

Implementação de dinâmica molecular em uma rede de transputers

Gonzalo Travieso e Jan F. W. Slaets

Instituto de Física e Química de São Carlos – USP

CP 369 – CEP 13560 – São Carlos – SP

Tel: (0162) 72-4496

e-mail: gonzalo@uspfc.ifqsc.usp.br

Resumo

Apresentamos neste artigo uma implementação de simulação de dinâmica molecular de partículas de Lennard-Jones em um sistema de processamento paralelo constituído por uma rede de transputers. Mostramos que o tempo de processamento cresce de forma aproximadamente linear com o número de partículas, e decai também de forma aproximadamente linear com o número de processadores.

Abstract

In this work, a new method for molecular dynamics calculations of Lennard-Jones particles in a network of transputers is presented. We show that the scaling of simulation time is approximately linear with the number of particles, and the speed-up is approximately linear with the number of processors.

1 Introdução

Começando com as primeiras simulações de Fermi, Pasta e Ulam (ver [7]), que procuravam desvendar o mecanismo de surgimento da irreversibilidade nos sistemas macroscópicos (determinada pela segunda lei da termodinâmica) à partir de equações de movimento reversíveis (equações de Newton), e tomando um forte impulso com as famosas simulações de Rahman [9] utilizando potenciais contínuos no estudo de argônio líquido, a importância da dinâmica molecular no estudo e descrição teórica dos sistemas de muitos corpos vem crescendo constantemente.

Muitas das simulações podem ser levadas a cabo com uma quantidade relativamente pequena de partículas [7], e portanto com relativamente pequena demanda computacional, sem comprometer os resultados encontrados. No entanto, em alguns estudos é fundamental o uso de uma maior quantidade de partículas, de forma a aproximar mais adequadamente o caso real [1], e também pode ser necessária a execução por um tempo simulado maior, de forma a possibilitar o estabelecimento do equilíbrio do sistema [2].

Desta forma, o uso de um sistema de computação de alto desempenho se torna essencial, o que levou muitos pesquisadores à utilização de supercomputadores [6] ou de máquinas especificamente projetadas para o problema (por exemplo, [5] e [4]).

Neste artigo, procuraremos mostrar que uma solução também viável para o problema de escala envolvido nessas simulações pode ser encontrada através do uso de sistemas de processamento paralelo.

Primeiro apresentaremos abreviadamente um dos problemas típicos de dinâmica molecular, enfatizando apenas a parte mais importante do ponto de vista computacional. Em seguida, apresentaremos uma proposta de paralelização, e a análise de seu desempenho.

2 Dinâmica Molecular

Lidaremos aqui com o problema da simulação da dinâmica de um conjunto de N partículas esféricas idênticas que interagem através de um potencial de Lennard-Jones [2]:

$$V(r) = 4\epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right] \quad (1)$$

onde r é a distância entre os centros das partículas, ϵ e σ são constantes que dependem do tipo de partícula. A força de interação será então dada pelo gradiente negativo do potencial:

$$F(r) = -\nabla_r V \quad (2)$$

Notemos que a força de interação é simétrica, isto é, se a força da partícula j sobre a partícula i é F_{ij} , então a força exercida pela partícula i sobre a partícula j será $F_{ji} = -F_{ij}$. Isso divide pela metade o número de interações que devem ser calculadas.

As partículas estão distribuídas numa caixa, em geral quadrada (para as simulações bidimensionais) ou cúbica (para as simulações tridimensionais), de volume V , que é considerada como sendo repetida periodicamente em todas as direções (fig. 1). Assim, ao sair por um dos lados da caixa, a partícula entra pelo lado oposto, o que garante a conservação do número de partículas, sem a introdução das distorções que o confinamento das partículas em uma caixa fechada pequena poderia gerar¹

¹Se introduzimos paredes rígidas delimitando o sistema de partículas, isso representará uma grande

