



Simulação de Partículas: Usando CUDA e Ferramentas de Visualização e interação interativa por OpenGL

¹Edgar Andrés Patiño Nariño

Departamento de Mecânica Computacional – DMC

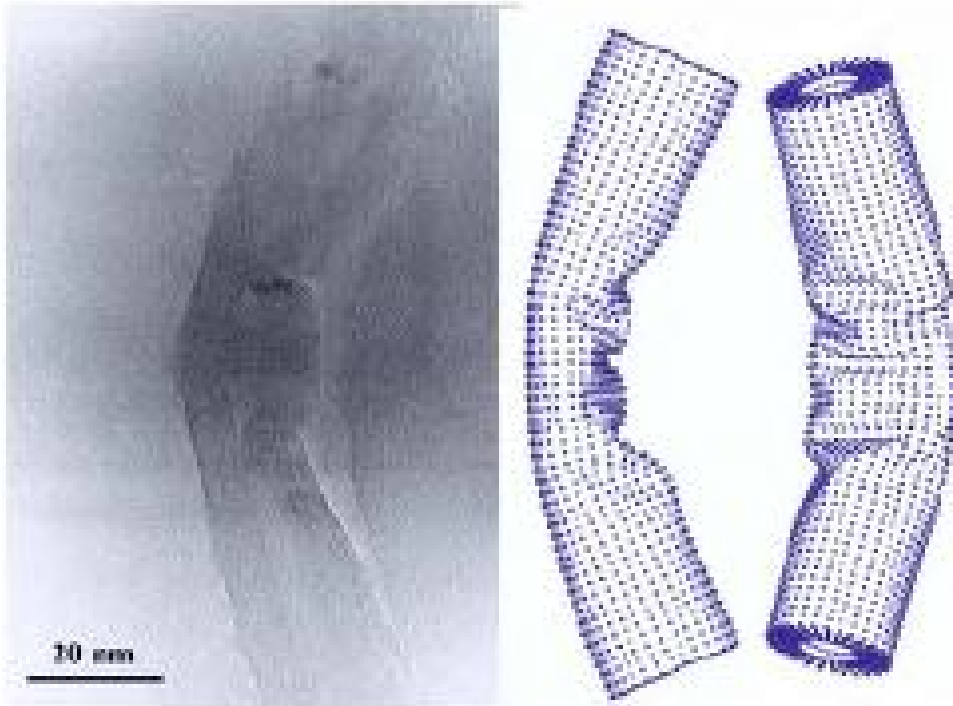
Faculdade de Engenharia Mecânica – FEM

Universidade Estadual de Campinas - UNICAMP

[¹eapatinon@fem.unicamp.br](mailto:eapatinon@fem.unicamp.br)

Introdução

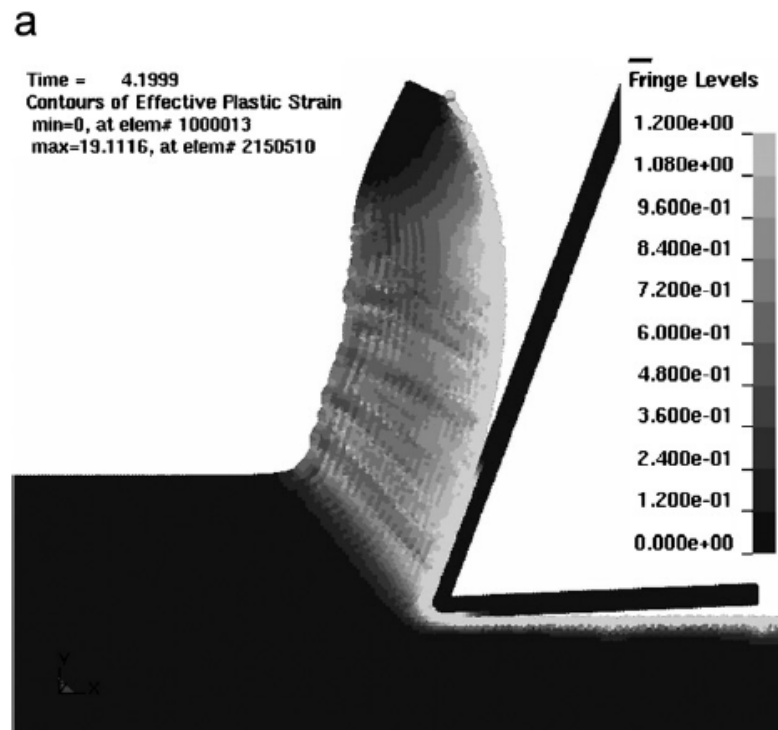
Os métodos de partículas [1] são um técnica comumente usada pela simulação de problemas físicos [2].



Observação experimental de de encurvadura de nanotubos de carbono de paredes múltiplas (b) Simulado flambagem padrão MD em 2D. (c) Flambagem padrão da nanoestrutura MD em 3D [2].

Introdução

Os métodos de partículas [1] são um técnica comumente usada pela simulação de problemas físicos [2].



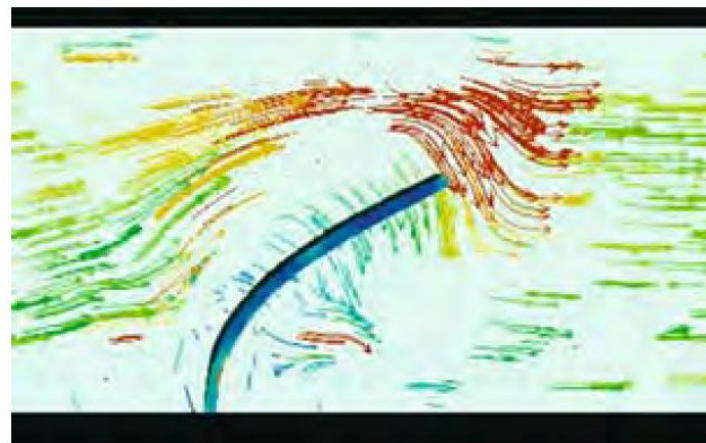
Limido, 2009, Usinagem
Ortogonal
SPH [3].

Introdução

Os métodos de partículas [1] são um técnica comumente usada pela simulação de problemas físicos [2].



