

# Otimização do algoritmo de dinâmica molecular para esferas duras



Aluno: Gustavo Tavares Cabral<sup>1</sup>,  
Orientadora: Carolina Brito<sup>2</sup>

<sup>1</sup> Bacharelado em Ciência da Computação, UFRGS

<sup>2</sup> Instituto de Física, UFRGS

## Introdução

Muito utilizado para descrever interações entre partículas em teorias de fluidos e sólidos em mecânica estatística, o modelo de esferas duras descreve partículas esféricas que não podem se sobrepor no espaço. Apesar de ser muito simples, o sistema apresenta um diagrama de fases bastante rico, podendo ser utilizado para estudo de meios granulares, coloides, fluidos e polímeros [1].

No modelo de esferas duras, as esferas do sistema não podem se sobrepor. Assim, duas esferas não interagem se a distância entre elas é maior do que a soma dos seus raios; caso contrário, o potencial de repulsão entre elas é infinito. Esta característica possibilita o uso de um algoritmo orientado a eventos [2] (tipicamente, um evento é uma colisão entre duas partículas). Tal algoritmo permite diversas otimizações que tornam o modelo aplicável a uma ampla gama de sistemas. Neste trabalho, apresento duas destas otimizações: uso de células e uso de um *min-heap* binário [3].

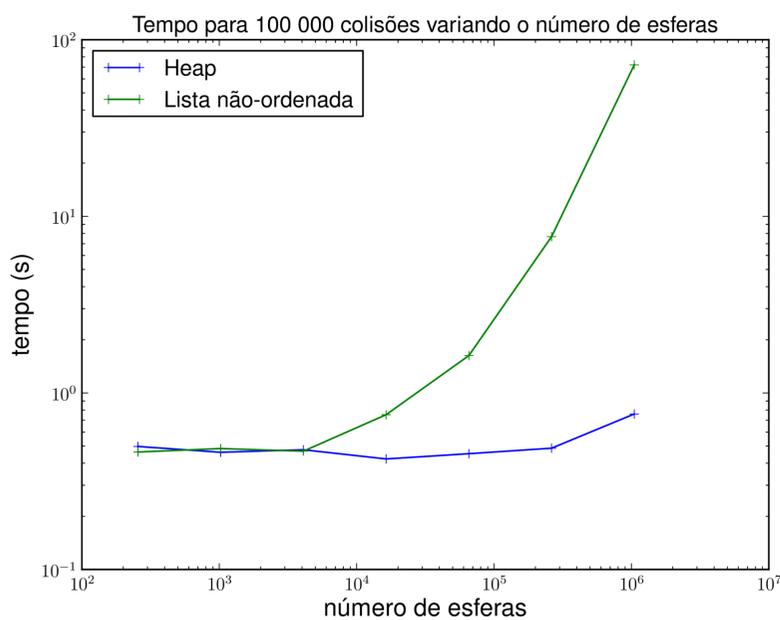


Figura 2: O uso do *heap* como fila de prioridade reduz o custo de  $O(n)$  para  $O(\log n)$  a cada iteração (sendo  $n$  o número de partículas), viabilizando simulações de grande escala.

## Resultados

Os aprimoramentos realizados no trabalho fazem com o tempo de processamento varie pouco em relação aos parâmetros do sistema, possibilitando simulações eficientes em sistemas com baixa densidade ou com milhões de partículas.

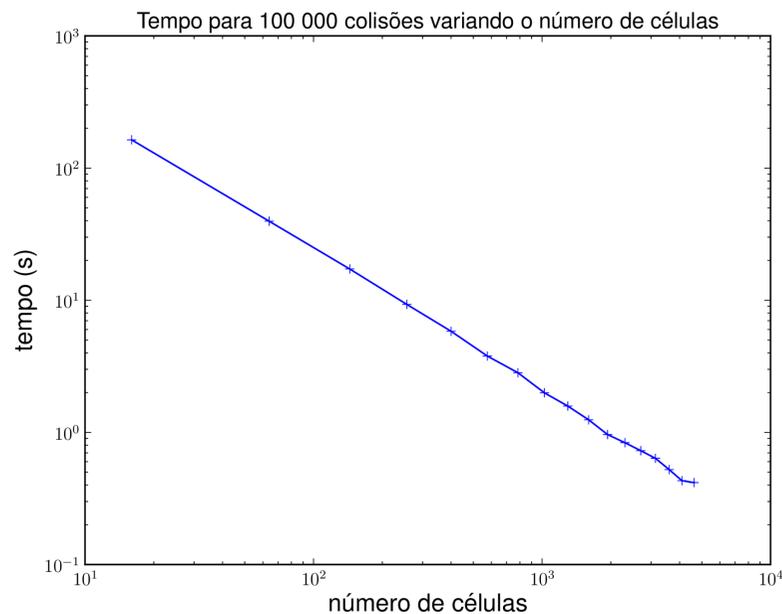


Figura 1: O gráfico mostra um aumento expressivo no desempenho com o aumento no número de células. O tamanho das células (e o número, por consequência) é limitado pelo tamanho das partículas.

## Otimizações realizadas

O ganho no uso de células se baseia no fato de que partículas distantes têm pouca probabilidade de realizar a próxima colisão. Com isso, o espaço pode ser dividido em células, e a verificação das colisões possíveis é feita apenas com as partículas das células vizinhas. A melhora de performance alcançada com o uso de células pode ser vista na figura 1.

Já o *min-heap* binário [4] é uma árvore binária em que um nó pai é sempre menor ou igual aos seus filhos. Em um heap de tamanho  $n$ , o custo para encontrar o menor valor é  $O(1)$ , enquanto uma atualização custa  $O(\log n)$ . Para a descoberta da próxima interação, o uso do *min-heap* é vantajoso em relação a uma lista não-ordenada porque esta tem custo  $O(n)$  para encontrar o menor valor e o número de atualizações na estrutura a cada iteração é aproximadamente constante. Uma comparação entre o uso de um *min-heap* e uma lista não-ordenada no algoritmo é mostrada na figura 2.

## Referências

- [1] G.Parisi, F.Zamponi - *Rev. Mod. Phys.* 82, 789 (2010).
- [2] M. P. Allen and D. J. Tildesley - *Computer Simulation of Liquids*. Oxford University Press, NY (1987).
- [3] A. Donev et al - *Jour. Comp Phys*, v 202, 737 (2005).
- [4] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest - *Introduction to algorithms*. MIT Press / McGraw-Hill (1990).