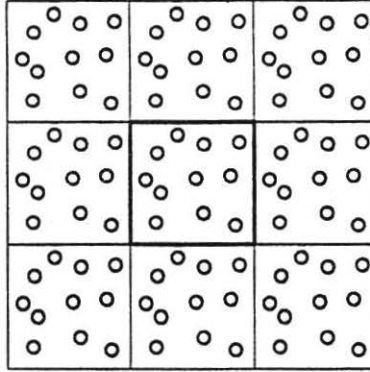


Figura 1: Condições periódicas de contorno

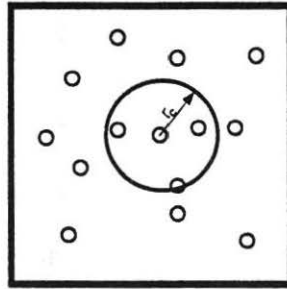


Figura 2: Partículas interagentes com uma partícula dada

Um ponto importante a ser notado com relação ao potencial utilizado é o de que ele decai rapidamente com a distância. Desta forma, para encontrar a força total atuando em uma dada partícula, basta calcular a interação desta partícula com aquelas que se encontrem dentro de um certo raio, chamado *raio de corte*, r_c dessa partícula. O valor utilizado com mais frequência é $r_c = 2.5\sigma$. (fig. 2).

Tendo sido calculada a força sobre cada partícula em um dado instante t , e conhecendo-se os valores da posição r_i e da velocidade v_i dessa partícula nesse instante, podemos então encontrar a posição no instante $t + \Delta t$ através da integração numérica das suas equações de movimento:

$$\dot{r}_i = p_i/m_i \quad (3)$$

$$\dot{p}_i = F_i \quad (4)$$

onde p_i é o momento associado à partícula, e m_i sua massa. Esse processo é então repetido para um certo número de intervalos de tempo Δt . Os métodos de integração mais utilizados

distorção em relação aos sistemas reais, pois qualquer sistema real confinado, mesmo que pequeno, apresenta um número muito maior de partículas do que é possível simular em um computador atual. A introdução da periodicidade mencionada elimina essa distorção, pois o sistema deixa de ser confinado. Com relação às distorções por sua vez introduzidas por essa periodicidade, são de caráter muito mais suave. O leitor pode encontrar estudos referentes a isso em [8].

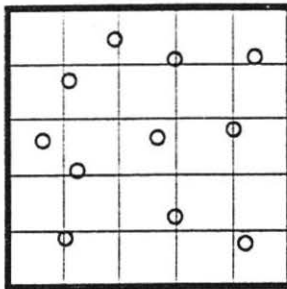


Figura 3: Divisão da caixa em células

são o *preditor-corretor de Gear* [2] e o de Verlet [11] em diversas variantes. No presente trabalho, utilizamos uma variante do método de Verlet proposta em [3].

Possuindo os valores de r_i e p_i podemos calcular características do sistema, tanto termodinâmicas (pressão, temperatura, etc.) como outras (função de distribuição de pares, coeficientes de transporte, etc).

3 Forma de paralelização

O modo utilizado para a paralelização do processamento tem como base o fato de que cada partícula somente interage com partículas próximas a ela. Isso permite uma divisão do processamento, associando a cada processador um conjunto de partículas “próximas”, de forma que as comunicações entre os processadores fiquem limitadas.

Com esse intuito, dividimos a caixa de simulação em diversos elementos retangulares chamados *células*, cada um tendo seus lados de tamanho maior ou igual ao raio de corte r_c (fig. 3). Podemos garantir que as partículas interagente com uma partícula de uma dada célula estão contidas na mesma célula ou em algumas das células vizinhas (8 vizinhas no caso bidimensional, 26 no caso tridimensional), e que portanto somente estas devem ser levadas em consideração durante o cálculo das forças. Isto permite reduzir o tempo de processamento de $O(N^2)$ para $O(N)$ (veja [10]).