(a) Experiment

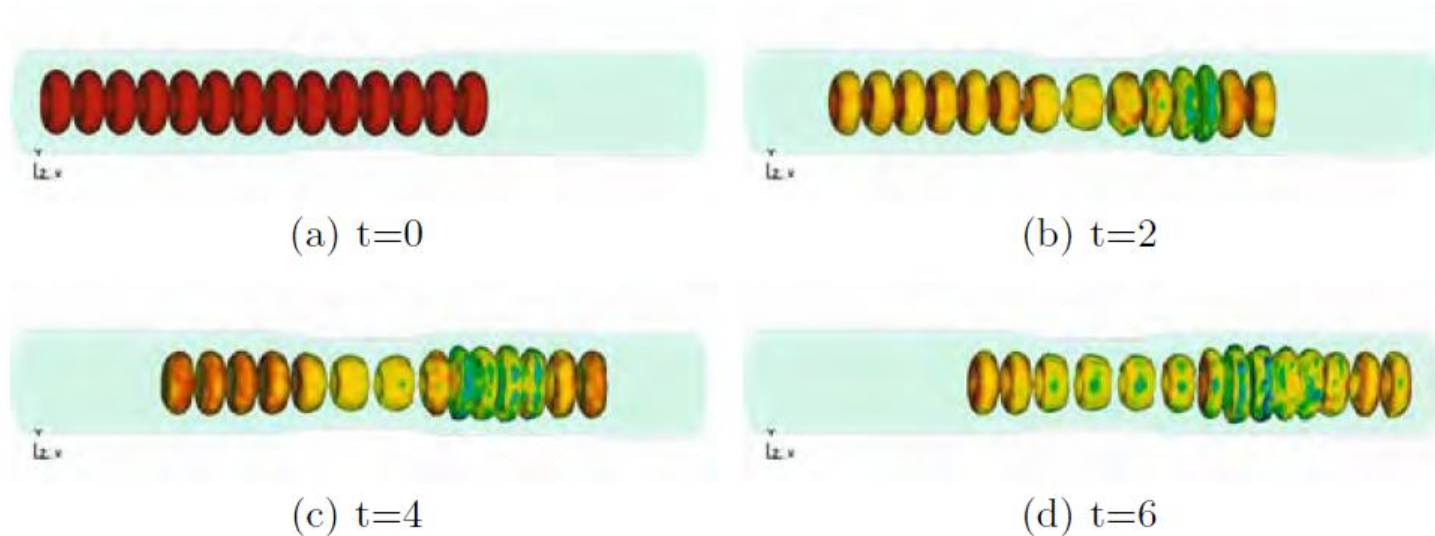


(b) Simulation

Comparação de experimental e resultado de uma simulação de uma lamina de borracha em deflexão por uma coluna de água [2].

Introdução

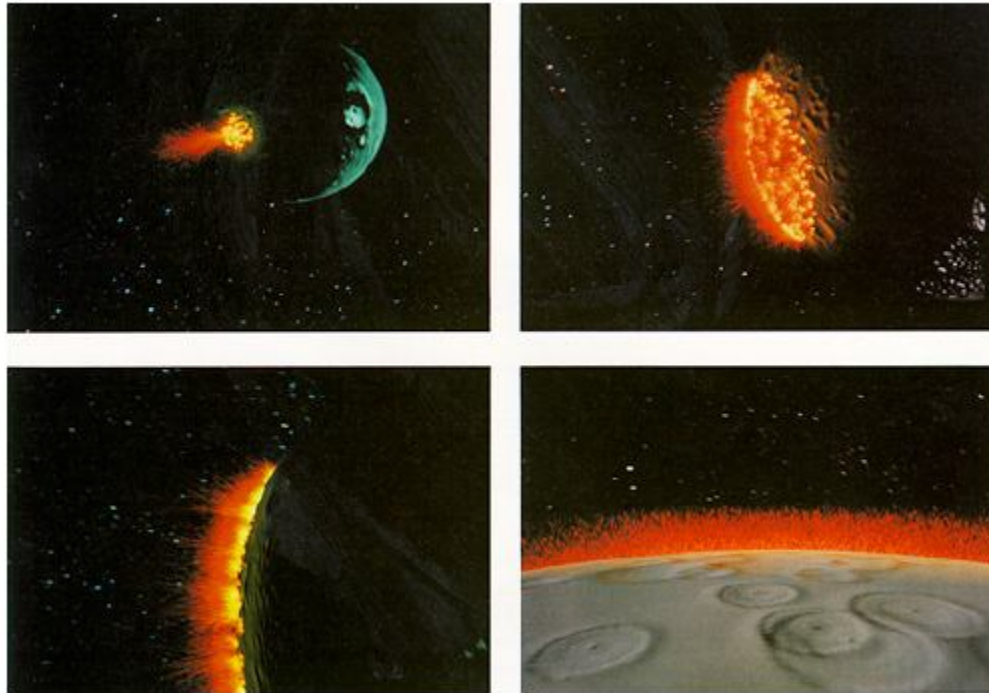
Os métodos de partículas [1] são um técnica comumente usada pela simulação de problemas físicos [2].



O fluxo normal das células vermelhas do sangue, com a velocidade de entrada de 10 m / s, em intervalos de tempo diferentes [2].

Introdução

Os métodos de partículas [1] são um técnica comumente usada pela simulação de problemas físicos [2].



*“Particle system: A technique for modeling a class of fuzzy objects”
Rees 1983 [4].*

Star Trek (Jornada nas estrelas): The Wrath of Khan

Simulação de fluidos baseados em partículas

Vantagem

- A conservação de massa é trivial.
- Fácil aplicação no problemas de superfície livre.
- Apenas executa cálculos quando é necessário.
- Não precisa de malha.
- Fácil de paralelizar.

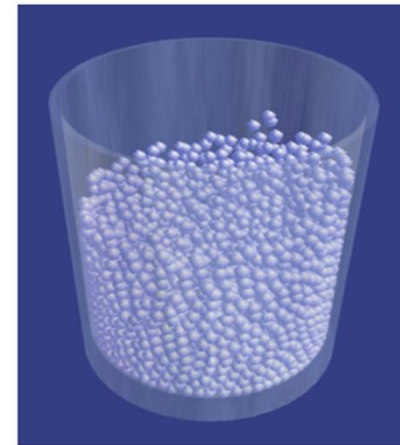
Desvantagem

- Precisa de um longo número de partículas para resultados realistas .

