado pelo triângulo  $T$  (Figura 6.9(a)). O tempo  $t_I$  é calculado utilizando a projeção:

$$t_I = \frac{\|(\mathbf{x}_i - \mathbf{V}_T) \cdot \mathbf{N}_T\| - r_i}{\|\mathbf{v}_i \cdot \mathbf{N}_T\|}, \quad (6-10)$$

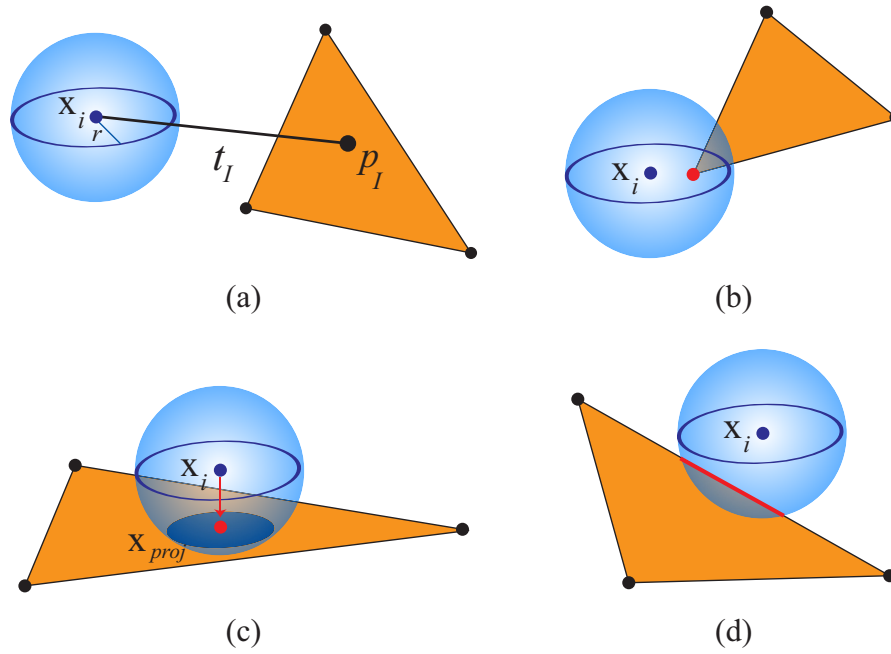


Figura 6.9: Intersecção esfera  $\times$  triângulo.

onde  $r_i$  é o raio da partícula e  $\mathbf{V}_T$  é um vértice de  $T$ . Após calcular o tempo de colisão  $t_I$ , posicionamos a partícula  $i$  no provável ponto de colisão  $\mathbf{x}'_i$  utilizando um passo do método de Euler

$$\mathbf{x}'_i = \mathbf{x}_i + t_I \mathbf{v}_i. \quad (6-11)$$

**Intersecção esfera  $\times$  triângulo.** Finalmente é feito o teste de intersecção entre a esfera associada à partícula  $i$  e o triângulo  $T$ . O teste de intersecção é realizado em três passos:

1. Teste se algum vértice de  $T$  está no interior da esfera. Se sim, a partícula  $i$  intersecta  $T$  (Figura 6.9(b)). Caso contrário, passe para o passo 2.
2. Projete o centro da esfera  $\mathbf{x}_i$  no plano determinado por  $T$ :

$$\mathbf{x}_{proj} = \mathbf{x}_i - [(\mathbf{x}_i - \mathbf{V}_T) \cdot \mathbf{N}_T] \mathbf{N}_T. \quad (6-12)$$

Verifique se  $\mathbf{x}_{proj}$  está no interior de  $T$ . Se sim, a partícula  $i$  intersecta  $T$  (Figura 6.9(c)). Caso contrário, passe para o passo 3.

3. Teste se a esfera intersecta alguma aresta de  $T$ . Se sim, a partícula  $i$  intersecta  $T$  (Figura 6.9(d)). Caso contrário, não há intersecção.

**Intersecção esfera  $\times$  segmento de reta.** No teste apresentado acima, precisamos analisar o teste de intersecção entre a partícula  $i$  e uma aresta

de  $T$  (Figura 6.9(d)). Seja  $\overline{\mathbf{V}_T \mathbf{W}_T}$  uma aresta de  $T$  formada pelos vértices  $\mathbf{V}_T$  e  $\mathbf{W}_T$ , queremos determinar se esse segmento intersecta a esfera de centro  $\mathbf{x}_i$  e raio  $r_i$ . Se a reta que passa por  $\mathbf{x}_i$  é perpendicular ao segmento  $\overline{\mathbf{V}_T \mathbf{W}_T}$  com ponto de intersecção  $\mathbf{r}$ , o teste de intersecção entre a partícula  $i$  e o segmento  $\overline{\mathbf{V}_T \mathbf{W}_T}$  se resume a determinar se o ponto  $\mathbf{r}$  está entre os pontos  $\mathbf{V}_T$  e  $\mathbf{W}_T$  (Figura 6.10). Podemos determinar o ponto  $\mathbf{r}$  através da equação paramétrica do segmento de reta

$$\mathbf{r} = \mathbf{V}_T + u\mathbf{d}, \quad \text{com} \quad \mathbf{d} = \mathbf{W}_T - \mathbf{V}_T \quad \text{e} \quad u = \frac{\mathbf{d} \cdot \mathbf{x}_i - \mathbf{d} \cdot \mathbf{V}_T}{\|\mathbf{d}\|^2}. \quad (6-13)$$

Seja  $d^2 = \|\mathbf{r} - \mathbf{x}_i\|^2$ , se caso  $d^2 > r_i^2$  a reta que passa por  $\overline{\mathbf{V}_T \mathbf{W}_T}$  não intersecta a esfera. Caso  $d^2 \leq r_i^2$  e  $u \in [0, 1]$  logo o segmento  $\overline{\mathbf{V}_T \mathbf{W}_T}$  intersecta a esfera e o ponto  $\mathbf{r}$  está entre os pontos  $\mathbf{V}_T$  e  $\mathbf{W}_T$ . O ciclo completo do teste de colisão é dado pelo Algoritmo 2.

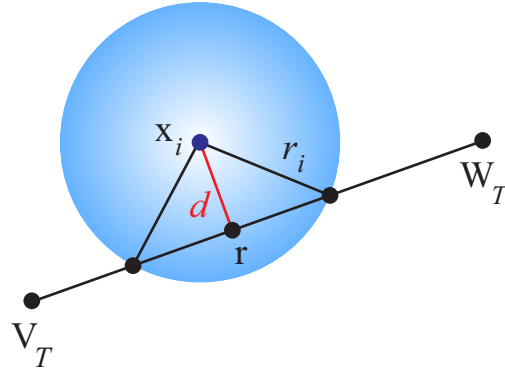


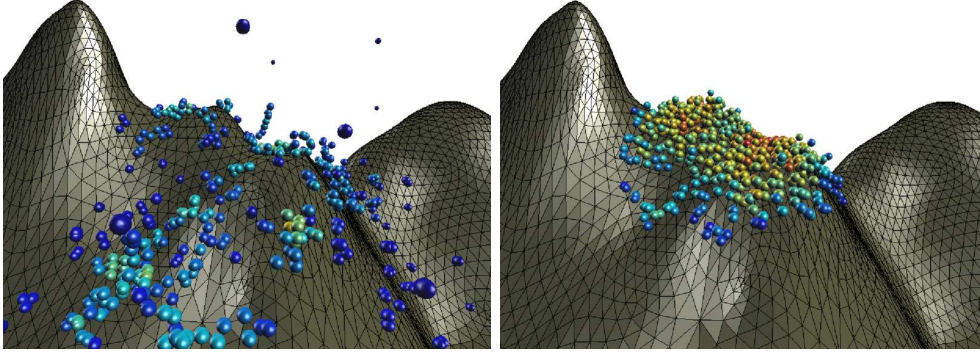
Figura 6.10: Intersecção esfera  $\times$  segmento de reta.

---

**Algoritmo 2** Teste de colisão

---

- 1: **for** cada partícula  $i$  **do**
  - 2:     **for** cada triângulo  $T$  na lista de triângulos  $\mathcal{T}$  **do**
  - 3:         **if**  $\mathbf{v}_i \cdot \mathbf{N} < 0$  **then**
  - 4:             Calcule o tempo de colisão  $t_I$  (equação (6-10))
  - 5:             **if**  $t_I < \Delta t$  **then**
  - 6:                 Calcule  $\mathbf{x}'_i = \mathbf{x}_i + t_I \mathbf{v}_i$
  - 7:                 **if** há intersecção entre a partícula na posição  $\mathbf{x}'_i$  e  $T$  **then**
  - 8:                     Calcule a velocidade de resposta  $\mathbf{v}'_i$  (Seção 6.3.1)
  - 9:                     Atualize:  $\mathbf{x}_i \leftarrow \mathbf{x}'_i + (\Delta t - t_I) \mathbf{v}'_i$  (Figura 6.7)
  - 10:                 **end if**
  - 11:             **end if**
  - 12:         **end if**
  - 13:     **end for**
  - 14: **end for**
-



6.11(a): Sem XSPH.

6.11(b): Com XSPH.

Figura 6.11: Simulação de um escoamento de lava após 1910 iterações, sem XSPH (esquerda) e com a correção XSPH (direita), com a mesma configuração da Figura 6.7. A simulação explode sem XSPH, devido à distância arbitrariamente pequena entre as partículas.

## 6.4

### Aspectos numéricos

Nessa seção, com o objetivo de melhorar a estabilidade numérica das simulações, introduziremos algumas ferramentas numéricas tais como viscosidade artificial e uma variante do método SPH conhecida como XSPH.

#### 6.4.1

##### Viscosidade artificial

Para evitar instabilidades numéricas devidas às oscilações nos campos vetoriais da velocidade e pressão, utilizamos uma técnica muito comum em elementos finitos que consiste em adicionar um termo de viscosidade artificial na equação de momento (4-38) a fim de dissipar essas oscilações indesejadas. Isso é feito da seguinte forma:

$$\frac{d\mathbf{v}_i}{dt} \leftarrow \frac{d\mathbf{v}_i}{dt} - \sum_{j \in N(\mathbf{x}_i)} \frac{m_j}{\rho_i} \Pi_{ij} \nabla_i W_{ij}. \quad (6-14)$$

O efeito da viscosidade artificial em sistemas SPH se deve ao seguinte termo:

$$\Pi_{ij} = \begin{cases} \frac{-\alpha \mu_{ij} c}{0.5(\rho_i + \rho_j)}, & \mathbf{v}_{ij} \cdot \mathbf{x}_{ij} < 0 \\ 0, & \mathbf{v}_{ij} \cdot \mathbf{x}_{ij} \geq 0 \end{cases} \quad \text{com} \quad \mu_{ij} = \frac{h(\mathbf{v}_{ij} \cdot \mathbf{x}_{ij})}{\|\mathbf{x}_{ij}\|^2 + 0.01h^2}, \quad (6-15)$$

com  $\mathbf{x}_{ij} = \mathbf{x}_i - \mathbf{x}_j$  e  $\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$ . O termo linear da equação (6-15) corresponde à viscosidade volumétrica (*bulk viscosity*) [37]. Geralmente, o valor da constante  $\alpha$  é tomado próximo de 1.

### 6.4.2

#### Correção de velocidade XSPH

Com o objetivo de prevenir a interpenetração de partículas, a qual pode gerar aglomerados instáveis de partículas, Monaghan [46] introduziu a técnica XSPH (X de desconhecido) que consiste em calcular uma média das velocidades das partículas vizinhas. Essa técnica permite que as partículas se movimentem de uma forma mais ordenada num escoamento incompressível, reduzindo o problema de desordem de partícula nas simulações SPH (Figura 6.11).