Associamos então, a cada processador, uma *fatia* da caixa, dividida “verticalmente”, de acordo com a divisão das células, conforme a fig. 4. As partículas que estão em células de uma dada fatia têm os seus dados armazenados localmente no processador associado. Podemos ver então que cada processador somente necessita se comunicar com os seus vizinhos, o que nos dá uma estrutura em anel (fig. 5)².

Notemos agora então o seguinte: devido à simetria das forças acima referida, não necessitamos calcular as interações das partículas de uma dada célula com todos os seus vizinhos, mas apenas com metade. Desta forma, supondo que a varredura³ seja feita “verticalmente, de baixo para cima”, podemos considerar apenas as células vizinhas posteriores a uma dada

²Note-se aqui que estamos implicitamente assumindo que, ao calcular a nova posição de uma partícula, ela não será diferente da posição anterior em mais do que o tamanho de qualquer dos lados de uma célula. De fato, os deslocamentos apresentados por cada partícula devem ser muito menores ainda, pois se não fosse assim, haveria demasiada instabilidade na integração numérica, o que levaria à necessidade de redução do Δt empregado.

³A *varredura* designa aqui o método pelo qual consideramos cada célula por vez de forma a calcular todas as interações de todas as partículas associadas a um dado processador

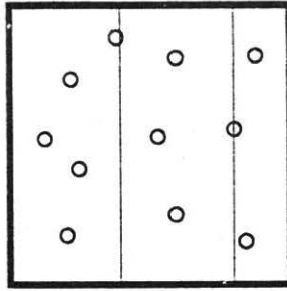


Figura 4: Divisão da caixa de simulação em faixas

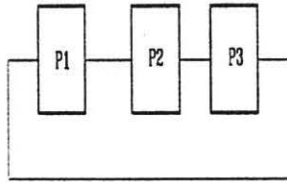


Figura 5: Anel de processadores

célula (fig. 6).

Utilizando-nos deste fato, podemos reduzir ainda mais a necessidade de comunicação entre processadores durante o cálculo através da repetição da primeira faixa “vertical” de células de cada processador como última faixa do processador anterior (no anel). Assim, durante todo o processamento de cálculo de interações, cada processador é auto-suficiente, necessitando interagir com os vizinhos apenas em duas ocasiões: na comunicação das forças calculadas por ele sobre as partículas repetidas (cujos dados estão associados ao próximo processador), e depois de calcular as próximas posições das partículas, quando elas devem ser redistribuídas pelas células.

Este método de paralelização foi implementado numa rede de *transputers*, ligada a um computador pessoal tipo IBM-XT.

4 Análise da proposta

Analisaremos agora o desempenho da forma proposta de paralelização, levando em consideração o tempo de processamento, primeiro através do desenvolvimento de algumas fórmulas aproximativas, e em seguida pela apresentação de resultados de levantamentos efetuados. Levantaremos primeiro a dependência com relação ao número de partículas, e em seguida com o número de processadores.

4.1 Dependência com o número de partículas

Utilizaremos à partir de agora, para simplicidade, as unidades reduzidas de Lennard-Jones, onde a massa da partícula é unitária, e as distâncias são medidas em unidades de σ . Faremos um levantamento considerando apenas o tempo necessário para o cálculo das interações, que é o fator dominante. Os levantamentos feitos aqui são para o caso bidimensional, mas podem

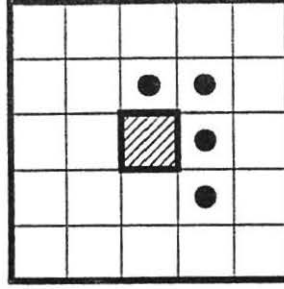


Figura 6: Celulas vizinhas a considerar

facilmente ser extendidos para o caso tridimensional.