Motivação

Métodos de partículas como é DPD

Outro grande é que **são projetados para ser métodos adimensionais e as partículas ficam muito dispersas**



Complicando a estabilização e estudo dos dados das partículas

Objetivos

Mostrar as vantagens de usar as ferramentas de visualização e iteração usando **OpenGL** e **GLUT** para o estudo de simuladores de partículas **DPD**, em simulações em tempo real.

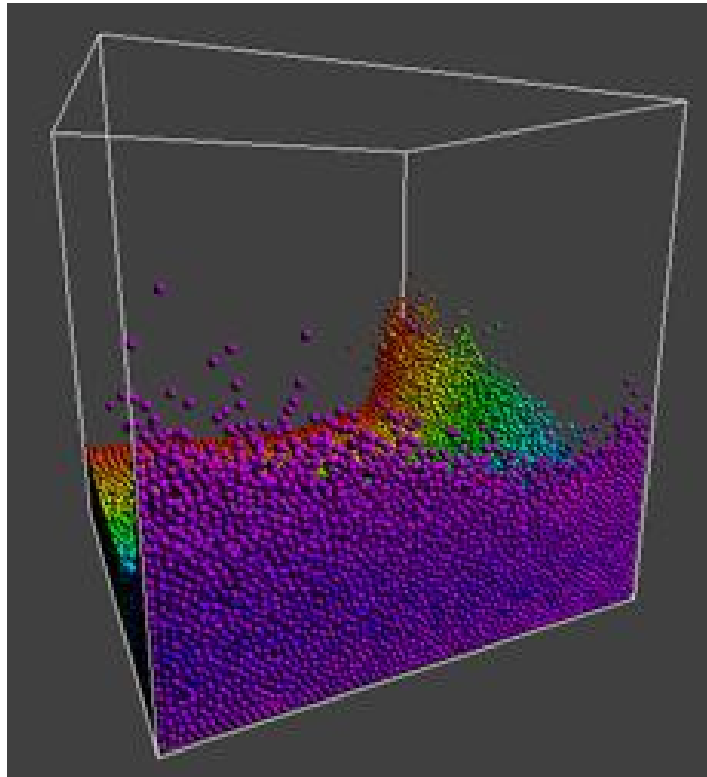
Modificar o código em CUDA "ParticleSimulation using CUDA" (Prof. Simon Green):

- **Função que transforma** os valores da grandeza das **velocidades das partículas em cores RGBA**.

Objetivos

- **Planejamento de novos exemplos:**
Escorregamento contínuo de partículas, queda de coluna de fluido e interação com obstáculo.
- ***Aplicação de Ferramentas interativas:***
 - Rotação
 - Zoom
 - Deslocamento longitudinal, vertical e transversal.
 - Menus auxiliares e comandos por teclas.

Particle Simulation using CUDA



<http://www.youtube.com/watch?v=RqduA7myZok>

Particle Simulation using CUDA

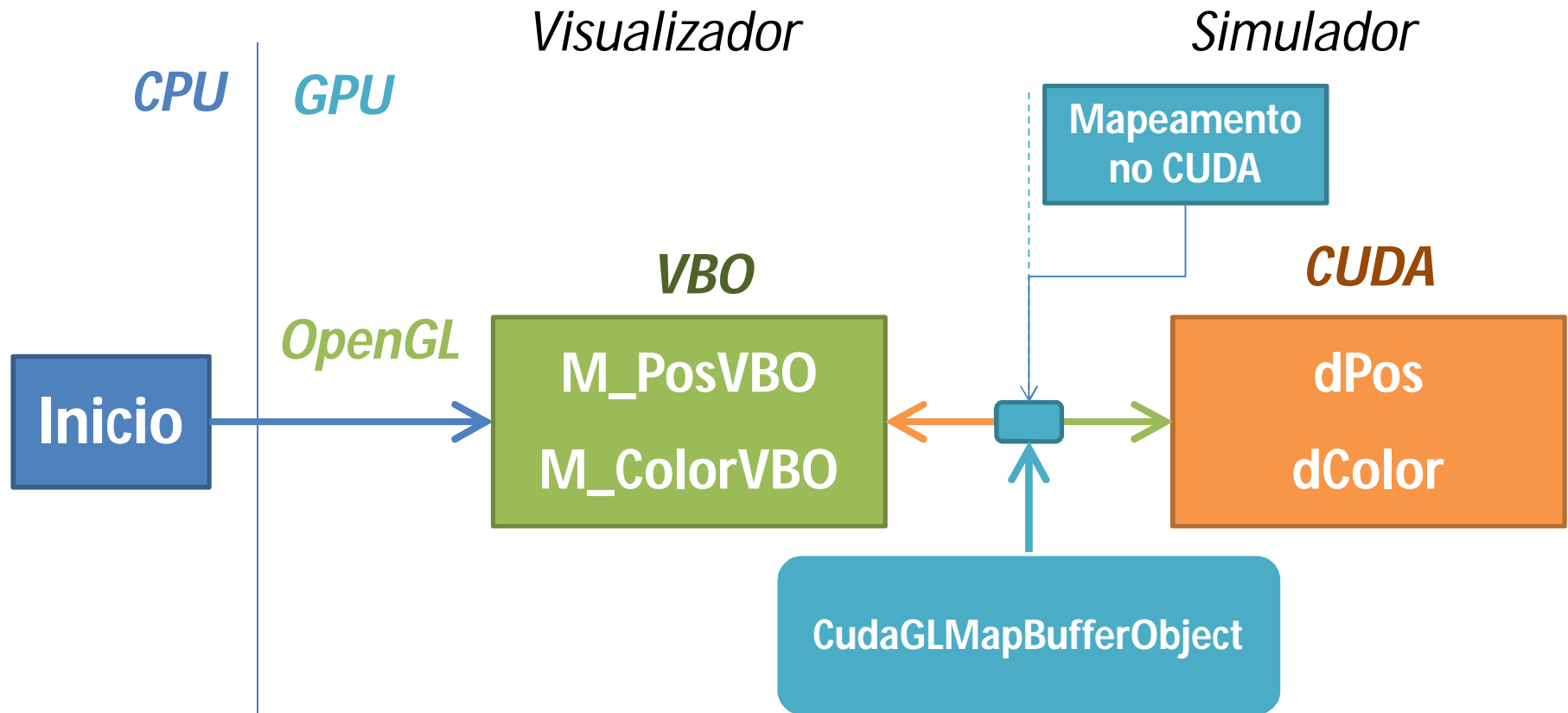
Implementação

Há três passos principais no desenvolvimento da simulação:

1. Integração temporal.
2. Interação Partícula-Partícula (pesquisador partículas).
3. Processamento de colisão: *Dissipative Particle Dynamics (DPD)*.