Na técnica XSPH, a velocidade de cada partícula  $i$  é corrigida através da média das velocidades de suas partículas vizinhas ponderada por um parâmetro global constante  $\varepsilon \in [0, 1]$  da seguinte forma:

$$\mathbf{v}_i \leftarrow \mathbf{v}_i + \varepsilon \sum_{j \in N(\mathbf{x}_i)} \frac{2m_j}{(\rho_i + \rho_j)} (\mathbf{v}_j - \mathbf{v}_i) W_{ij}. \quad (6-16)$$

### 6.4.3

#### Integração numérica

Na simulação SPH de fluidos viscoplásticos, também integramos as equações de Navier–Stokes utilizando o integrador leap–frog (Seção 5.2.2), que dentre os integradores de precisão de segunda ordem é o mais eficiente, pois realiza uma única avaliação da aceleração por passo de tempo e utiliza pouco consumo de memória de armazenamento por avaliação. Nessa simulação, a condição de estabilidade CFL leva em conta os efeitos viscosos, resultando num passo de tempo adaptativo determinado pela expressão [50]:

$$\Delta t = 0.1 \min_i \left\{ \frac{h}{\|\mathbf{v}_i\| + c}, \frac{h^2 \rho_i}{8 \eta_i} \right\}. \quad (6-17)$$

## 6.5

### Implementação

A implementação do sistema SPH para fluidos viscoplásticos é feita de maneira muito semelhante a que foi mostrada no Capítulo 5. A principal diferença nessa implementação consiste no cálculo da viscosidade a partir do tensor de deformação, o ciclo de simulação SPH para fluidos viscoplásticos segue o esquema da Figura 6.12.

A inicialização das partículas no interior da malha triangular dos objetos simulados é feita utilizando um algoritmo de detecção de ponto no interior de um poliedro e pode ser encontrado no livro de O'Rourke [55], assim como um algoritmo para calcular o volume desses objetos (poliedros). Nessa implementação, existem os atributos de sistema e os atributos de partícula. Os atri-

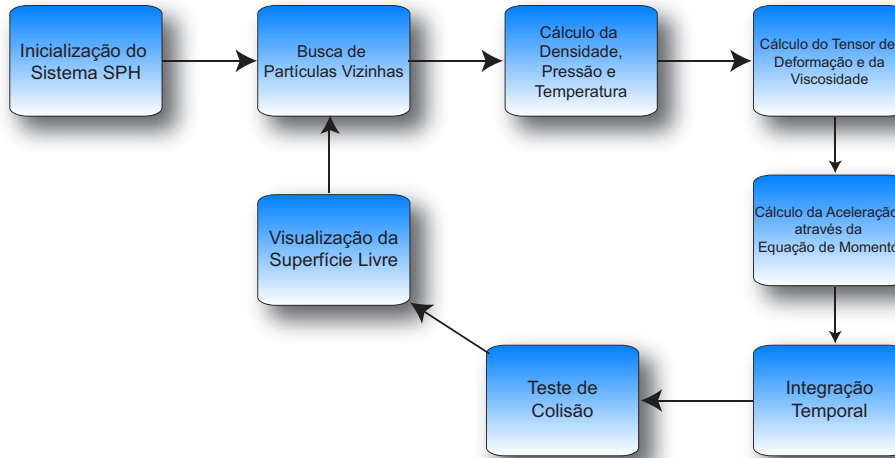


Figura 6.12: Ciclo de simulação de um sistema SPH para fluidos não-newtonianos.

butos de sistema como massa, velocidade do som e o comprimento suave  $h$  são constantes globais e não se alteram com o tempo. Os atributos de partícula variam em relação ao tempo e são armazenados em cada partícula. Esses atributos são dados pela Tabela 6.1 e são atualizados na seqüência do Algoritmo 3.

Atributo	Descrição
$\mathbf{x}$	posição
$\mathbf{v}$	velocidade
$\mathbf{a}$	aceleração
$\mathbf{D}$	tensor de deformação
$\rho$	densidade
$\eta$	viscosidade
$T$	temperatura

Tabela 6.1: Atributos da partícula.

---

**Algoritmo 3** Sistema SPH para fluidos viscoplásticos

---

```
1: repeat
2:   Faça a busca das partículas vizinhas. (Seção 5.1.2)
3:   for cada partícula  $i$  do
4:     Atualize a pressão  $p_i$ . (equação (4-41) )
5:     Atualize a viscosidade  $\eta_i$ . (equação (6-1))
6:   end for
7:   for cada partícula  $i$  do
8:     Calcule a derivada da densidade. (equação (4-37))
9:     Calcule a aceleração. (equações (4-38) e (6-14))
10:    Calcule a derivada da temperatura. (equação 6-2)
11:  end for
12:  for cada partícula  $i$  do
13:    Atualize  $\mathbf{v}_i$ ,  $T_i$  e  $\rho_i$  com o integrador leap–frog. (Seção 5.2.2)
14:    Correção de  $\mathbf{v}_i$  com XSPH. (equação (6-16))
15:    Eventual colisão:  $(\mathbf{x}_i, \mathbf{v}_i) \leftarrow \text{colisao}(\mathbf{x}_i + \Delta t \mathbf{v}_i)$ . (Seção 6.3)
16:  end for
17:  Atualize  $\Delta t$  usando a condição CFL. (equação (6-17))
18:   $\text{tempo} = \text{tempo} + \Delta t$ 
19: until  $\text{tempo} < \text{tempo}_{total}$ 
```

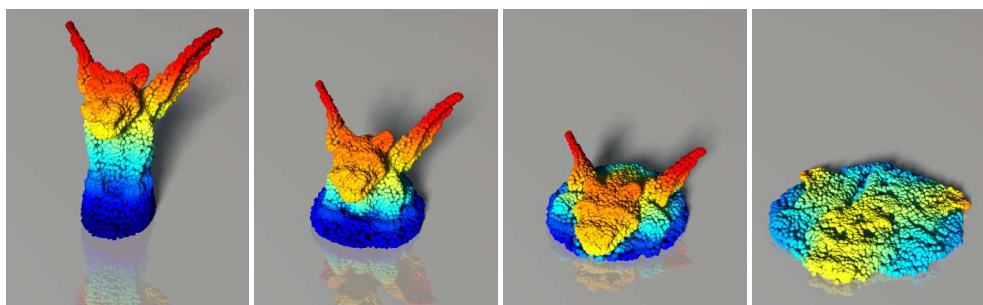
---

## 6.6 Resultados

Testamos o método descrito nesse capítulo com a finalidade de simular fenômenos físicos complexos. Nosso principal objetivo está ilustrado nas Figuras 1.1 e 6.5, onde os resultados batem com a intuição dos processos de deformação plástica, derretimento de objetos sólidos e fluxos de lava.

Propomos uma nova formulação lagrangeana para fluidos não-newtonianos usando o método SPH combinado com algumas técnicas numéricas com intuito de garantir a estabilidade das simulações. A estabilidade aparece claramente na Figura 6.13, onde a cabeça do modelo Gargoyle permanece bem definida mesmo quando o modelo está quase completamente derretido. Nesse caso, todas as 6976 partículas iniciam com a temperatura acima do ponto de fusão e fluem como fluido não-newtoniano.

O controle eficiente da viscosidade através da formulação de Fluidos Newtonianos Generalizados permite resultados visualmente realísticos. No exemplo da Figura 6.14, criamos um objeto viscoplástico a partir da superfície chair com 7000 partículas e simulamos a sua interação com um objeto sólido complexo representado pelo esqueleto de uma mão com 2351 triângulos. O nosso método, além de capturar muito bem os efeitos viscosos do material, permite que o objeto sofra grandes mudanças topológicas sem controle explícito de sua superfície livre. Perceba que, mesmo após quase todo material escorrer entre



6.13(a): 20 iter.      6.13(b): 550 iter.      6.13(c): 780 iter.      6.13(d): 1320 iter.

Figura 6.13: Derretimento do modelo Gargoyle totalmente líquido usando 6976 partículas: o mapa de cores representa a temperatura de cada partícula. A estabilidade do método preserva a forma do objeto sem uma representação explícita da malha mesmo após várias iterações.

os dedos e tocar o chão, parte dele fica agarrado no esqueleto.

A flexibilidade do nosso método, além de permitir simulações de fluidos viscoplásticos, também possibilita a simulação de objetos sólidos que sofrem deformações. No exemplo da Figura 6.15, simulamos o impacto de uma esfera de metal contra uma parede plástica (verde). Note que, momentos antes da esfera perfurar a parede, a energia dissipada pelo choque da esfera produz uma grande deformação no material.

O exemplo mostrado na Figura 6.16 simula o derretimento do modelo Stanford Bunny com 10188 partículas. A simulação é inicializada com um gradiente linear de temperatura tal que a orelhas derretem enquanto o corpo permanece frio e sólido. Note que o comportamento físico é coerente, especialmente quando uma das orelhas esfria após tocar o corpo, enquanto a outra permanece quente (Figura 6.4).

A adaptatividade do passo de tempo devido à condição CFL além de garantir estabilidade numérica na integração temporal permite que as simulações sejam mais eficientes. Por exemplo, o derretimento das letras “CGF 2007” mostrado na Figura 6.17 é discretizado utilizando 10773 partículas, ou seja, mais partículas que o modelo Bunny. Entretanto, essa simulação requer um tempo médio de execução menor para cada passo de tempo (veja Tabela 6.2). Isso se deve à baixa densidade de partículas do modelo, a qual permite que os passos de tempo sejam maiores.

Nossas simulações também obtiveram sucesso em reproduzir a morfologia vista em escoamentos reais de lava tais como o espalhamento da frente de lava na ausência de solidificação e o desenvolvimento de complexas e assimétricas estruturas em forma de dedos conhecidas como *lobus* (veja Figura 6.18). A formação dessas estruturas se deve à influência da topografia do terreno na lava; essa influência é reproduzida graças ao teste de colisão introduzido na

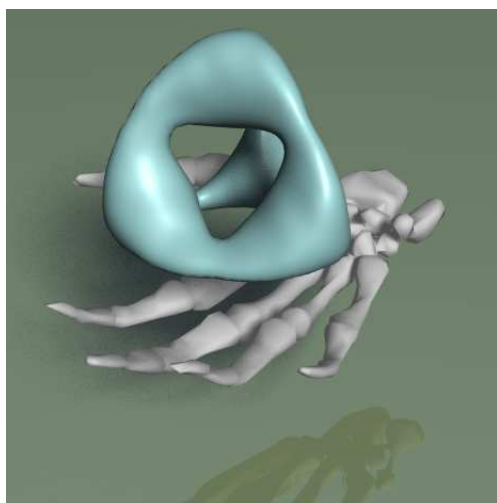


Seção 6.3.

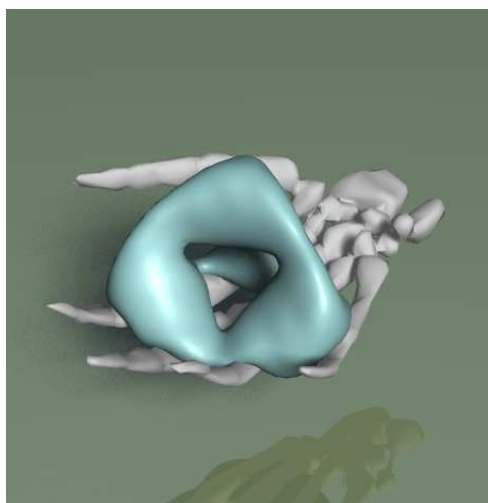
Resumindo, o modelo de Fluido Newtoniano Generalizado proposto por Mendes se adapta muito bem em nossas simulações SPH, pois além de sua formulação ser concisa, ela mantém o significado físico de todo processo.

<b>Simulação</b>	<b>Número de partículas</b>	<b>Tempo por iteração</b>
Mão	5000	0.13s
Parede	5000	0.09s
Chair	7000	0.15s
Bunny	10188	0.18s
Vela	7364	0.11s
Gargoyle	7007	0.16s
CGF	10773	0.13s
Terreno	4900	0.08s
Pão de Açúcar	10137	0.48s

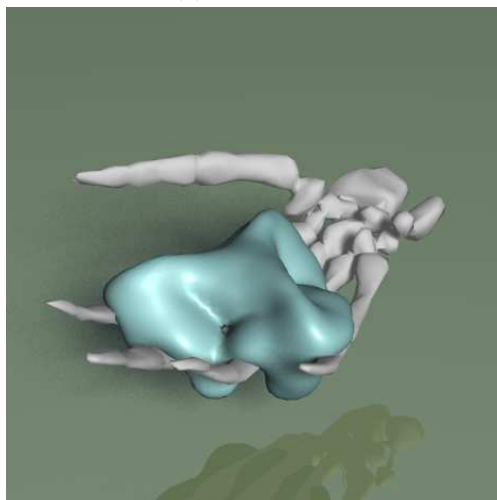
Tabela 6.2: Tempo médio de iteração em cada simulação executadas num computador com processador Pentium 4 – 2.4 GHz e com 2 Gb de RAM.