Sejam dados o número de partículas N e a densidade ρ , e suponhamos que a caixa de simulação é quadrada, isto é: $V = L^2$, onde L é o lado da caixa. Devemos ter então: $\rho = \frac{N}{L^2}$, e portanto:

$$L = \sqrt{\frac{N}{\rho}} \quad (5)$$

Cada célula será quadrada, de lado $\lambda \geq r_c$. O número total de células será então $M = m^2$, onde:

$$m = \lceil \frac{L}{r_c} \rceil \quad (6)$$

$\lceil x \rceil$ denota o maior inteiro menor que x .

Considerando as partículas distribuídas, em média, igualmente em cada célula, o número de partículas por célula será $n = N/M$. O número médio de interações entre as partículas de cada célula será então:

$$\frac{n(n-1)}{2} \approx \frac{n^2}{2} \quad (7)$$

e da célula com cada uma de suas vizinhas será n^2 .

Como devemos usar apenas a metade de seus vizinhos ao verificar os pares que podem interagir, os valores aproximados de tempo de cálculo de distância de pares serão:

$$T_p \approx k_1 4.5 M n^2 \quad (8)$$

onde k_1 é uma constante.

As interações que precisam efetivamente ser calculadas são metade das partículas que se encontram dentro de um círculo de raio r_c em relação a uma partícula dada. Utilizamos apenas a metade pois cada par precisa ser considerado apenas uma vez. Portanto, dentro da suposição de partículas distribuídas uniformemente:

$$T_i \approx k_2 \frac{\pi r_c^2 \rho}{2} N \quad (9)$$

onde k_2 é uma constante.

O tempo total para um passo de simulação (de t a $t + \Delta t$) será então:

$$T \approx k_1 4.5 M n^2 + k_2 \frac{\pi r_c^2 \rho}{2} N \quad (10)$$

Tabela 1: Valores de tempo de cálculos

N	M	n	T
98	16	6.13	0.06
512	81	6.32	0.31
1058	196	5.40	0.60
2048	361	5.67	1.19
5000	900	5.56	2.88
10082	1849	5.45	5.75

Convém salientar que n é afetado principalmente pela densidade ρ , e somente de forma secundária pelo número de partículas, e mesmo assim, sem um crescimento contínuo. Já M é afetado principalmente pelo número de partículas. Podemos concluir, então, que o tempo de processamento cresce linearmente com o número de partículas. O único fator que determina a quebra da linearidade é a necessidade de arredondar L/r_c para um número inteiro no cálculo de m .

Na tabela 1, apresentamos os valores calculados de M e n para diversos valores de N , com $\rho = 0.8442$, juntamente com o tempo de execução (em segundos) levantado para um passo de simulação em um *transputer*.

Um cálculo pelo método dos mínimos quadrados nos dá $k_1 = 1.06 \cdot 10^{-5}$ e $k_2 = 3.72 \cdot 10^{-5}$.

4.2 Dependência com o número de processadores

Analisemos agora a dependência do tempo de processamento com o número de processadores utilizados. A primeira coisa a notar é que a análise anterior se aplica a cada um dos processadores se, ao invés dos M e N totais, utilizarmos o número de células e de partículas *dcsse dado processador*. Observe-se também que, como os cálculos de força são feitos independentemente pelos processadores, o tempo total será determinado pelo tempo do processador que tiver a maior carga.

Como já vimos, as M células são divididas em fatias “verticais” para cada processador. Desta forma, cada processador ficará com uma parte das m colunas de células. No entanto, a divisão nem sempre poderá ser igual, mas apenas quando m for múltiplo de p (número de processadores)⁴. Assim, o número de colunas máximo em um processador será:

$$\tilde{m} = \lfloor m/p \rfloor \quad (11)$$

onde $\lfloor x \rfloor$ indica o menor inteiro maior que x . Teremos, para o maior número de células em um processador, $\tilde{M} = m\tilde{m}$. O número de partículas nesse mesmo processador será $\tilde{N} \approx \tilde{M}n$. Podemos então utilizar a expressão (10) com \tilde{M} no lugar de M e \tilde{N} no lugar de N .