O **renderizado** das partículas é realizado usando OpenGL, fazendo uso do **Point Sprites** e um **GLSL** (OpenGL Shading Language) **Pixel Shader** que faz os pontos tornar-se esferas.

Interação CUDA-OpenGL



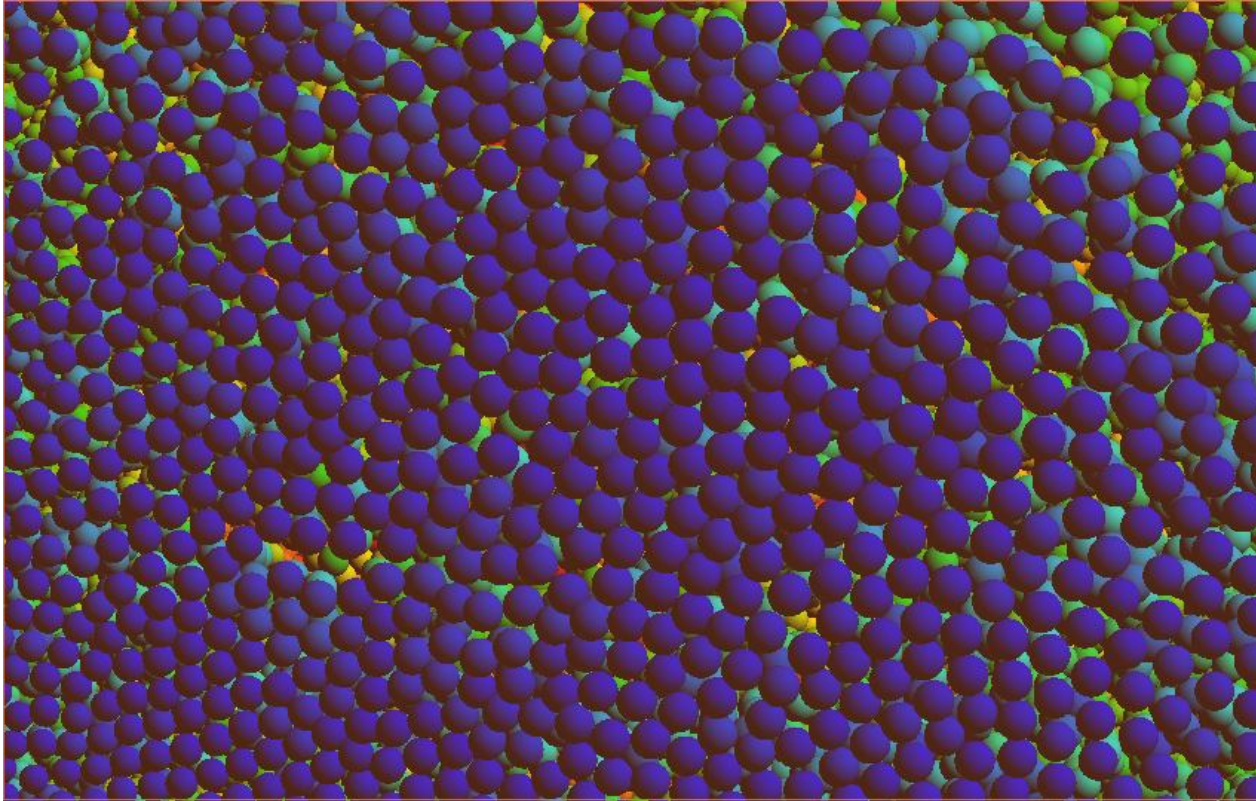
***Onde Color esta relacionada a Velocidade nas Partículas**

Renderizado das partículas

O renderizado das partículas é realizado pela máquina de estados OpenGL, fazendo uso do *Point Sprites*.

- Com Point Sprites você pode colocar uma imagem de textura 2D em qualquer lugar na tela, desenhando um único ponto 3D.

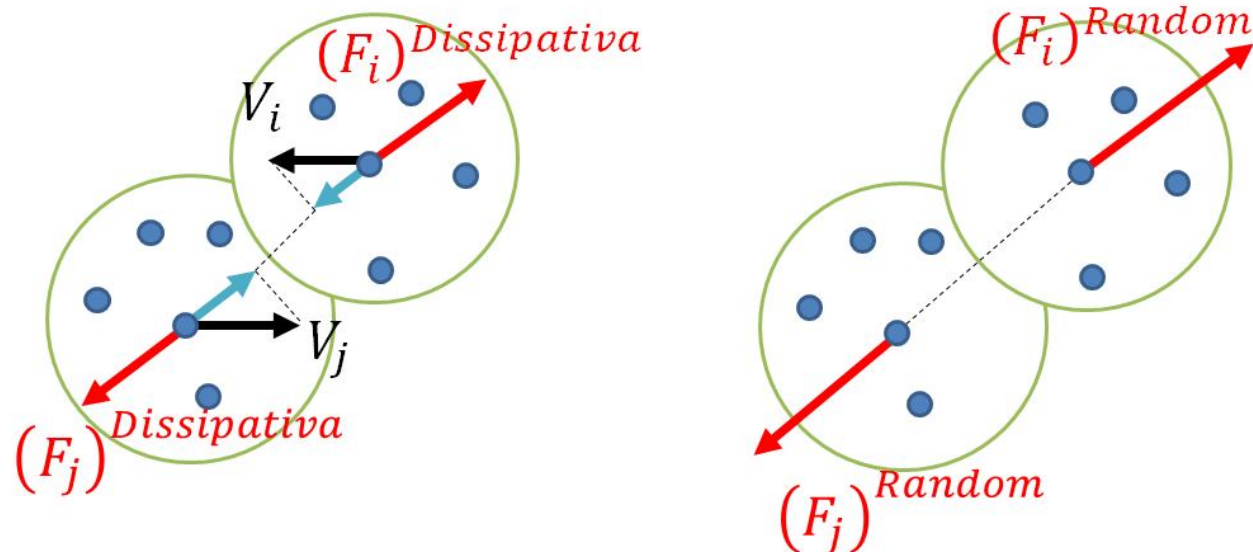
Renderizado das partículas



Point Sprites

Processamento de colisão

Dissipative Particle Dynamics



A **força Dissipativa** (atrito) reduz a relação entre pares de partículas. A **força Aleatória** (Random) compensa os graus de liberdade eliminados.