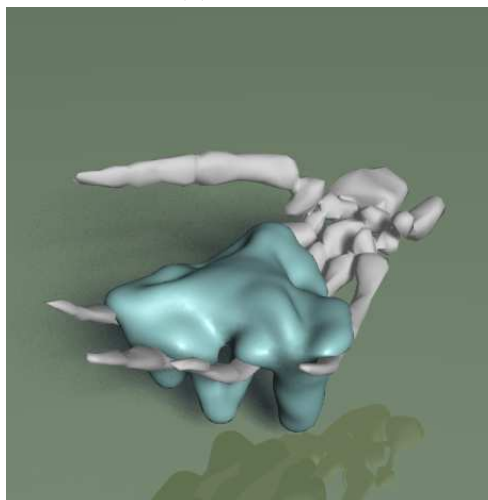
6.14(a): Objeto inicial.



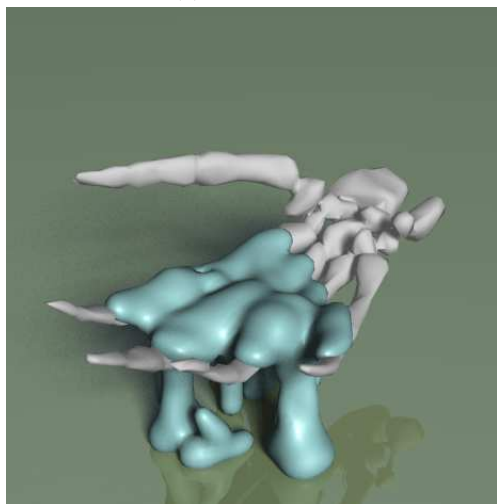
6.14(b): 600 iterações.



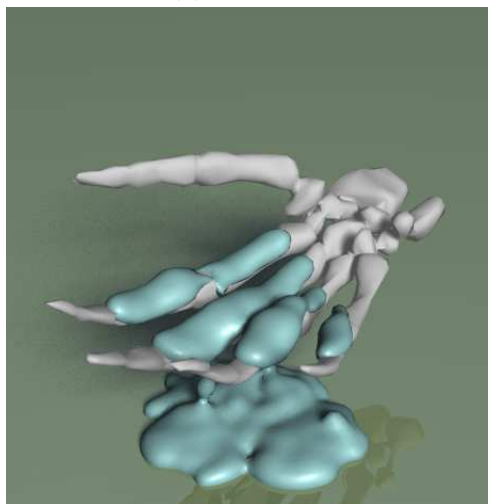
6.14(c): 1400 iterações.



6.14(d): 2800 iterações.

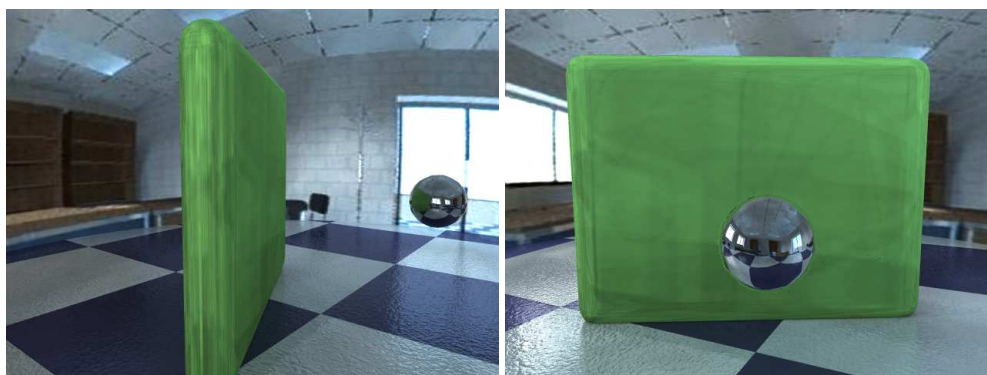


6.14(e): 6000 iterações.



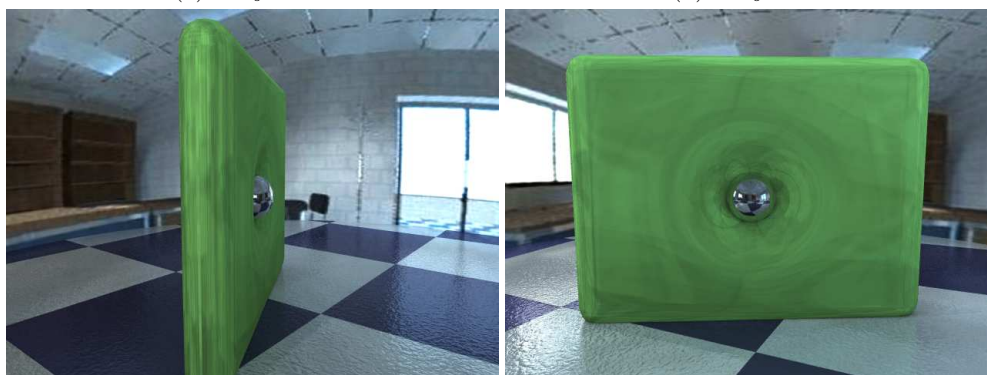
6.14(f): 19000 iterações.

Figura 6.14: Superfície chair modelada como um material viscoplástico com 7000 partículas, interagindo com um objeto sólido complexo representado pelo esqueleto de uma mão com 2351 triângulos.



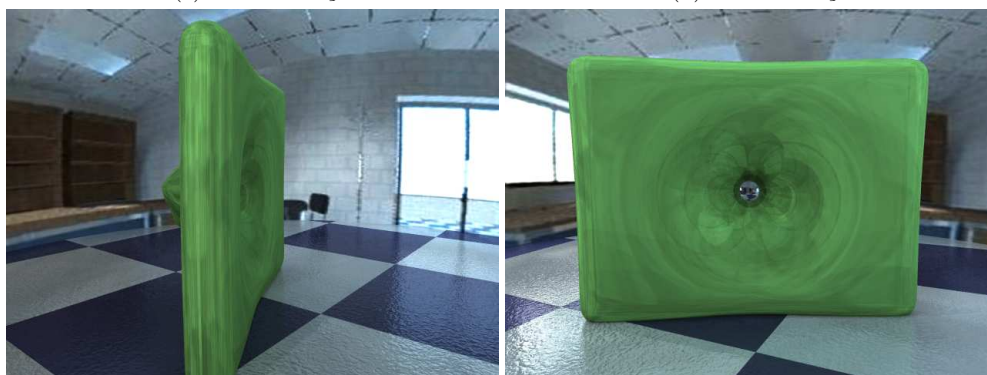
6.15(a): Objeto inicial.

6.15(b): Objeto inicial.



6.15(c): 1300 iterações.

6.15(d): 1300 iterações.



6.15(e): 1500 iterações.

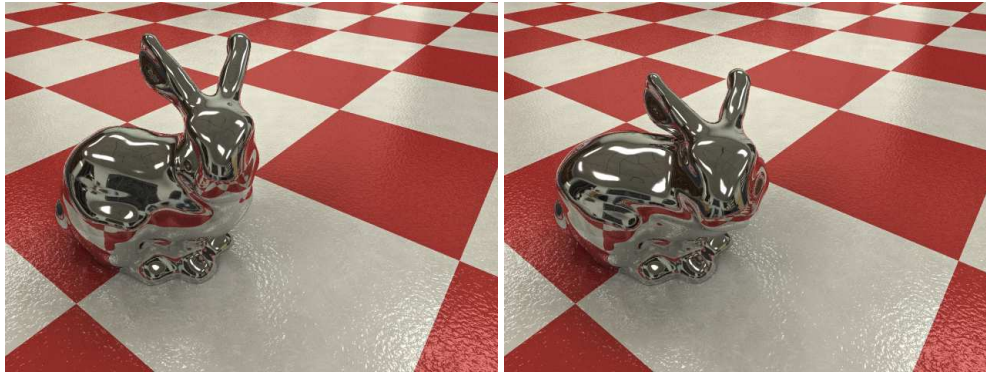
6.15(f): 1500 iterações.



6.15(g): 2000 iterações.

6.15(h): 2000 iterações.

Figura 6.15: Colisão de uma esfera metálica contra uma parede plástica (verde) constituída por 5000 partículas. A simulação é vista ao mesmo tempo em duas posições diferentes: vista lateral (esquerda) e vista frontal (direta).

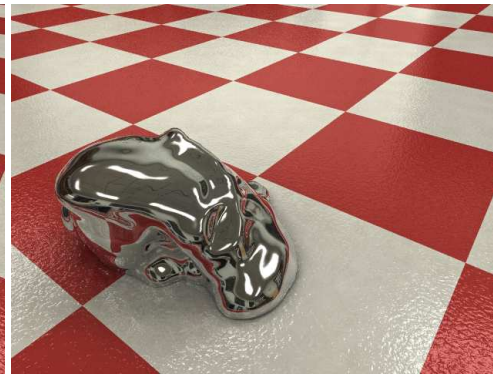


6.16(a): Objeto inicial.

6.16(b): 400 iterações.



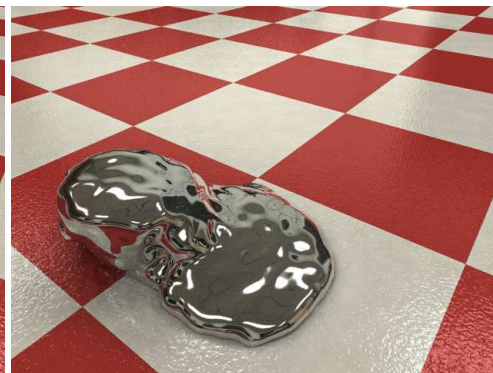
6.16(c): 800 iterações.



6.16(d): 1200 iterações.



6.16(e): 1800 iterações.



6.16(f): 4000 iterações.

Figura 6.16: Evolução da superfície livre da simulação de derretimento do modelo Stanford Bunny com 10188 partículas, iniciando frio na base e quente no topo do modelo.



6.17(a): Objeto inicial.



6.17(b): 450 iterações.



6.17(c): 900 iterações.



6.17(d): 1500 iterações.



6.17(e): 2500 iterações.



6.17(f): 4150 iterações.

Figura 6.17: Derretimento das letras “CGF 2007” usando 10773 partículas, iniciando frio na base e quente no topo do modelo. A adaptatividade do passo de tempo permite uma simulação mais precisa com poucas iterações.

