

O MÉTODO DOS ELEMENTOS DISCRETOS EM PROBLEMAS DE FLUXO GRANULAR USANDO PROCESSAMENTO PARALELO

The Discrete Element Method in Granular Flux Problems using Parallel Processing

Luiz Carlos Facundo Sanches (1); Ronaldo Carrion (2); Carlos Heracles M. Lima (3)

(1) Dr. Prof., Universidade Estadual Paulista, São João da Boa Vista - SP, Brasil.

(2) Dr. Prof., Universidade de São Paulo, São Paulo - SP, Brasil.

(3) Dr. Prof., Universidade Estadual Paulista, São João da Boa Vista - SP, Brasil

E-mail para Correspondência: luiz.cf.sanches@unesp.br; (P) Apresentador

Resumo: O Método dos Elementos Discretos (MED) tem sido usado para simular fluxos de partículas, embora ultimamente esteja se tornando interessante para representar meios descontínuos. Em geral, o MED é computacionalmente caro. Entretanto, a computação reconfigurável pode ser usada para torná-lo menos exigente. As Unidades Centrais de Processamento (CPU) com vários núcleos atraíram muita atenção para acelerar as simulações de computador em várias aplicações. Na verdade, existem arquiteturas para computação paralela que aceleram o processamento, enquanto fazem uso efetivo da memória disponível. Resultados da literatura mostram que a programação paralela funciona de forma semelhante em plataformas de CPUs e é bem conhecido que algoritmos de elementos discretos têm maior escalabilidade. Além disso, o uso de memória é bastante reduzido usando este esquema paralelo. Este artigo apresenta um procedimento computacional de elementos discretos em problemas de deposição e fluxo granular e discute a viabilidade de usar computação paralela em computadores pessoais com processadores de vários núcleos.

Palavras chave: *Método dos Elementos Discretos; fluxo granular; computação paralela; processadores com vários núcleos.*

Abstract: The Discrete Element Method (MED) has been used to simulate particle flows, although lately it has become interesting in discontinuous medium. In general the MED is computationally expensive. However, reconfigurable computing can be used to make it less demanding. Multi-core central processing units (CPUs) have attracted a lot of attention to speed up computer simulations in various applications. In fact, there are parallel computing architectures that speed up processing while making effective use of available memory. The results in the literature show that parallel programming works similarly on CPU platforms and it is well known that discrete element algorithms have greater scalability. Additionally, memory usage is reduced using this parallel scheme. This paper presents a computational procedure of discrete elements in collapse of particles, granular flow problems and discusses the feasibility of using parallel computing on personal computers with multi-core processors.

Keywords: *discrete element method, granular flow, parallel computing, multi-core processors*



1 INTRODUÇÃO

O Método de Elementos Discretos (MED) foi originalmente descrito por Cundall e Strack (1979). Ele tem sido amplamente utilizado em análises numéricas envolvendo dinâmica de meios fluidos e sólidos. O esquema essencial é uma abordagem Lagrangeana onde as partículas individuais são calculadas com base na segunda lei de *Newton* do movimento. O método permite investigar por exemplo, a deposição, o fluxo granular de partículas e seu movimento precisamente.

Entretanto, quando o número de partículas aumenta, os algoritmos tradicionais tornam-se muito dispendiosos. Novas modelagens e técnicas de processamento paralelo são necessárias para implementar esta metodologia quando é necessário utilizar processos em grande escala. Nos últimos anos, poderosos *clusters* de computadores e super computadores foram usados para simular sistemas complexos com grande número de partículas. No entanto, seus custos excessivos de implantação limitaram sua popularidade. Problemas semelhantes foram relatados para outras aplicações como, dinâmica de fluidos computacional, análises estruturais e simulações de corpo rígido.

Recentemente, observa-se que o número de núcleos disponíveis em uma *Central Processing Unit* (CPU) de um computador pessoal tem aumentando constantemente. Shigueto e Sakai (2011) destacam que o processamento paralelo em vários núcleos de uma CPU pode ser uma promissora alternativa para permitir que a paralelização se torne mais acessível. Neste sentido, destacamos que algumas estratégias gerais para paralelização em CPUs foram descritas por Munjiza, et al. (2012). Por outro lado, segundo Wang, et al. (2013), outra alternativa é o processamento em *Graphics Processing Unit* (GPU). Adicionalmente, Lukas et al. (2014) apresentou uma abordagem bidimensional para a paralelização do método híbrido de elementos finitos e elementos discretos visando o uso de computadores de mesa.