Portanto, o tempo de processamento decai de forma aproximadamente linear com o número de processadores, sendo que a quebra da linearidade só ocorre devido à necessidade de arredondamento de m/p para um valor inteiro no cálculo de \tilde{m} . No entanto, este fator não apresenta nenhuma tendência dominante de crescimento, seja com N seja com p .

Na tabela 2 apresentamos o cálculo de tempo de processamento no caso $N = 10082$, $\rho = 0.8442$, para 1, 2 e 3 processadores, utilizando os valores de k_1 e k_2 encontrados anteriormente. Apresentamos também, na última coluna, o tempo efetivamente levantado. Vemos

⁴Uma outra forma seria alterar o tamanho das células, de forma que tivéssemos m múltiplo de p . Esse método não será estudado aqui.

Tabela 2: Tempos de processamento em função do número de processadores

p	n	M	$T(\text{calculado})$	$T(\text{real})$
1	43	1849	5.73	5.75
2	22	946	2.93	2.84
3	15	645	2.00	1.94

que os valores calculados são bastante próximos dos valores encontrados nas simulações.

5 Conclusão

Apresentamos uma forma de implementação em paralelo de simulações de dinâmica molecular. As análises apresentadas mostram principalmente dois fatos:

- O crescimento do tempo de processamento é aproximadamente linear com o número de partículas envolvido, a menos de algumas descontinuidades determinadas por aproximação para inteiro no cálculo do número total de células;
- A queda do tempo de processamento com o número de processadores é também aproximadamente linear, com algumas descontinuidades devidas a aproximação para inteiro no cálculo do número de colunas de células por processador.

Com relação às descontinuidades apresentadas, devemos notar que elas se tornam cada vez menos importantes à medida que cresce o número total de partículas. Desta forma, o método apresentado é adequado para a simulação de sistemas grandes por dinâmica molecular.

Referências

- [1] Abraham, F. F. Computation statistical mechanics — methodology, properties and supercomputing. *Advances in Physics*, v. 35, n. 1, p. 1-111, 1986.
- [2] Allen, M. P. and Tildesley, D. J. *Computer Simulation of Liquids*. Clarendon Press-Oxford, 1987.
- [3] Andersen, H. C. Rattle: a “velocity” version of the shake algorithm for molecular dynamics calculations. *Journal of Computational Physics*, n. 52, p. 24-34, 1983.
- [4] Bakker, A. F.; Bruin C.; van Dieren, F. and Hilhorst, H. J. Molecular dynamics of 16000 Lennard-Jones particles. *Physics Letters*, v. 93A, n. 2, 1982.
- [5] Bakker, A. F.; Gilmer, G.H.; Grabow, M. H. and Thompson, K. A special purpose computer for molecular dynamics calculations. *Journal of Computational Physics*, n. 90, p. 313-335, 1990.
- [6] Grest, G. S.; Dünweg, B. and Kremer, K. Vectorized link cell FORTRAN code for molecular dynamics simulations for a large number of particles. *Computer Physics Communications*, n. 55, p. 269-285, 1989.

- [7] Hoover, W. G. Molecular dynamics. In *Lecture Notes in Physics*. Springer-Verlag, 1986.
- [8] Pratt, L. R. and Haan, S. W. Effects of periodic boundary conditions on equilibrium properties of computer simulated fluids. I. Theory. *Journal of Chemical Physics*, v. 74, n. 3, 1981.
- [9] Rahman, A. Correlations in the motion of atoms in liquid argon. *Physical Review*, v. 136, n. 2A, p. A405-A411, oct. 1964.
- [10] Travieso, G. Estudo de processamento paralelo para dinâmica molecular. Dissertação (Mestrado em Física Aplicada), Instituto de Física e Química de São Carlos, Universidade de São Paulo, mar. 1989.
- [11] Verlet, L. Computer experiments on classical fluids: I. thermodynamical properties of Lennard-Jone molecules. *Physical Review*, v. 159, n. 1, p. 98-103, july 1967.