As **força Dissipativas e Aleatórias** formam o termostato do DPD. As grandezas das **força Dissipativas e Aleatórias** são definidas pelo teorema de **Flutuação-dissipação**.

Processamento de colisão

Dissipative Particle Dynamics

$$d\mathbf{r}_i = \mathbf{v}_i dt$$

$$m_i d\mathbf{v}_i = \sum_{i \neq j} \mathbf{F}_{ij}^{(C)}(r_{ij}) \vec{e}_{ij} dt - \gamma \sum_{i \neq j} \omega^D(r_{ij}) (\mathbf{v}_{ij} \cdot \vec{e}_{ij}) \vec{e}_{ij} dt$$

$$+ \sigma \sum_{i \neq j} \omega^R(r_{ij}) \xi_{ij} \vec{e}_{ij} dW_{ij}$$

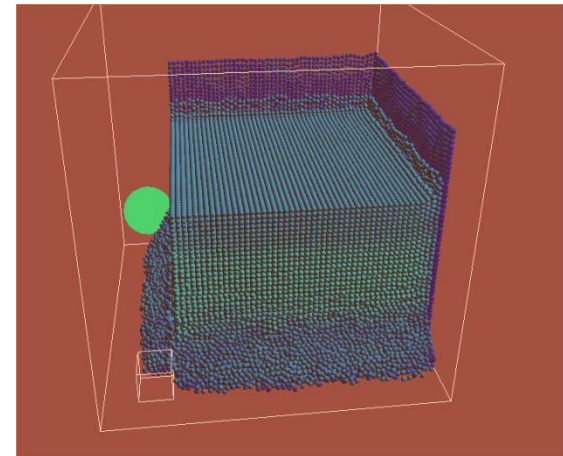
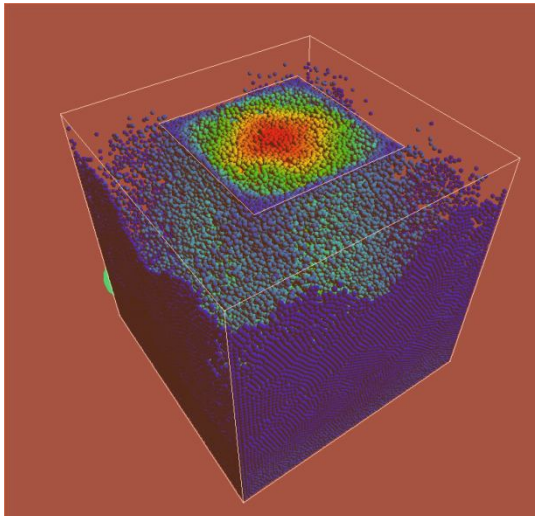
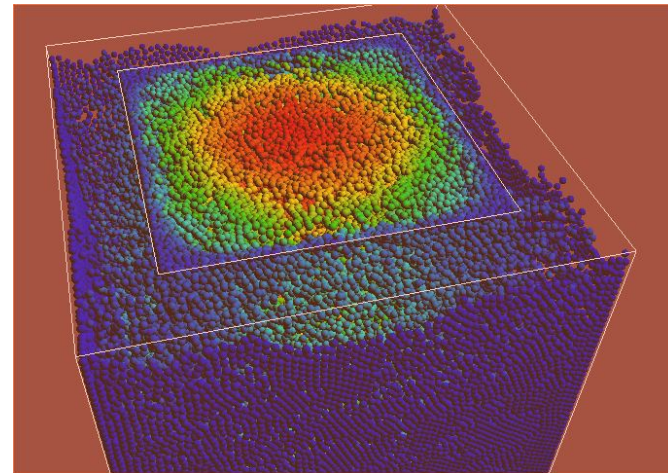
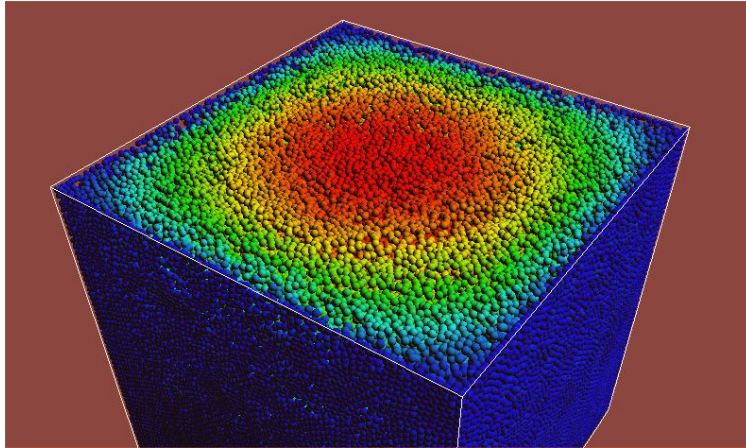
Atualização temporal por meio de Euler de primeira ordem