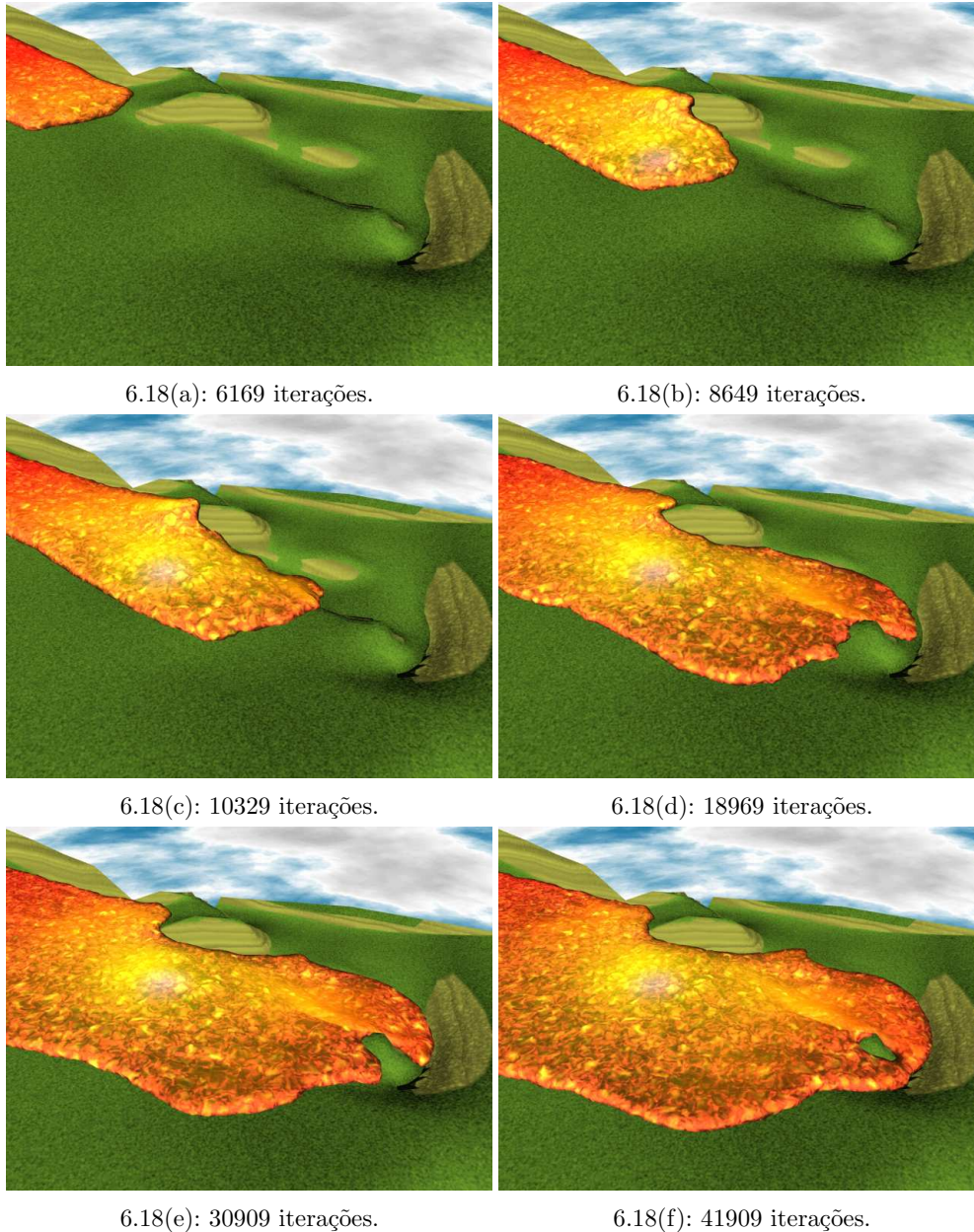


Figura 6.18: Escoamento de lava num terreno virtual de 1547 triângulos, com 4900 partículas.

## 7

### Fluidos multifásicos incompressíveis

Nesse capítulo, introduziremos um novo método híbrido grid-partícula para simular com eficiência a dinâmica de fluidos multifásicos imiscíveis, ou seja, fluidos cujas as fases que não trocam massa entre si. Para validar o método, simularemos o fenômeno clássico de instabilidade que ocorre quando fluidos de densidade diferentes se misturam. Esse fenômeno é conhecido como *instabilidades de Rayleigh-Taylor* [9].

O método proposto consiste em discretizar os fluidos utilizando uma versão modificada do método SPH. Com o objetivo de garantir a incompressibilidade dos fluidos, integramos a velocidade de cada partícula para obter uma velocidade intermediária e em seguida projetamos essa velocidade num espaço de divergência livre. Para realizar a projeção, utilizamos um grid virtual para resolver implicitamente uma equação de Poisson da pressão. A validação do método é feita em simulações 2D das instabilidades de Rayleigh-Taylor e do movimento hidrodinâmico de gotas.

Apresentaremos o novo método proposto através das seguintes seções desse capítulo. Na Seção 7.1, introduziremos uma versão modificada do SPH para discretizar a física que governa a dinâmica dos fluidos multifásicos. Mais adiante na Seção 7.2, discutiremos como é feito o cálculo implícito da pressão com intuito de garantir a incompressibilidade do SPH. Na Seção 7.3, discutiremos detalhes de implementação. Finalmente, na Seção 7.4, mostraremos os resultados obtidos pelo nosso método.

## 7.1

### Formulação SPH para fluidos multifásicos

Assim como nos capítulos anteriores, a formulação lagrangeana de fluidos multifásicos incompressíveis é dada pelas equações de Navier-Stokes (2-9) – (2-10):

$$\frac{d\mathbf{v}_i}{dt} = -\frac{1}{\rho_i} \nabla p_i + \frac{\mu_i}{\rho_i} \nabla^2 \mathbf{v}_i + \mathbf{g}. \quad (7-1)$$

A aproximação SPH de (7-1) é feita utilizando as aproximações (4-42) para o gradiente da pressão e (6-9) para o vetor laplaciano da velocidade, logo

$$\frac{d\mathbf{v}_i}{dt} = - \sum_{j \in N(\mathbf{x}_i)}^n m_j \left( \frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \nabla_i W_{ij} + \sum_{j \in N(\mathbf{x}_i)}^n \frac{\mu_i m_j}{\rho_i \rho_j} \mathbf{v}_{ij} \frac{\mathbf{x}_{ij} \cdot \nabla_i W_{ij}}{\|\mathbf{x}_{ij}\|^2} + \mathbf{g}. \quad (7-2)$$

Entretanto, a aproximação (7-2), quando utilizada em simulações de fluidos multifásicos, produz uma tensão superficial artificial na interface entre os dois fluidos [67]. Essa tensão superficial se deve ao salto da densidade na interface; logo precisamos reescrever a equação (7-2) de forma que não dependa da densidade: para isso vamos definir o *número de densidade da partícula* como sendo

$$n_i = \frac{1}{\Delta V_i} = \frac{\rho_i}{m_i}. \quad (7-3)$$

O número de densidade pode ser calculado a partir da aproximação SPH (4-12) da seguinte forma:

$$n_i = \sum_{j \in N(\mathbf{x}_i)}^n n_j W_{ij} \Delta V_j = \sum_{j \in N(\mathbf{x}_i)}^n \frac{1}{\Delta V_j} W_{ij} \Delta V_j = \sum_{j \in N(\mathbf{x}_i)}^n W_{ij}. \quad (7-4)$$

Assim, multiplicando os dois lados da equação (7-1) por  $m_i$  temos

$$\mathbf{F}_i = m_i \frac{d\mathbf{v}_i}{dt} = -\frac{1}{n_i} \nabla p_i + \frac{\mu_i}{n_i} \nabla^2 \mathbf{v}_i + m_i \mathbf{g} \quad (7-5)$$

e em seguida repetindo a aproximação SPH que foi feita em (7-2) em (7-5), segue que

$$\mathbf{F}_i = - \sum_{j \in N(\mathbf{x}_i)}^n \left( \frac{p_i}{n_i^2} + \frac{p_j}{n_j^2} \right) \nabla_i W_{ij} + \sum_{j \in N(\mathbf{x}_i)}^n \frac{\mu_i}{n_i n_j} \mathbf{v}_{ij} \frac{\mathbf{x}_{ij} \cdot \nabla_i W_{ij}}{\|\mathbf{x}_{ij}\|^2} + m_i \mathbf{g}, \quad (7-6)$$

onde  $\mathbf{F}_i$  é a força total atuando na partícula  $i$ . Portanto, para as simulações SPH de fluidos multifásicos, utilizamos a aproximação baseada no número de densidade da partícula (7-6).

## 7.2

### Incompressibilidade SPH através de um grid

A equação da continuidade nos diz que um fluido incompressível possui densidade constante  $\rho^0$ . Isso é equivalente a dizer que o número de densidade



de cada partícula é constante,  $n_0$ . Nessa seção, vamos apresentar uma nova forma de garantir a incompressibilidade do método SPH (Seção 7.1) através de um método de projeção da pressão (Seção 3.1).

No método que propomos, a projeção é realizada da seguinte maneira: em cada passo de tempo, a força viscosa e gravitacional da equação (7-6) são explicitamente calculadas e cada partícula é deslocada para uma posição intermediária  $\mathbf{x}_i^*$ ,

$$\mathbf{v}_i^* = \mathbf{v}_i^n + \frac{\Delta t}{m_i} \left( \sum_{j \in N(\mathbf{x}_i)} \frac{\mu_i}{n_0^2} \mathbf{v}_{ij} \frac{\mathbf{x}_{ij} \cdot \nabla_i W_{ij}}{\|\mathbf{x}_{ij}\|^2} + m_i \mathbf{g} \right), \quad (7-7)$$

$$\mathbf{x}_i^* = \mathbf{x}_i^n + \Delta t \mathbf{v}_i^*. \quad (7-8)$$

Na etapa seguinte, calculamos para cada partícula o número de densidade  $n_i^*$  na posição intermediária  $\mathbf{x}_i^*$  através da equação (7-4). Quando o número de densidade  $n_i^*$  não é  $n_0$ , ele é corrigido para  $n_0$  através do valor de correção

$$n'_i = n_0 - n_i^*. \quad (7-9)$$

O valor  $n'_i$  está associado a uma velocidade de correção  $\mathbf{v}'_i$ , logo pela equação da continuidade (2-7) e por (7-3) temos:

$$\nabla \cdot \mathbf{v}'_i = -\frac{1}{\Delta t} \frac{\rho_i^0 - \rho_i^*}{\rho_i^0} = -\frac{1}{\Delta t} \frac{m_i n_0 - m_i n_i^*}{m_i n_0} = -\frac{1}{\Delta t} \frac{n_0 - n_i^*}{n_0}, \quad (7-10)$$

sendo  $\rho_i^0$  a densidade inicial do fluido a qual a partícula  $i$  pertence. Usando o fato de  $\mathbf{v}'_i = \mathbf{v}_i^{n+1} - \mathbf{v}_i^*$  e substituindo (7-10) em (3-2), obtemos a pressão dada implicitamente pela equação de Poisson:

$$\nabla^2 P_i^{n+1} = -\frac{1}{\Delta t^2} \frac{n_i^* - n_0}{n_0}, \quad (7-11)$$

com  $P_i^{n+1} = \frac{p_i^{n+1}}{\rho_i^0}$ .

Após resolver a equação (7-11) e obter a pressão  $p_i^{n+1} = \rho_i^0 P_i^{n+1}$ , a atualização velocidade com divergência livre  $\mathbf{v}_i^{n+1}$  é feita através da correção da velocidade  $\mathbf{v}_i^*$  adicionando o termo da força de pressão de (7-6):

$$\mathbf{v}_i^{n+1} = \mathbf{v}_i^* - \frac{\Delta t}{m_i} \left[ \sum_{j \in N(\mathbf{x}_i)} \left( \frac{p_i^{n+1} + p_j^{n+1}}{n_0^2} \right) \nabla_i W_{ij} \right]. \quad (7-12)$$

Finalmente, a posição da partícula é atualizada

$$\mathbf{x}_i^{n+1} = \mathbf{x}_i^n + \Delta t \frac{\mathbf{v}_i^{n+1} + \mathbf{v}_i^n}{2}. \quad (7-13)$$

Os métodos lagrangeanos que calculam a pressão de forma implícita, tais como o PSPH [15] e o MPS [33], utilizam uma aproximação por partículas do

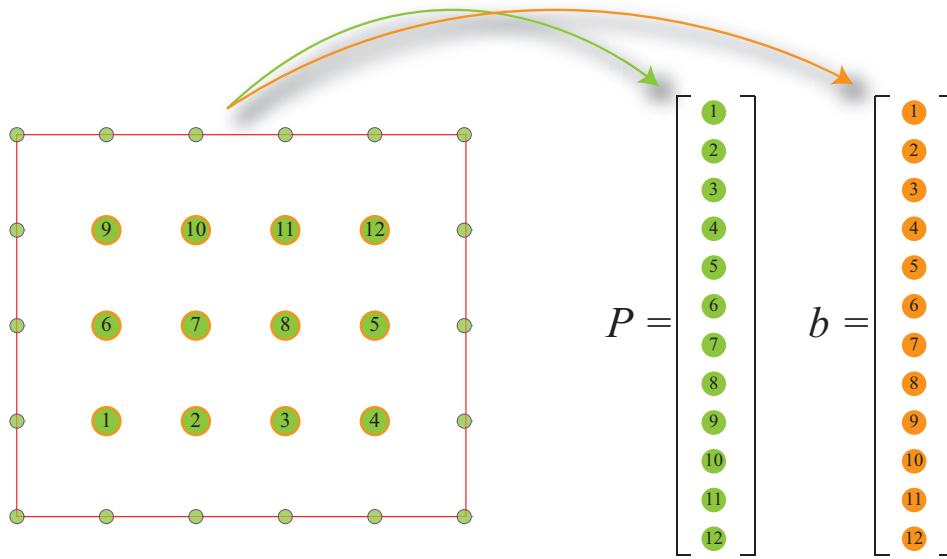


Figura 7.1: Os vetores associados à  $P$  e  $b$  seguem a ordenação natural do grid 2D.