Resultados semelhantes foram relatados para outras simulações como, por exemplo, dinâmica de fluidos computacional, análises estruturais e simulações de corpos rígidos. Foi observado que o algoritmo DEM é altamente escalável em computação paralela com vários núcleos em uma CPU.

O presente trabalho, tem como proposta descrever o método dos elementos discretos e examinar o desempenho de um esquema paralelo em CPUs com vários processadores considerando dois problemas clássicos. O primeiro modelo, avalia a deposição de material granular em uma base rígida fixa e o segundo avalia o fluxo geral de partículas passando por um orifício.

2 O MÉTODO DOS ELEMENTOS DISCRETOS

Diversos materiais com forma granular podem ser representados como um conjunto de partículas, normalmente esféricas, cada uma das quais interagem entre si e também com paredes planas através de um esquema simplificado de interação força-deslocamento. A solução numérica envolve a computação da força que atua sobre cada partícula em um dado instante. O montante e uma taxa de sobreposição entre elementos vizinhos são utilizados para determinar as forças de contato em cada instante de tempo. São atualizadas as velocidades e posições das partículas através de um esquema explícito de integração de diferenças finitas de primeira ordem no tempo. As forças e momentos agindo sobre cada partícula são calculados com base no equilíbrio local de vizinhos, e são usados para estimar a aceleração instantânea de cada partícula. As forças de contato são atualizadas e as novas forças e momentos resultantes no centro de gravidade de cada elemento são computados. O ciclo é repetido com pequenos incrementos até a solução final ser obtida. Na formulação, o tempo pode representar o tempo real ou um parâmetro fictício para representar o incremento de carga de um estágio de carregamento para o seguinte (Cundall e Strack, 1979).

2.1 Equações de equilíbrio

As equações que regem o equilíbrio dinâmico de translação e rotação de uma partícula i com massa m_i são:

$$\frac{d^2 x_i(t)}{dt^2} = \ddot{x}_i(t) = \frac{F_{x_i}(t)}{m_i}, \quad (1)$$

$$\frac{d^2 \theta(t)}{dt^2} = \ddot{\theta}_i(t) = \dot{\omega}_i(t) = \frac{T_i(t) + M_i(t)}{I_i}, \quad (2)$$

sendo $x_i(t)$ as posições dos centróides de uma partícula esférica i dispostas em um sistema de coordenadas cartesiano fixo, \ddot{x}_i é o vetor de aceleração para a i -ésima partícula, F_{x_i} são as forças de contato, m_i é a massa da partícula, θ é a rotação, $\dot{\omega}_i$ é o vetor da aceleração angular, T_i é o torque da partícula i , M_i é o momento resultante que atua através do centróide da partícula i e I_i é o seu momento de inércia. Para uma partícula circular o momento de inércia é igual a $\rho \pi r_i^4 / 2$ onde r_i é o raio e ρ é a densidade.

2.2 Posição das partículas

Considerando o equilíbrio dinâmico e conhecendo as forças resultantes agindo sobre as partículas, é possível calcular as acelerações para a i -ésima partícula. Particularmente, para movimento de translação, temos:

$$m_i a_i^t = F_i^t, \quad (3)$$

sendo m_i a matriz de inércia (massa), a_i^t o vetor de aceleração no tempo t e F_i^t o vetor de força resultante.

Para atualizar tais parâmetros é necessário implementar métodos de integração (Cundall e Strack, 1979). A relação entre o vetor de aceleração e velocidade é:

$$a_i^t = \frac{1}{\Delta t} (v_i^{t+\Delta t/2} - v_i^{t-\Delta t/2}), \quad (4)$$

sendo $v_i^{t-\Delta t/2}$ e $v_i^{t+\Delta t/2}$ as velocidades em $t - \Delta t/2$ e $t + \Delta t/2$, respectivamente, para a i -ésima partícula.