Para velocidade V_i e Posição X_i para tempo

$$t_{n+1} = t_n + \Delta t$$

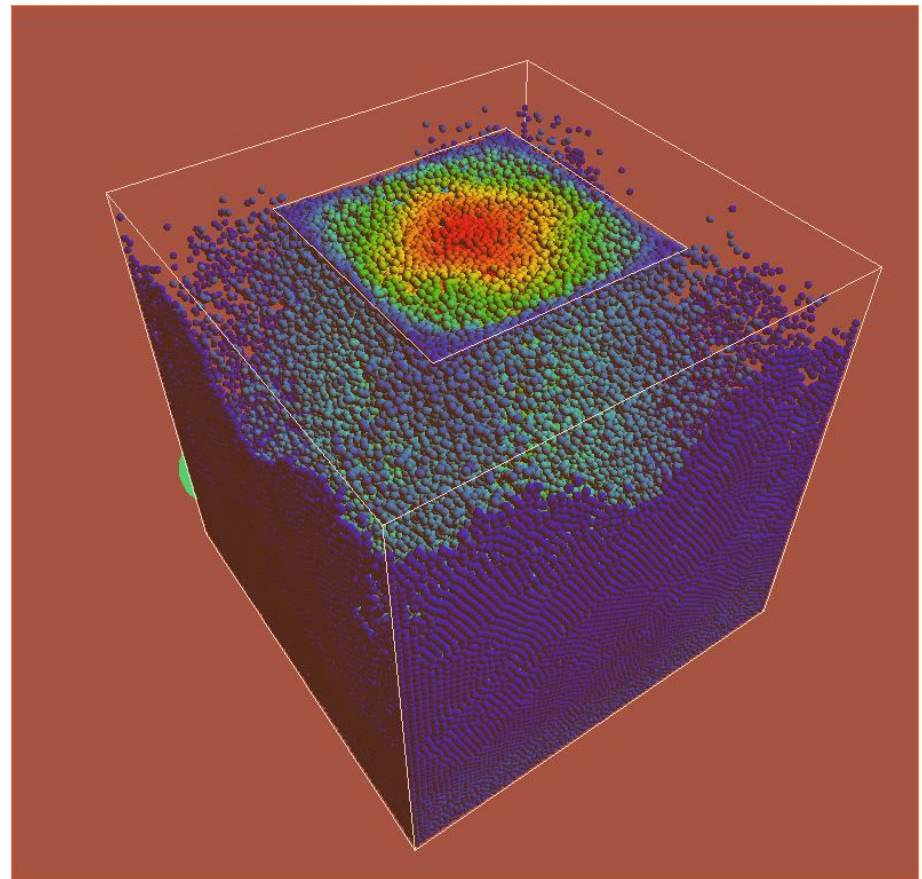
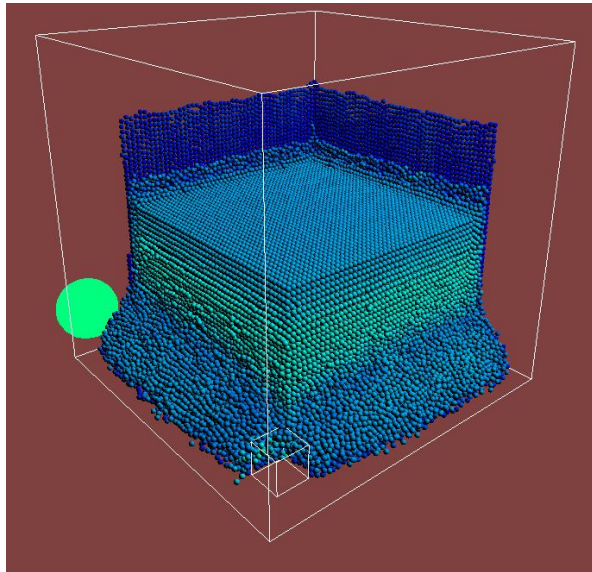
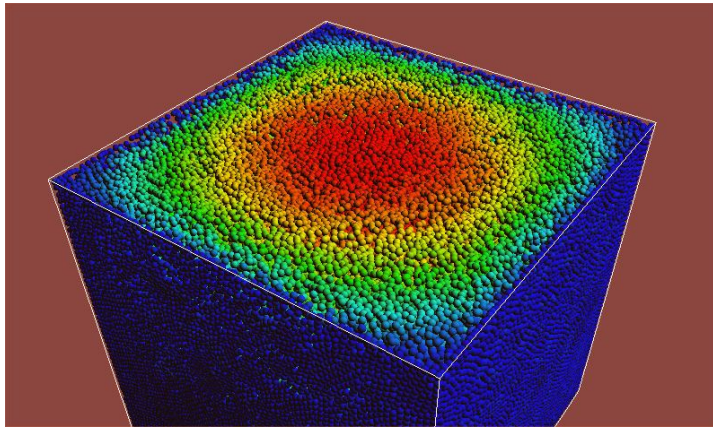
Visualizador