operador laplaciano da equação (7-11), semelhante à que foi utilizada em (6-9). Entretanto, o cálculo da pressão através dessa aproximação é dado pela solução de um grande sistema, pois a dimensão da matriz esparsa obtida por essa aproximação é de ordem  $N^2$ , onde  $N$  é o número total de partículas envolvidas na simulação. Além disso, essa matriz se modifica a cada instante de tempo, pois essa aproximação depende da posição corrente das partículas. Por essas razões, utilizamos um grid fixo no espaço para representar o campo escalar da pressão e discretizar o operador laplaciano da equação (7-11) através do método de diferenças finitas central de segunda ordem:

$$L_{i_g, j_g} P = \frac{P_{i_g+1, j_g} - 2P_{i_g, j_g} + P_{i_g-1, j_g}}{\delta x^2} + \frac{P_{i_g, j_g+1} - 2P_{i_g, j_g} + P_{i_g, j_g-1}}{\delta y^2}, \quad (7-14)$$

onde  $(i_g, j_g)$  é a localização no grid,  $\delta x$  e  $\delta y$  são os espaçamentos horizontal e vertical do grid, respectivamente. Em particular assumiremos que  $\delta x = \delta y$ , logo o operador (7-14) assume a seguinte forma

$$L_{i_g, j_g} P = \frac{P_{i_g+1, j_g} + P_{i_g-1, j_g} + P_{i_g, j_g+1} + P_{i_g, j_g-1} - 4P_{i_g, j_g}}{\delta x^2}. \quad (7-15)$$

Desde que os valores na fronteira do grid  $P$  sejam conhecidos, enumeramos os vértices do grid 2D no sentido da esquerda para a direita e de baixo para cima. Essa enumeração do grid é chamada de *ordenação natural* e é mostrada na Figura 7.1. Em seguida montamos os vetores associados a  $P$  e  $b$  seguindo essa enumeração, dessa forma a matriz associada ao operador (7-15) é uma matriz simétrica positiva definida e tem a seguinte estrutura de blocos

$$A = \begin{pmatrix} B & -I & & \\ -I & B & -I & \\ & -I & B & \end{pmatrix} \quad \text{com} \quad B = \begin{pmatrix} 4 & -1 & & \\ -1 & 4 & -1 & \\ & -1 & 4 & -1 \\ & & -1 & 4 \end{pmatrix} \quad (7-16)$$

e  $I$  é a matriz identidade.

### 7.3 Implementação

Na Seção 7.2, mostramos como a pressão é calculada através do método da projeção utilizando um grid para discretizar a equação de Poisson (7-11). Nessa seção, vamos descrever algumas etapas importantes na implementação do método híbrido grid-partícula (veja Figura 7.2), em especial a maneira como é feita a integração do grid da pressão com as partículas do SPH.

As simulações que utilizam o método proposto são feitas em domínios retangulares 2D. A inicialização do SPH é feita de forma semelhante aos capítulos anteriores, a diferença mais significativa na simulação de fluidos multifásicos é que cada fluido possui a sua própria densidade e viscosidade. Após a inicialização do SPH são gerados dois grids: grid da pressão  $P$  e um grid  $b$  contendo os valores do vetor solução da equação de Poisson (lado direito da equação (7-11)). A dimensão de cada grid é a mesma que a do grid de busca de partículas vizinhas (veja Seção 5.1.2). A configuração do método é dada conforme a Figura 7.3.

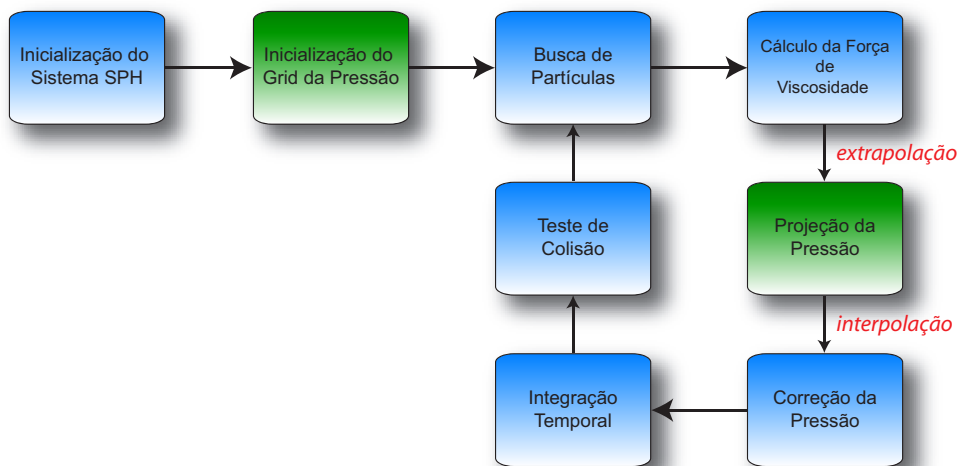


Figura 7.2: Ciclo de simulação do método híbrido grid-partícula: em azul são as etapas que utilizam SPH e em verde as etapas que utilizam o grid da pressão.

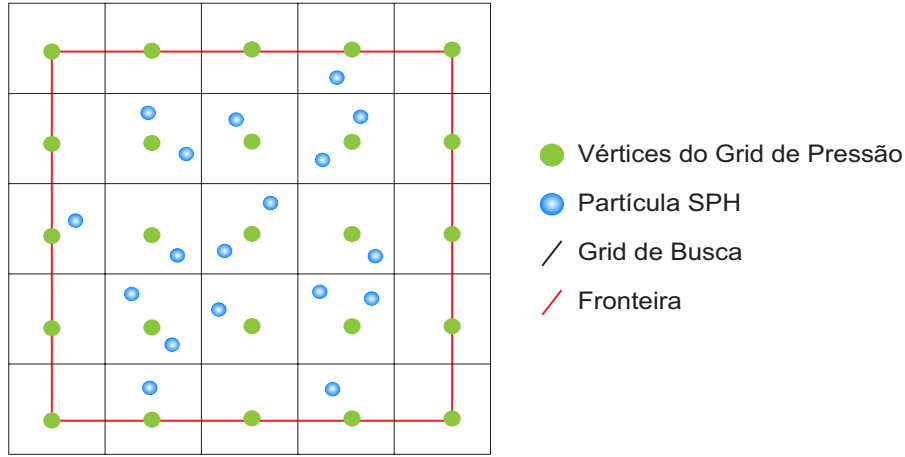


Figura 7.3: Configuração do método híbrido grid-partícula. A dimensão do grid de pressão é a mesma que a do grid utilizado na busca de partículas vizinhas.

**Extrapolação partícula → grid.** Depois de mover cada partícula para a posição intermediária  $\mathbf{x}_i^*$  através de (7-7) – (7-8), precisamos extrapolar o número de densidade  $n_i^*$  para os vértices do grid  $b$ . Isso é feito utilizando a aproximação SPH (7-4), assim temos:

$$n_{i_g, j_g}^* = \sum_{j \in N(\mathbf{x}_i)} W(\mathbf{p}_{i_g, j_g} - \mathbf{x}_i^*, h), \quad (7-17)$$

sendo  $\mathbf{p}_{i_g, j_g}$  a posição do vértice do grid com coordenadas  $(i_g, j_g)$ . Os valores do grid  $b$  são dados por:

$$b_{i_g, j_g} = -\frac{1}{\Delta t^2} \frac{n_{i_g, j_g}^* - n^0}{n^0}. \quad (7-18)$$

Para acelerar o processo de extrapolação, utilizamos o grid de busca para fazer as avaliações de (7-17).

**Solução da equação de Poisson.** Uma vez criado o grid  $b$ , para obter a pressão  $P$  precisamos resolver o sistema

$$AP = b, \quad (7-19)$$

onde  $A$  é a matriz obtida em (7-16). Para isso, utilizamos o *método iterativo de Jacobi*. O método de Jacobi assim como todos os métodos iterativos, inicia com um “chute” inicial  $P^{(0)}$  (por exemplo, o vetor nulo) como solução do sistema (7-19). A cada iteração  $k$  é produzida uma solução aproximada  $P^{(k)}$  de (7-19) da seguinte forma:

$$P_{i_g, j_g}^{(k)} = \frac{P_{i_g+1, j_g}^{(k-1)} + P_{i_g-1, j_g}^{(k-1)} + P_{i_g, j_g+1}^{(k-1)} + P_{i_g, j_g-1}^{(k-1)} - \delta x^2 b_{i_g, j_g}}{4}. \quad (7-20)$$

Métodos mais sofisticados, tais como o *método do gradiente conjugado* [63] e o *método multigrid* [70], convergem mais rapidamente do que o método de Jacobi. Utilizamos o método de Jacobi devido à sua simplicidade e pela facilidade de sua implementação. Uma abordagem mais ampla do método de Jacobi pode ser encontrada no livro de Saad [63].

**Condição de fronteira para a pressão.** A equação de Poisson é um problema de contorno, logo para cada iteração do método de Jacobi a solução desse problema requer a *condição de fronteira de Neumann* para a pressão

$$\frac{\partial P}{\partial \mathbf{n}} = 0. \quad (7-21)$$

Isso significa que na fronteira do domínio, a taxa de variação da pressão na direção normal  $\mathbf{n}$  é nula. Entretanto, precisamos determinar o valor da pressão nos vértices de  $P$  situados na fronteira do domínio.

Seja  $(L+2) \times (M+2)$  a dimensão de  $P$ . Para calcular a pressão nos vértices de fronteira no lado esquerdo, discretizamos (7-21) utilizando diferenças finitas adiantadas e obtemos:

$$\frac{P_{1,j_g} - P_{0,j_g}}{\delta x} = 0, \quad \text{com } j_g \in [1, M]. \quad (7-22)$$

Portanto, a pressão na fronteira  $P_{0,j_g}$  é dada por

$$P_{0,j_g} = P_{1,j_g}, \quad \text{com } j_g \in [1, M]. \quad (7-23)$$

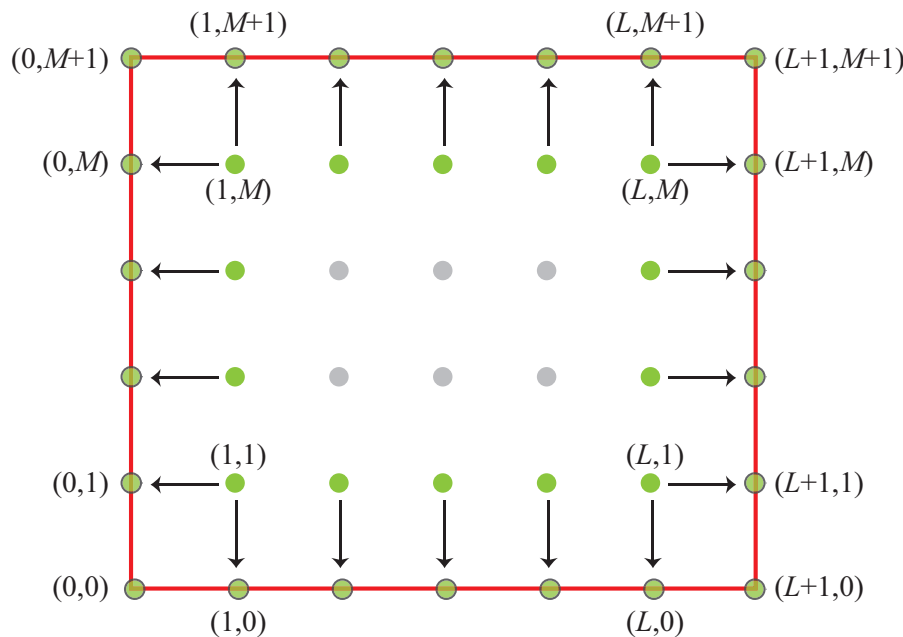


Figura 7.4: Condição de Neumann num grid  $(L+2) \times (M+2)$ : as setas indicam os valores de  $P$  copiados para os vértices da fronteira.