A Equação 4 é também conhecida como esquema de integração no tempo de *Verlet*. A velocidade no tempo $t + \Delta t/2$ é então calculada como:

$$v_i^{t+\Delta t/2} = v_i^{t-\Delta t/2} + \Delta t \frac{1}{m_i} (F_i^t), \quad (5)$$

A velocidade no tempo $t + \Delta t/2$ é igual à velocidade média calculada dentro do intervalo de t para $t + \Delta t$. Então, podemos calcular a posição da partícula atualizada como:

$$x_i^{t+\Delta t} = x_i^t + \Delta t \times v_i^{t+\Delta t/2}, \quad (6)$$

onde o vetor x de posição de partícula fornece as coordenadas cartesianas de partículas e, quando necessário, a rotação total em torno do eixo principal. Existem três tipos básicos de interações: interações interpartículas (colisão), forças de corpo (gravidade e viscosidade aparente) e interações partícula-parede. Foi especificado que a aceleração gravitacional é 9,81 m/s na direção negativa do eixo x_2 . Este é o valor usual para simulações em que o eixo x_2 é assumido como vertical. A viscosidade aparente ($< 0,05$ para problemas de elastodinâmica e $\sim 0,5$ a $0,6$ para problemas quase estáticos podem ser apropriados) é incluída como uma força de amortecimento proporcional à velocidade instantânea de cada partícula atuando na direção oposta ao movimento.

2.3 Forças de contato entre partículas

O movimento de translação e de rotação (quando necessário) das partículas foram incrementados utilizando um procedimento de integração numérica no qual as configurações de montagem foram geradas em intervalos de tempo Δt . As forças de contato consistem em componentes normais e tangenciais, cada uma das quais tinha um termo de dissipação viscoso proporcional à componente da velocidade nessa direção (O'Sullivan, 2011).

A força F_n denota a componente normal de contato, enquanto a componente tangencial é denotada por F_t . A força normal de contato pode ser calculada como sendo:

$$F_n = F_N - c_n v_n, \quad (7)$$

sendo F_N a parte elástica da interação normal, c_n o coeficiente de amortecimento da interação normal, $v_n = (v \cdot n)n$ com n representando o vetor unitário da interação entre os centros das duas partículas e v o vetor da velocidade de translação da partícula. O segundo termo do lado direito da Equação (7) representa o amortecimento de contato. Há também uma interação tangencial, a componente de atrito que é dada por:

$$F_t = \mu F_N \left(1 - \left(1 - \frac{|\delta_t|}{\delta_{máx}} \right)^{3/2} \right) \quad (8)$$

sendo δ_t o deslocamento tangencial total entre as duas superfícies a partir do ponto em que eles entraram em contato inicialmente. Se $|\delta_t| > \delta_{máx}$, então o deslizamento bruto é considerado como tendo iniciado e a força de atrito assume um valor constante dado pela lei de Amonton, $F_t = \mu F_N$, onde μ é o coeficiente de atrito.

Neste caso, a escolha de $\delta_{máx}$ (deslocamento tangencial máximo) é tipicamente várias ordens de grandeza maior do que para um contato típico. Além disso, a força de interação tangencial total em qualquer tempo é:

$$F_t = F_T - c_t v_t, \quad (9)$$

O segundo termo do lado direito da Equação (9) é a força de amortecimento de contato proporcional à componente tangencial da velocidade relativa, $v_t = v \times n$ e c_t é o coeficiente de amortecimento da interação tangencial. No modelo adotado, utilizou-se partículas não-rotacionais para avaliar a deposição e o fluxo em orifício.

3 PROCESSAMENTO PARALELO

A paralelização foi feita por meio de um código aberto orientado a objetos escrito em C++. Foi empregado o algoritmo de pesquisa de vizinhança de *Verlet* para detecção de partículas vizinhas e um esquema finito explícito de integração no tempo de primeira ordem. Uma *Application Programming Interface* (API) permitiu avaliar as simulações através de *scripts* escritos em linguagem *Python*. Cada conjunto de elementos discretos funcionou de forma independente. A paralelização foi conseguida pela decomposição do domínio do problema em subdomínios, onde a interface de cada subdomínio precisa ser duplicada para que as informações sejam intercambiadas entre dois subdomínios vizinhos (Weatherley, et al., 2014). As simulações foram feitas em *Linux* em um computador pessoal usando os núcleos de sua CPU. O processamento foi implementado em um *Intel Core i-7*, com 2,40 GHz de CPU clock e 8,0 GB de memória RAM.

3.1 Fluxograma MED para processamento paralelo

A Figura 1 mostra o fluxograma do processo para simulação usando o MED.