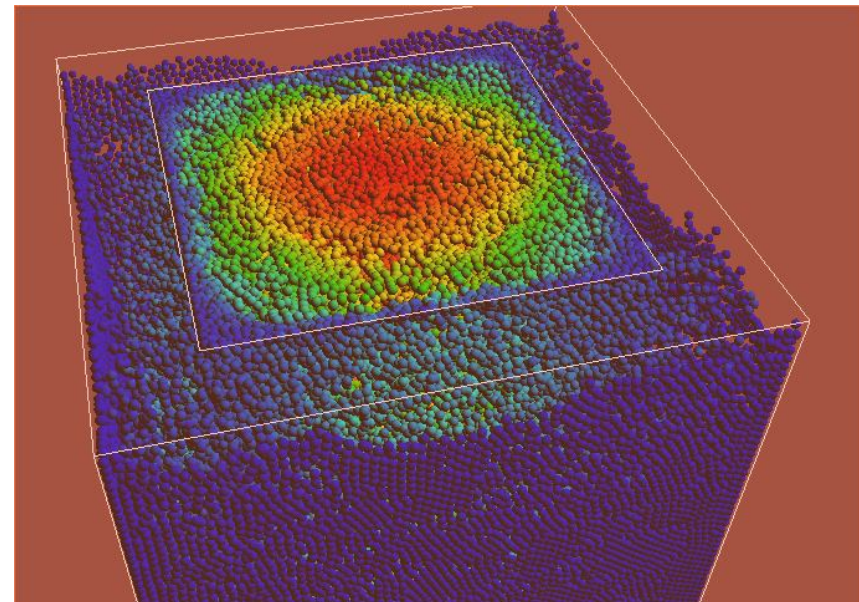
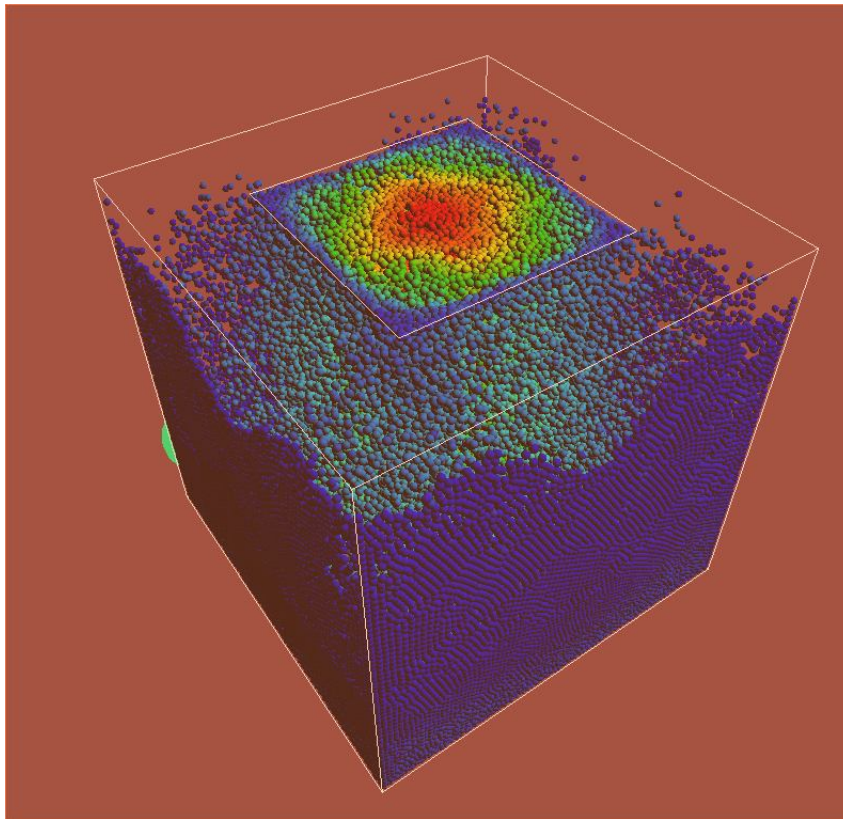
Visualizador

- *Função que transforma* os valores das grandezas das **velocidades das partículas em cores RGBA**.
- *Aplicação de Ferramentas interativas:*
 - Rotação
 - Zoom
 - Deslocamento longitudinal, vertical e transversal.
 - Menus auxiliares e comandos por teclas.

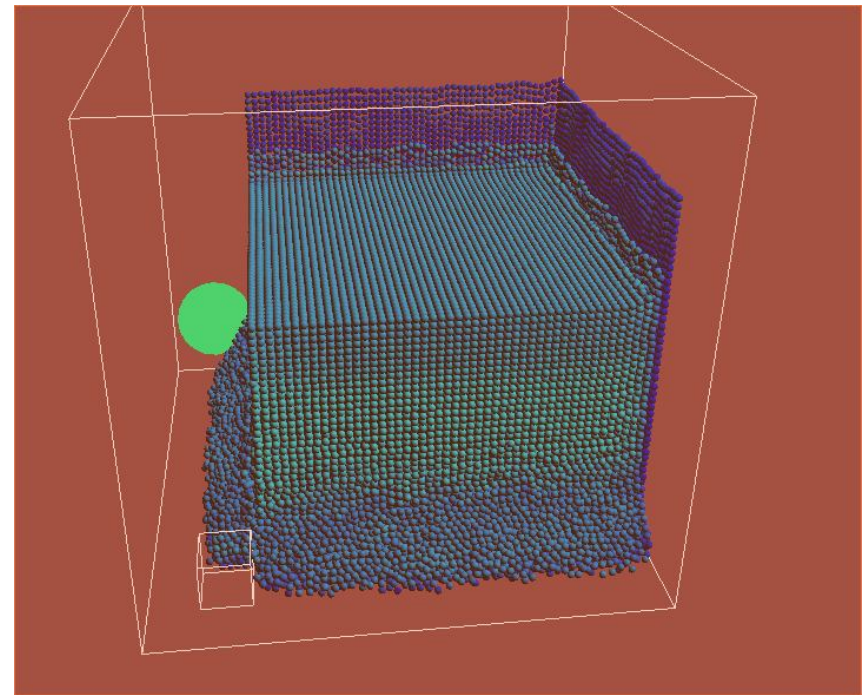
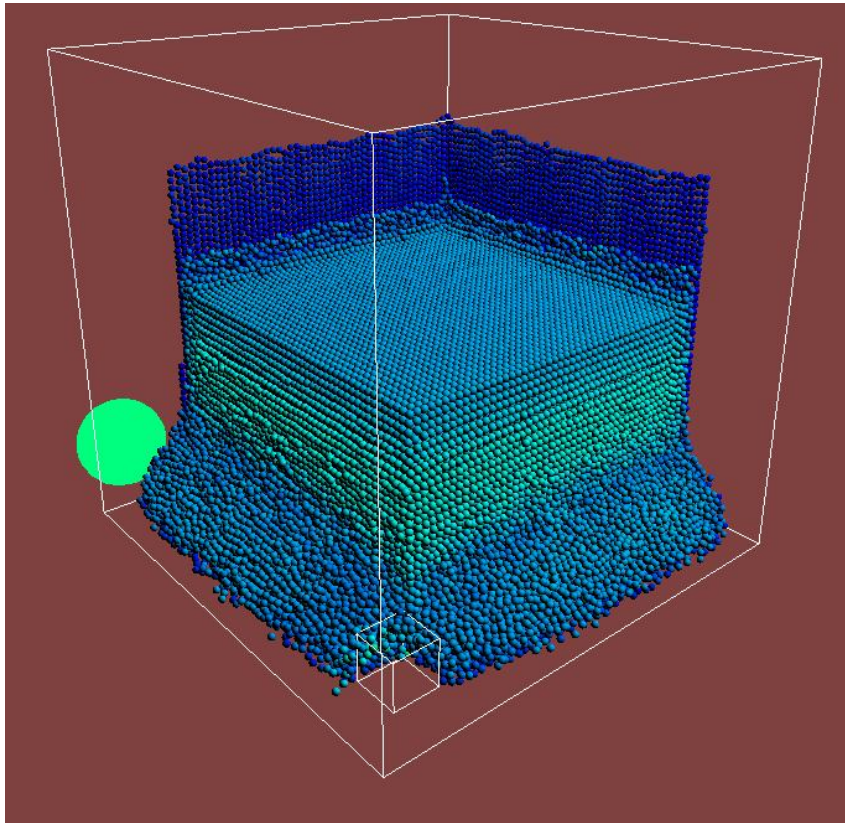
Velocidade