Em outras palavras, para determinar a pressão na fronteira basta copiar os valores da pressão no interior do domínio para a fronteira como é esquematizado na Figura 7.4. Os valores nas quinas de  $P$  são calculados fazendo uma média dos valores de seus vértices adjacentes.

**Interpolação grid  $\rightarrow$  partícula.** Uma vez resolvido o sistema (7-19) e obtido o grid  $P$ , a próxima etapa é “devolver” a pressão para as partículas. Logo, para cada partícula  $i$  precisamos localizar em qual célula de  $P$  ela está contida. A localização da partícula é feita usando um mapeamento semelhante a (5-1) e em seguida a pressão na partícula  $i$  é obtida através de uma interpolação bi-linear dos vértices dessa célula conforme é mostrado na Figura 7.5.

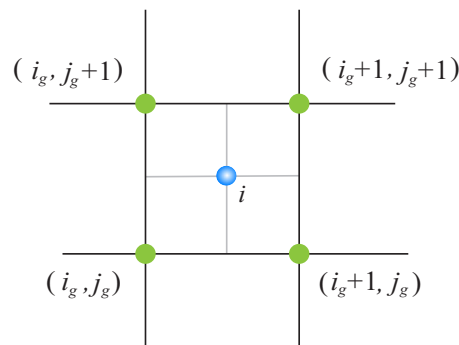


Figura 7.5: A pressão na partícula  $i$  é obtida através de uma interpolação dos vértices de  $P$  (verde).

**Correção da velocidade.** Após interpolar a pressão de volta para todas as partículas, a velocidade da partícula é corrigida através de (7-12), a posição é atualizada por (7-13) e o ciclo se repete. Devido à formulação incompressível do método, o uso da velocidade do som é dispensado no cálculo da pressão, logo a condição CFL é puramente baseada na velocidade do fluido, permitindo o uso de passos de tempo  $\Delta t$  consideravelmente maiores que os utilizados nas formulações SPH dos capítulos anteriores. Assim, a condição CFL que utilizamos é

$$\Delta t = 0.25 \min_i \left\{ \frac{h}{\|\mathbf{v}_i\|} \right\}. \quad (7-24)$$

Nas simulações de fluidos multifásicos utilizamos o núcleo quadrático

$$W(\mathbf{x} - \mathbf{x}_j, h) = \alpha_d \cdot w \left( \frac{\|\mathbf{x} - \mathbf{x}_j\|}{h} \right) \quad \text{com}$$

$$w(q) = \begin{cases} \frac{3}{16}q^2 - \frac{3}{4}q + \frac{3}{4} & ; 0 \leq q \leq 2 \\ 0 & ; q > 2 \end{cases}, \quad (7-25)$$

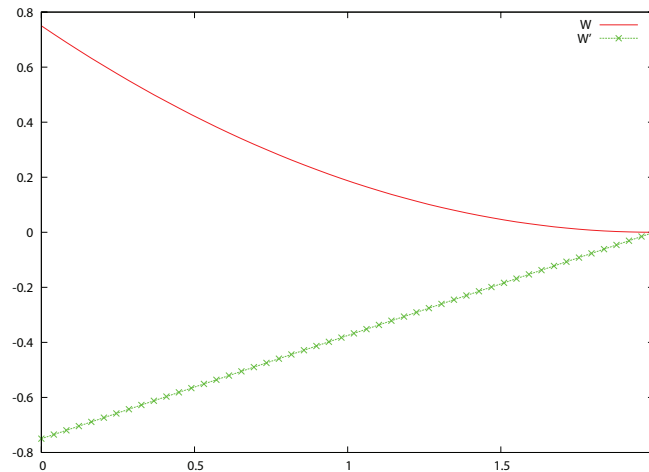


Figura 7.6: Gráfico do núcleo quadrático e de sua derivada de primeira ordem.

onde a constante de normalização do núcleo  $\alpha_d$  é  $1/h$ ,  $2/\pi h^2$  e  $5/4\pi h^3$  em espaços 1D, 2D e 3D, respectivamente. Apesar do núcleo quadrático fornecer uma aproximação de baixa ordem, esse núcleo é eficiente e dispensa o uso da correção XSPH (Seção 6.4.2), pois diferentemente das splines que em geral se anulam na origem (veja Figura 4.1) a derivada do núcleo quadrático alcança seu pico exatamente na origem (veja Figura 7.6). A dinâmica das partículas do método grid-partícula é dado resumidamente através do Algoritmo 4.

---

**Algoritmo 4** Método híbrido grid-partícula

---

- 1: **repeat**
  - 2:   **for** cada partícula  $i$  **do**
  - 3:     Calcule a velocidade  $\mathbf{v}_i^*$  e a posição  $\mathbf{x}_i^*$ . (equações (7-7) e (7-8))
  - 4:   **end for**
  - 5:   Construa o grid  $b$ . (equação (7-18))
  - 6:   Inicialize  $P^{(0)} \leftarrow 0$ .
  - 7:   **for**  $k = 0$  até  $k_{max}$  **do**
  - 8:     **for** cada vértice  $(i_g, j_g)$  dos grids  $P$  e  $b$  **do**
  - 9:      Calcule  $P_{i_g, j_g}^{(k)} \leftarrow \text{Jacobi}(P^{(k-1)}, b)$ . (equação (7-20))
  - 10:    **end for**
  - 11:    Aplique a condição de Neumann em  $P$ .
  - 12:   **end for**
  - 13:   **for** cada partícula  $i$  **do**
  - 14:     Calcule  $p_i$  fazendo uma interpolação em  $P$ .
  - 15:   **end for**
  - 16:   **for** cada partícula  $i$  **do**
  - 17:     Atualize  $\mathbf{v}_i^{n+1}$  e  $\mathbf{x}_i^{n+1}$ . (equações (7-12) e (7-13))
  - 18:   **end for**
  - 19:   Atualize  $\Delta t$  usando a condição CFL. (equação (7-24))
  - 20:    $tempo = tempo + \Delta t$
  - 21: **until**  $tempo < tempo_{total}$
-

## 7.4

### Resultados

Aqui apresentaremos alguns resultados para ilustrar o desempenho do método híbrido proposto em simulações de fenômenos clássicos de fluidos multifásicos, tais como as instabilidades de Rayleigh–Taylor e o movimento hidrodinâmico de uma gota.

#### 7.4.1

##### Instabilidades de Rayleigh–Taylor

As instabilidades de Rayleigh–Taylor são criadas quando um fluido mais denso é colocado sobre um fluido menos denso e esse estado de equilíbrio é perturbado devido à força da gravidade e pela diferença de densidade entre os fluidos. Como exemplo, simularemos essa instabilidade através de um caso básico de fluido bifásico. No estado inicial desse fenômeno, um fluido mais pesado com densidade  $\rho_H = 3$  é colocado sobre um fluido mais leve com densidade  $\rho_L = 1$  no interior de um canal de largura 1 e altura 2. A aceleração da gravidade é dada por  $g = 10$  e ambos os fluidos possuem a mesma viscosidade  $\mu = 0.001$ . Uma perturbação senoidal de amplitude 0.1 é feita na interface entre os dois fluidos para acelerar o processo de mistura. Por causa da força da gravidade, o fluido mais pesado é acelerado para baixo enquanto que o fluido mais leve sobe. Esse movimento resulta no surgimento de vórtices de velocidade que geram uma forma de cogumelo invertido na interface entre os fluidos. Todas as quantidades mencionadas acima são adimensionais. Os dois fluidos são discretizados utilizando 6000 partículas SPH com um grid para o cálculo da pressão de dimensão  $15 \times 29$  (veja Figura 7.7).

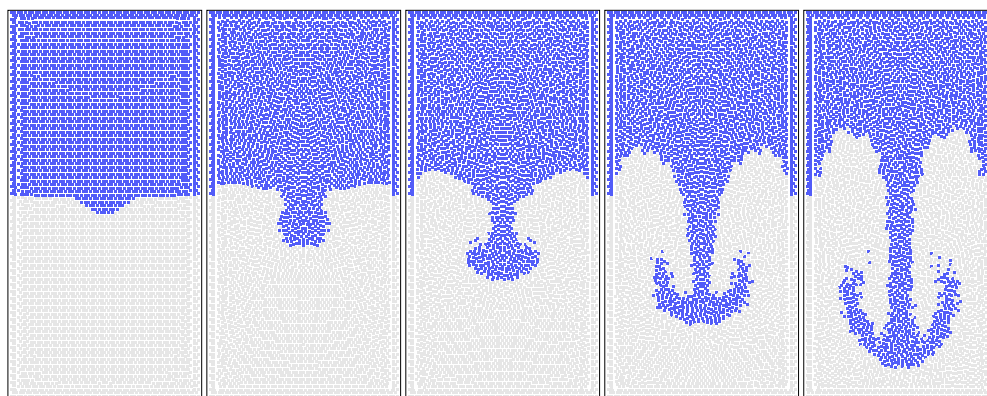


Figura 7.7: Instabilidade de Rayleigh–Taylor formada entre dois fluidos estratificados utilizando 6000 partículas: um mais denso (azul) e um menos denso (cinza). Da esquerda para direita, as imagens acima mostram a evolução da instabilidade nos instantes: 0.0s, 0.8s, 1.2s, 1.7s e 2.1s.



A altura do envelope da cabeça da instabilidade é dada analiticamente por [71, 61]

$$h(t) = \alpha A g t^2, \quad (7-26)$$

com  $\alpha \approx 0.06$  para simulações de fluidos incompressíveis com viscosidade  $\mu = 0.001$  e

$$A = \frac{\rho_H - \rho_L}{\rho_H + \rho_L}. \quad (7-27)$$

O parâmetro  $A$  é conhecido como o *número de Atwood*. Pela equação (7-26) temos que a altura do envelope cresce quadraticamente em relação ao tempo, a comparação da solução analítica com os resultados numéricos obtidos com o nosso método é dada através da Figura 7.8. O tempo médio de execução de cada passo dessa simulação num computador com processador Centrino 1.86 GHz foi de 0.07 segundos.

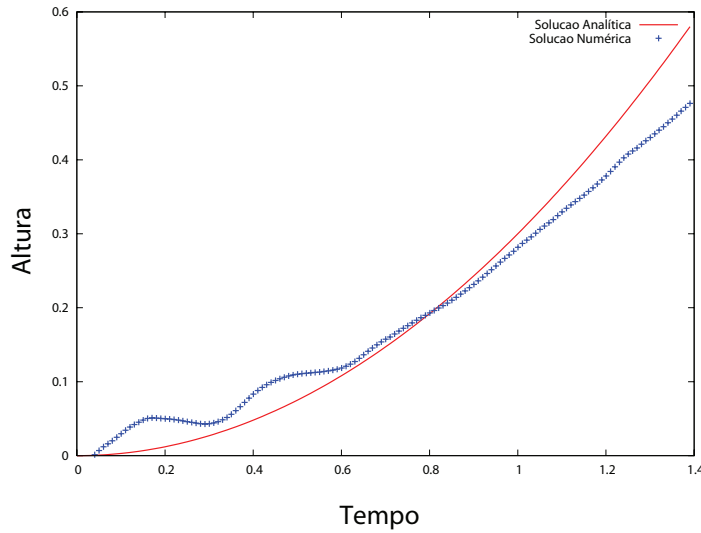
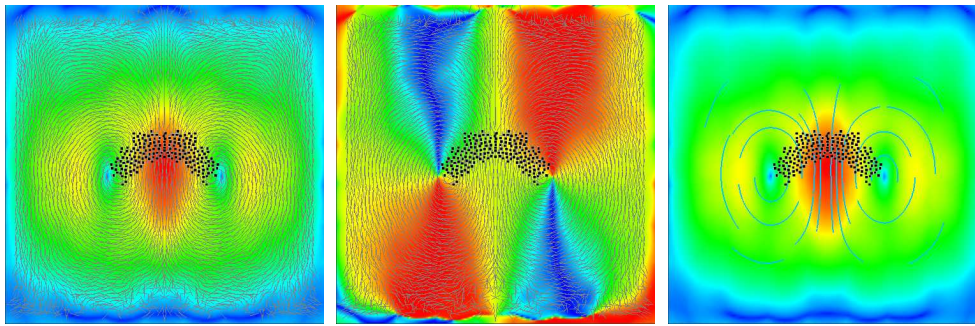


Figura 7.8: Evolução da altura da instabilidade de Rayleigh–Taylor: comparação da simulação numérica com a solução analítica.



7.9(a): Campo de velocidade      7.9(b): Campo de fase      7.9(c): Linhas de fluxo

Figura 7.9: Campo vetorial da velocidade: (a) As cores representam a magnitude da velocidade do fluido, variando de baixa velocidade (azul) para alta (vermelho). (b) No campo de fase as cores representam a variação do cosseno do ângulo formado pelo vetor velocidade e o eixo das abscissas indo de -1 (vermelho) até 1 (azul). (c) Os vórtices são melhores identificados através das linhas de fluxos.

#### 7.4.2

##### Hidrodinâmica de uma gota

Utilizamos o nosso método para simular o movimento hidrodinâmico de uma gota de um fluido mais leve emergindo dentro de um fluido mais denso. No estado inicial desse fenômeno, um fluido mais pesado com densidade  $\rho_H = 1$  preenche um reservatório quadrado de lado 1. No interior do fluido mais denso é colocada uma gota com densidade  $\rho_L = 0.3$  e de raio 0.1. A aceleração da gravidade é dada por  $g = 1$  e ambos os fluidos possuem a mesma viscosidade  $\mu = 0.001$ . Todas as quantidades mencionadas acima são adimensionais. A gota se move devido à diferença de densidade entre os fluidos, durante o movimento o fluido mais denso começa a penetrar a interface da gota gerando vórtices de velocidade que deformam sua forma esférica para a famosa forma de ferradura que é observada em experimentos reais (Figura 7.9). A simulação hidrodinâmica da gota é feita utilizando o método proposto com 6000 partículas SPH e um grid para o cálculo da pressão com dimensão  $21^2$  (veja Figura 7.10). A simulação é validada através de (7-26) e a comparação com a simulação pode ser vista na Figura (7.11). O tempo médio de execução de cada passo dessa simulação num computador com processador Centrino 1.86 GHz foi de 0.06 segundos.

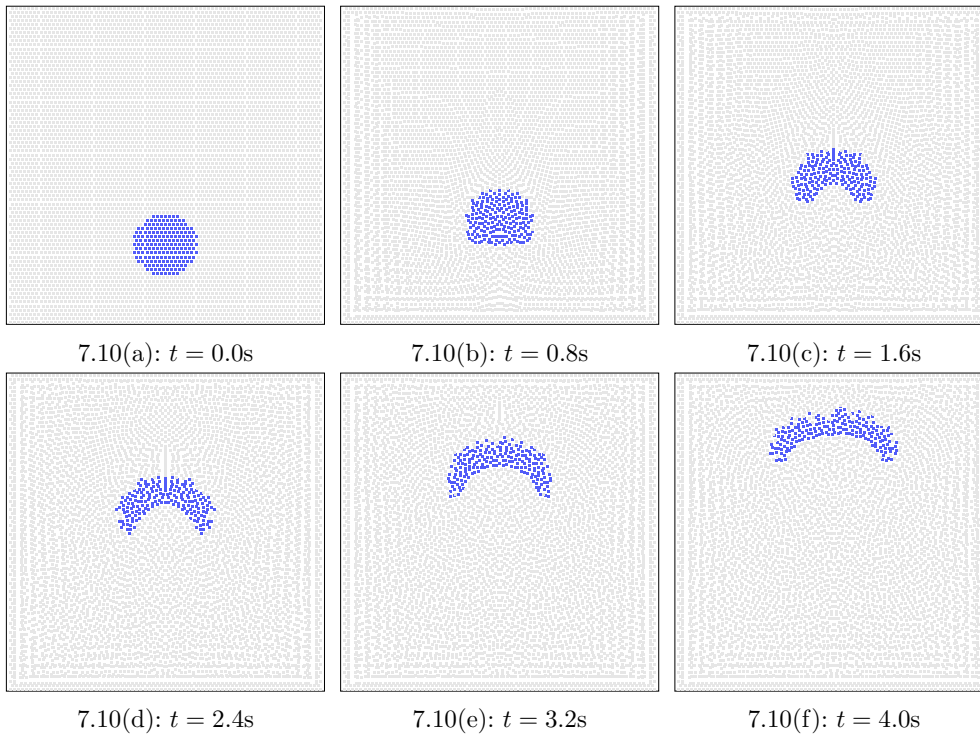


Figura 7.10: Movimento hidrodinâmico de uma gota de um fluido mais leve (azul) no interior de um fluido mais pesado (cinza) utilizando 6000 partículas.

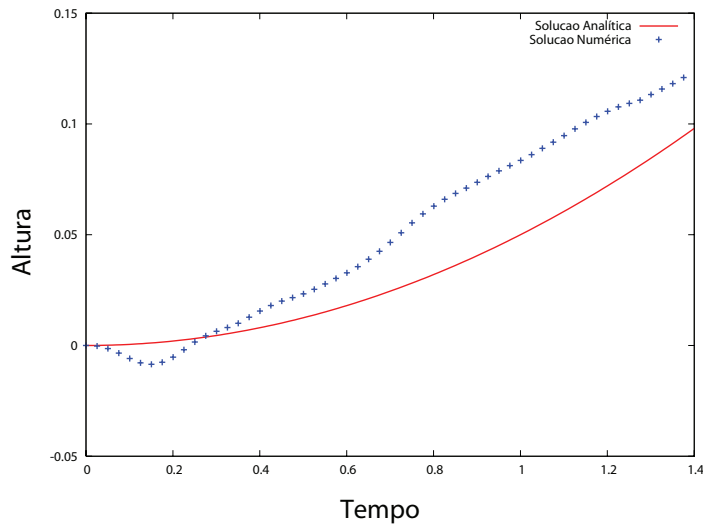


Figura 7.11: Evolução da altura de uma gota: comparação da simulação numérica com a solução analítica.

## 8

### Conclusão e trabalhos futuros

Nessa tese foram apresentados novos métodos de simulação física de fluidos viscoplásticos e multifásicos baseados na aproximação SPH das equações de Navier–Stokes.

Na simulação de fluidos viscoplásticos foi introduzida uma versão SPH do modelo de Fluido Newtoniano Generalizado. A flexibilidade desse modelo nos permitiu simular deformações plásticas assim como derretimento de objetos sólidos de forma visualmente realística. Também foi apresentada uma nova forma de impor condição de fronteira em sistemas SPH de natureza puramente geométrica dado por um simples teste de colisão. O teste de colisão capturou muito bem a influência de objetos sólidos nos fluidos, como por exemplo, na simulação de escoamento de lava sobre um terreno inclinado. A estabilidade numérica do método foi melhorada com a introdução de novas ferramentas numéricas e por uma nova aproximação SPH do operador laplaciano para o cálculo da equação do calor. A qualidade do método foi ilustrada através de exemplos simples que assemelham com a intuição física, tornando o método proposto ideal para animações e simulações de fluidos não-newtonianos.

Depois, foi introduzido um novo método híbrido que acopla SPH com o método de diferenças finitas para simular fluidos multifásicos. A combinação dessas duas técnicas de características tão distintas foi feita de forma elegante e eficiente. A incompressibilidade do SPH foi preservada através do cálculo implícito de uma equação de Poisson da pressão. Esse cálculo da pressão fez com que fosse eliminado o uso da velocidade do som na condição de estabilidade CFL permitindo que as partículas se desloquem através de um passo de tempo maior. Foi observado que, mesmo com o uso de passos de tempo maiores, o método conseguiu capturar a vorticidade gerada pela mistura “turbulenta” entre os fluidos de densidades distintas. A utilização do grid para discretizar a equação de Poisson nos possibilitou gerar sistemas lineares de baixas dimensões, isso se deve ao fato do grid não depender da posição e nem do número de partículas SPH. Foi testado o método híbrido em fenômenos como instabilidade de Rayleigh–Taylor e da hidrodinâmica de uma gota, a validação do método se deu de maneira satisfatória, pois o método reproduziu

bem o crescimento não-linear dessas instabilidades.

Essa tese pode ser estendida em várias direções. Do lado físico, podemos simular outras classes de fluidos não-newtonianos como os fluidos viscoelásticos. Acreditamos que pode ser melhorada a eficiência do método viscoplástico utilizando partículas que se adaptem de acordo com a força exercida sobre os objetos. Podemos utilizar outras técnicas de captura de superfície livre tais como level-sets e splatting. A visualização pode ser completada através do uso de um processo de síntese de textura dinâmica para objetos deformáveis [73]. O método híbrido pode ser naturalmente estendido para simulações 3D e simulações de fluidos miscíveis. Além disso, podemos melhorar a interpolação da pressão nas partículas, uso de um método iterativo mais sofisticado como o método do gradiente conjugado para resolver a equação de Poisson, assim como o cálculo explícito da tensão superficial. Finalmente, a não conectividade entre as partículas SPH permite que futuramente possamos paralelizar os nossos algoritmos e aumentar consideravelmente o número de partículas em nossas simulações. Além disso, podemos portar os nossos códigos com objetivo de utilizar os recursos eficientes e altamente paralelizáveis das placas gráficas através da GPU (Graphics Processing Unit) e o recente microprocessador dedicado a efeitos físicos desenvolvido pela AGEIA Technologies conhecido como *Physics Processing Unit* (PPU) (<http://www.ageia.com>).

## Referências Bibliográficas

- [1] AGUIAR, C. E.; KODAMA, T.; OSADA, T. ; HAMA, Y.. **Smoothed particle hydrodynamics for relativistic heavy-ion collisions**. Journal of Physics G, 27:75–94, 2001.
- [2] AKENINE-MÖLLER, T.. **Fast 3d triangle–box overlap testing**. Journal of Graphics Tools, 6(1):29–33, 2001.
- [3] ANDERSON, J. D.. **Computational Fluid Dynamics**. McGraw–Hill, 1995.
- [4] BALBUENA, P.; SEMINARIO, J.. **Molecular Dynamics**. Elsevier Science, 1999.
- [5] BARAFF, D.. **Interactive simulation of solid rigid bodies**. IEEE Computer Graphics & Applications, 15(3):63–75, 1995.
- [6] BARGTEIL, A. W.; WOJTAN, C.; HODGINS, J. K. ; TURK, G.. **A finite element method for animating large viscoplastic flow**. ACM Transactions on Graphics, 26(3), 2007.
- [7] BATCHELOR, G. K.. **An Introduction to Fluid Dynamics**. Cambridge University Press, 3rd edition, 2000.
- [8] BENSON, D. J.. **Computational methods in lagrangian and eulerian hydrocodes**. Computer Methods in Applied Mechanics and Engineering, 99(2-3):235–394, 1992.
- [9] BRENNEN, C. E.. **Fundamentals of Multiphase Flow**. Cambridge University Press, 2005.
- [10] CARLSON, M.; MUCHA, P.; VAN HORN III, B. ; TURK, G.. **Melting and flowing**. In: SYMPOSIUM ON COMPUTER ANIMATION, p. 167–174, 2002.
- [11] CHORIN, A. J.; MARSDEN, J. E.. **A Mathematical Introduction to Fluid Dynamics**. Springer, 2nd edition, 2000.