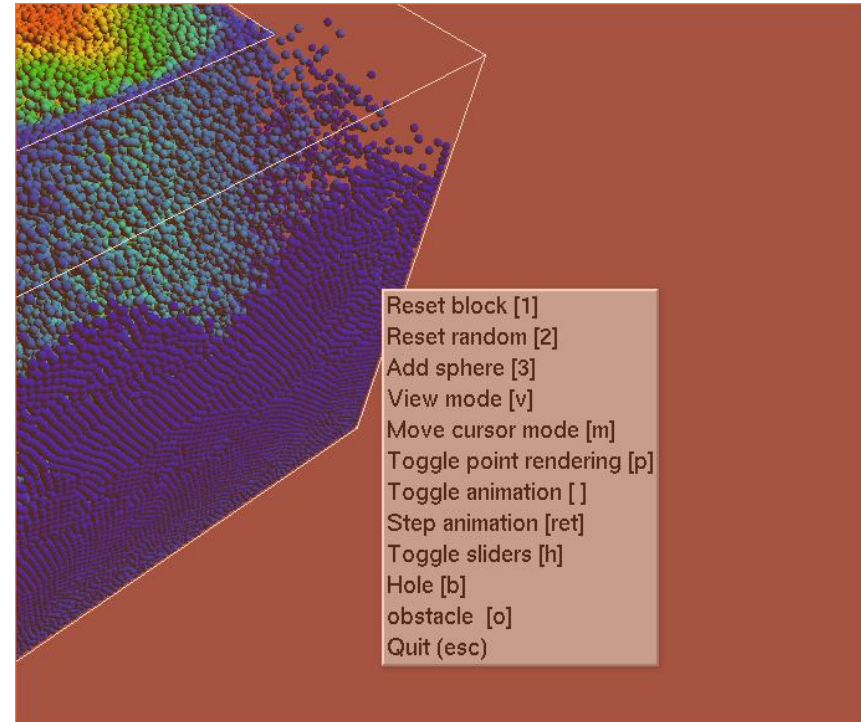
Condições de fronteira



Obstáculo



Menus



Conclusões

- *Se consegue que o visualizador interativo usando método de DPD, seja uma **ferramenta útil** para o **estudo e estabilização** do método, para vários exemplos.*
- **A visualização da grandeza das velocidades** nas partículas, foi uma aplicação bem sucedida, devido a que com ele se logra encontrar condições para o funcionamento ótimo do modelo. No caso especial da aplicação das **condições de contorno**.

Conclusões

- **As ferramentas iterativas, como menus, zoom, deslocamentos e rotação.** Ajudam para uma melhor observação dos fenômenos, em especial nas partículas que observação individual e detalhada se torno critica.
- A **execução de vários exemplos** em um mesmo visualizador e além de sua modificação em tempo real, acrescenta e facilita o estudo dos **fenômenos em partículas**, de uma maneira mais geral.

Conclusões

- **A interação em GPU de OpenGL e CUDA**, é uma interessante ferramenta tanto para a visualização como na simulação de fenômenos físicos.

Perspectivas

- **Gerar corte**, para visualização das partículas, em posições intermediárias .
- Conseguir visualizar **Linhas de fluxo**.
- **Instrumentos virtuais de medição**.
- Para o método de coleção nas partículas usar um método mais complexo como SPH ou MLS.
- **Implementação de problemas mas complexos, para casos multiescala**.



UNICAMP

Obrigado pela sua atenção

Perguntas?

Edgar Andres Patiño Nariño
(eapatinon@fem.unicamp.br)

REFERÊNCIAS

- [1] S. Li, W. K. Liu, and Shaofan Li ·Wing Kam Liu, *Meshfree particle method*, vol. 25, no. 2–3. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 289–296 vol.1.
- [2] G. R. Liu, “Mesh Free Methods: Moving Beyond the Finite Element Method,” *Applied Mechanics Reviews*, vol. 56, no. 2, p. B17, 2003.
- [3] J. Limido, C. Espinosa, M. Salau, and J. L. Lacome, “SPH method applied to high speed cutting modelling,” *International Journal of Mechanical Sciences*, vol. 49, pp. 898–908, 2007.
- [4] Ericson, C. (2004). Real-time collision detection. *Chemistry &*
Retrieved from
<http://onlinelibrary.wiley.com/doi/10.1002/cbdv.200490137/abstract>
- [5] A. A. Munjiza, *The Combined Finite-Discrete Element Method*. Wiley, 2004, p. 348.
- [6] M. Revenga, I. Zúñiga, and P. Español, “Boundary conditions in dissipative particle dynamics,” *Computer Physics Communications*, vol. 121–122, no. null, pp. 309–311, Sep. 1999.

REFERÊNCIAS

- [7] M. Griebel, S. Knapek, and G. Zumbusch, *Numerical simulation in molecular dynamics*. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 2007.
- [8] Ericson, C. (2004). Real-time collision detection. *Chemistry & ...*. Retrieved from <http://onlinelibrary.wiley.com/doi/10.1002/cbdv.200490137/abstract>.
- [9] Satish, N., Sciences, C., Harris, M., Garland, M., & Clara, S. (2009). Designing Efficient Sorting Algorithms for Manycore GPUs, (May), 1–10.
- [10] Nguyen, H. (2007). *Gpu gems 3*. Retrieved from <http://dl.acm.org/citation.cfm?id=1407436>
- [11] Hou, Q., Zhou, K., & Guo, B. (2008). BSGP: bulk-synchronous GPU programming. *ACM Transactions on Graphics (TOG)*. Retrieved from <http://dl.acm.org/citation.cfm?id=1360618>