- [12] CLAVET, S.; BEAUDOIN, P. ; POULIN, P.. **Particle-based viscoelastic simulation**. Symposium on Computer Animation, p. 219–228, 2005.
- [13] CLEARY, P. W.; MONAGHAN, J. J.. **Conduction modelling using smoothed particle hydrodynamics**. Journal of Computational Physics, 148:227–264, 1999.
- [14] COURANT, R.; FRIEDRICHS, K. ; LEWY, H.. **On the partial difference equations of mathematical physics**. IBM Journal, 11:215–234, 1967.
- [15] CUMMINS, S. J.; RUDMAN, M.. **An SPH projection method**. Journal of Computational Physics, 152:584–607, 1999.
- [16] DE SOUSA, F. S.; MANGIAVACCHI, N.; NONATO, L.; CASTELO, A.; TOMÉ, M.; FERREIRA, V. G.; CUMINATO, J. A. ; MCKEE, S.. **A front-tracking/front-capturing method for the simulation of 3D multi-fluid flows with free surfaces**. Journal of Computational Physics, 198:469–499, 2004.
- [17] DESBRUN, M.; CANI, M. P.. **Smoothed particles: A new paradigm for animating highly deformable bodies**. In: COMPUTER ANIMATION AND SIMULATION '96, p. 61–76. Proceedings of EG Workshop on Animation and Simulation, Springer-Verlag, 1996.
- [18] EBERLY, D. H.. **Game Physics**. Morgan Kaufmann, 2004.
- [19] ETZMUSS, O.; KECKEISEN, M. ; STRASSER, W.. **A fast finite element solution for cloth modelling**. In: 11TH PACIFIC CONFERENCE ON COMPUTER GRAPHICS AND APPLICATIONS, p. 244–251, 2003.
- [20] FEDKIW, R.; STAM, J. ; JENSEN, H. W.. **Visual simulation of smoke**. In: PROCEEDINGS OF ACM SIGGRAPH 2001, Computer Graphics Proceedings, Annual Conference Series, p. 15–22, 2001.
- [21] FOLEY, J. D.; VAN DAM, A.; FEINER, S. K. ; HUGHES, J. F.. **Computer Graphics: Principles and Practice in C**. Addison-Wesley, 2nd edition, 1995.
- [22] FOSTER, N.; METAXAS, D.. **Realistic animation of liquids**. In: GRAPHICS INTERFACE '96, p. 204–212, 1996.
- [23] GAVETE, L.; RUIZ, A.; BENITO, J. J. ; FALCON, S.. **Implementation of essential boundary conditions in a meshless method**. Communications in Numerical Methods in Engineering, 16(6):409–421, 2000.

- [24] GINGOLD, R. A.; MONAGHAN, J. J.. **Smoothed particle hydrodynamics: theory and application to non-spherical stars.** Monthly Notices of the Royal Astronomical Society, 181:375–389, 1977.
- [25] GOKTEKIN, T. G.; BARGTEIL, A. W. ; O'BRIEN, J. F.. **A method for animating viscoelastic.** ACM Transactions on Graphics, 23(3):463–468, 2004.
- [26] HARLOW, F. H.; WELCH, J. E.. **Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface.** The Physics of Fluids, 8:2182–2189, 1965.
- [27] HIRT, C. W.; NICHOLS, B. D.. **Volume of fluid (VOF) method for the dynamics of free boundaries.** Journal of Computational Physics, 39(1):201–225, 1981.
- [28] IORIO, R. J.; DE MAGALHÃES IORIO, V.. **Fourier Analysis and Partial Differential Equations: An Introduction.** Cambridge University Press, 2001.
- [29] KARABASSI, E.-A.; PAPAIOANNOU, G.; THEOHARIS, T. ; BOEHM, A.. **Intersection test for collision detection in particle systems.** Journal of Graphics Tools, 4(1):25–37, 1999.
- [30] KEISER, R.; ADAMS, B.; GASSER, D.; BAZZI, P.; DUTRÉ, P. ; GROSS, M.. **A unified lagrangian approach to solid-fluid animation.** In: SYMPOSIUM ON POINT-BASED GRAPHICS, p. 125–134, 2005.
- [31] KLINGNER, B. M.; FELDMAN, B. E.; CHENTANEZ, N. ; O'BRIEN, J. F.. **Fluid animation with dynamic meshes.** ACM Transactions on Graphics, 25(3):820–825, 2006.
- [32] KODAMA, T.; AGUIAR, C. E.; OSADA, T. ; HAMA, Y.. **Entropy-based relativistic smoothed particle hydrodynamics.** Journal of Physics G, 27:557–560, 2001.
- [33] KOSHIZUKA, S.; NOBE, A. ; OKA, Y.. **Numerical analysis of breaking waves using the moving particle semi-implicit method.** International Journal for Numerical Methods in Fluids, 27(7):751–769, 1998.
- [34] LANGETEPE, E.; ZACHMANN, G.. **Geometric Data Structures for Computer Graphics.** A K Peters, 2006.
- [35] LENGYEL, J.. **Real-time hair.** In: RENDERING TECHNIQUES 2000: 11TH EUROGRAPHICS WORKSHOP ON RENDERING, p. 243–256, 2000.



- [36] LEWINER, T.; LOPES, H.; VIEIRA, A. W. ; TAVARES, G.. **Efficient implementation of marching cubes with topological guarantees.** *Journal of Graphics Tools*, 8(2):234–241, 2003.
- [37] LI, S.; LIU, W. K.. **Meshfree Particle Methods.** Springer, 2004.
- [38] LIU, G. R.; LIU, M. B.. **Smoothed Particle Hydrodynamics.** World Science, 2005.
- [39] LIU, J.; KOSHIZUKA, S. ; OKA, Y.. **A hybrid particle–mesh method for viscous, incompressible, multiphase flows.** *Journal of Computational Physics*, 202:65–93, 2005.
- [40] LORENSEN, W. E.; CLINE, H. E.. **Marching cubes: A high resolution 3d surface construction algorithm.** In: *PROCEEDINGS OF SIGGRAPH 87*, p. 163–169, 1987.
- [41] LOSASSO, F.; SHINAR, T.; SELLE, A. ; FEDKIW, R.. **Multiple interacting liquids.** *ACM Transactions on Graphics*, 25(3):812–819, 2006.
- [42] LUCY, L. B.. **Numerical approach to testing the fission hypothesis.** *Astronomical Journal*, 82:1013–1024, 1977.
- [43] MAIR, H. U.. **Review: Hydrocodes for structural response to underwater explosions.** *Shock and Vibration*, 6(2):81–96, 1999.
- [44] MAO, H.; YANG, Y.. **A particle-based model for non-newtonian fluid.** Technical Report TR05-21, Department of Computing Science, University of Alberta, August 2005. <http://www.cs.ualberta.ca/TechReports/2005/TR05-21/TR05-21.pdf>.
- [45] MENDES, P. R. S.; DUTRA, E. S. S.; SIFFERT, J. R. R. ; NACCACHE, M. F.. **Gas displacement of viscoplastic liquids in capillary tubes.** *Journal of Non-Newtonian Fluid Mechanics*, 142:1–11, 2007.
- [46] MONAGHAN, J. J.. **On the problem of penetration in particle methods.** *Journal of Computational Physics*, 82:1–15, 1989.
- [47] MONAGHAN, J. J.. **Simulating free surface flow with SPH.** *Journal of Computational Physics*, 110:399–406, 1994.
- [48] MONAGHAN, J. J.. **Smoothed particle hydrodynamics.** *Reports on Progress in Physics*, 68:1703–1759, 2005.

- [49] MORRIS, J. P.. **Simulating surface tension with smoothed particle hydrodynamics**. International Journal for Numerical Methods in Fluids, 33(3):333–353, 2000.
- [50] MORRIS, J. P.; FOX, P. J. ; ZHU, Y.. **Modeling low Reynolds number for incompressible flows using SPH**. Journal of Computational Physics, 136:214–226, 1997.
- [51] MÜLLER, M.; CHARYPAR, D. ; GROSS, M.. **Particle-based fluid simulation for interactive applications**. In: SYMPOSIUM ON COMPUTER ANIMATION, p. 154–159, 2003.
- [52] MÜLLER, M.; KEISSER, R.; NEALEN, A.; PAULY, M.; GROSS, M. ; ALEXA, M.. **Point based animation of elastic, plastic and melting**. Symposium on Computer Animation, p. 141–151, 2004.
- [53] MÜLLER, M.; SCHIRM, S.; TESCHNER, M.; HEIDELBERGER, B. ; GROSS, M.. **Interaction of fluids with deformable solids**. Journal of Computer Animation and Virtual Worlds, 15(3–4):159–171, 2004.
- [54] O'BRIEN, J. F.; BARGTEIL, A. W. ; HODGINS, J. K.. **Graphical modeling and animation of ductile fracture**. ACM Transactions on Graphics, 21(3):291–294, 2002.
- [55] O'ROURKE, J.. **Computational Geometry in C**. Cambridge University Press, 2000.
- [56] OSHER, S. J.; FEDKIW, R. P.. **Level Set Methods and Dynamic Implicit Surfaces**. Springer, 2002.
- [57] OWENS, R. G.; PHILLIPS, T. N.. **Computational Rheology**. Imperial College Press, 2002.
- [58] PAIVA, A.; LOPES, H.; LEWINER, T. ; DE FIGUEIREDO, L. H.. **Robust adaptive meshes for implicit surfaces**. In: 19TH BRAZILIAN SYMPOSIUM ON COMPUTER GRAPHICS AND IMAGE PROCESSING, p. 205–212, 2006.
- [59] PAIVA, A.; PETRONETTO, F.; LEWINER, T. ; TAVARES, G.. **Particle-based non-Newtonian fluid animation for melting objects**. In: 19TH BRAZILIAN SYMPOSIUM ON COMPUTER GRAPHICS AND IMAGE PROCESSING, p. 78–85, 2006.

- [60] PAIVA, A.; PETRONETTO, F.; LEWINER, T. ; TAVARES, G.. **Simulating viscoplastic fluid using smoothed particle hydrodynamic**. In: III BRAZILIAN CONFERENCE ON RHEOLOGY, p. 93–94, 2006.
- [61] READ, K. I.. **Experimental investigation of turbulent mixing by Rayleigh–Taylor instability**. *Physica D*, 12:45–58, 1984.
- [62] RIDER, W.. **Approximate projection methods for incompressible flows: implementation, variants and robustness**. Technical Report LA-UR-2000, Los Alamos National Laboratory, 1995. <http://citeseer.ist.psu.edu/rider95approximate.html>.
- [63] SAAD, Y.. **Iterative Methods for Sparse Linear Systems**. SIAM, 2nd edition, 2003.
- [64] SCHAEFER, S.; WARREN, J.. **Dual marching cubes: primal contouring of dual grids**. In: 12TH PACIFIC CONFERENCE ON COMPUTER GRAPHICS AND APPLICATIONS, p. 70–76, 2004.
- [65] STAM, J.. **Stable fluids**. In: PROCEEDINGS OF SIGGRAPH 99, Computer Graphics Proceedings, Annual Conference Series, p. 121–128, 1999.
- [66] STORA, D.; AGLIATI, P.-O.; CANI, M.-P.; NEYRET, F. ; GASCUEL, J.-D.. **Animating lava flows**. In: GRAPHICS INTERFACE, p. 203–210, 1999.
- [67] TARTAKOVSKY, A. M.; MEAKIN, P.. **A smoothed particle hydrodynamics model for miscible flow in three–dimensional fractures and the two–dimensional Rayleigh–Taylor instability**. *Journal of Computational Physics*, 207:610–624, 2005.
- [68] THOMPSON, J. F.; SONI, B. ; WEATHERILL, N. P.. **Handbook of Grid Generation**. CRC Press, 1998.
- [69] WEI, X.; LI, W. ; KAUFMAN, A.. **Melting and flowing of viscous volumes**. *Computer Animation and Social Agents (CASA)*, p. 54–59, 2003.
- [70] WESSELING, P.. **An Introduction to Multigrid Methods**. Wiley, 2004.
- [71] YOUNG, D.. **Numerical simulation of turbulent mixing by Rayleigh–Taylor instability**. *Physica D*, 12:32–44, 1984.
- [72] ZIENKIEWICZ, O. C.; TAYLOR, R. L. ; ZHU, J.. **The Finite Element Method: Its Basis and Fundamentals**. Butterworth-Heinemann, 6th edition, 2005.

- [73] ZONENSCHIN, R.; GOMES, J.; VELHO, L.; DE FIGUEIREDO, L. H.; TIGGES, M. ; WYVILL, B.. **Texturing composite deformable implicit objects**. In: PROCEEDINGS OF SIBGRAPI '98, p. 346–353, 1998.

# 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](#)