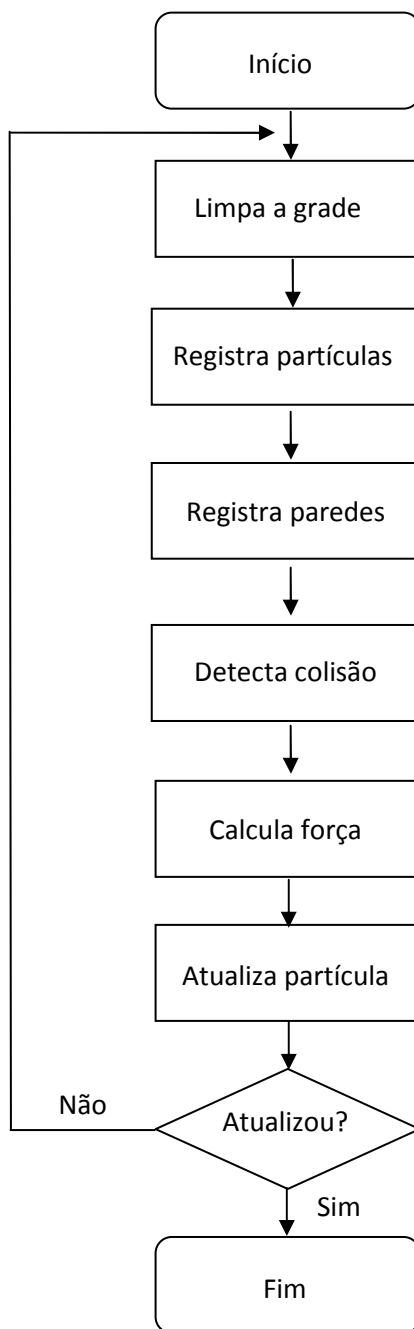


Figura 1. Fluxograma simplificado da simulação

Fonte: (Shigueto e Sakai, 2011)



Os processos incluem limpar o domínio, registrar partículas, registrar paredes, detecção de colisão, cálculo da força de contato e atualização de partículas. Desta forma, são três as etapas computacionais: avaliação da força interna, na qual as forças de contato são calculadas; integração de equações de movimento, em que os deslocamentos de elementos são computados; e detecção de contato, onde novos contatos são identificados e contatos interrompidos são removidos. A interação dos elementos é tratada como um processo dinâmico que alterna entre a aplicação da segunda lei de *Newton* e a avaliação de uma lei de força-deslocamento nos contatos. Esta lei dá a aceleração de um elemento resultante das forças que atuam sobre ele, incluindo forças gravitacionais, forças externas prescritas pelas condições de contorno e forças internas desenvolvidas nos contatos de inter-elementos. A aceleração é então integrada para obter a velocidade e o deslocamento. A lei de deslocamento de força é usada para encontrar forças de contato de deslocamentos conhecidos. As equações de movimento são então integradas no tempo. Detalhes deste processo são dados Cundall e Strack (1979). O método é computacionalmente exigente e, portanto, algoritmos paralelos eficientes, especialmente para as avaliações internas de força e detecção de contato, são necessários. Por este motivo, uma estrutura subdomínios e cálculo de força são propostas para o cálculo paralelo. Uma vez registradas as partículas calculadas, elas são armazenado na memória usando uma estrutura vinculada. Esta efetividade computacional é particularmente importante para grandes discretizações. A eficiência do estimativa de força de contato por meio de um método de cálculo da força é aumentado com a computação paralela.

4 RESULTADOS NUMÉRICOS

4.1 Colapso de um conjunto de partículas

Neste primeiro modelo, estudou-se o colapso de um conjunto de partículas não ligadas mas, com atrito ente elas e sob a ação da gravidade, incorporando tanto repulsão elástica como resistência ao atrito entre as partículas de contato. Para isto, considerou-se um cubo tridimensional quadrado, contendo partículas esféricas cujo raio variou de 0,2 a 0,5 mm, apoiado na sua parte inferior por uma parede rígida.

Foi implementada repulsão elástica e atrito entre as partículas e entre as partículas e a parede. Nas partículas, o coeficiente de rigidez normal foi adotado o valor de 1.000,0 N/m, o tangencial 100,0 N/m e o coeficiente de atrito 0,63. Além disso, foram definidos para o coeficiente de aceleração da gravidade o valor de $9,81 \text{ m/s}^2$ e para a viscosidade 0,6. Na parede, o coeficiente de rigidez normal é 10.000,0 N/m.

Executou-se a simulação para 10^5 interações com um incremento de passo de tempo de 10^{-4} segundos. Também foi usado o programa de visualização *POVray* (Weatherley, et al., 2014) para gravar instantâneos (*snapshots*) a cada 10^3 passos de tempo. A Figura 2 mostra o domínio do modelo gerado para a simulação.

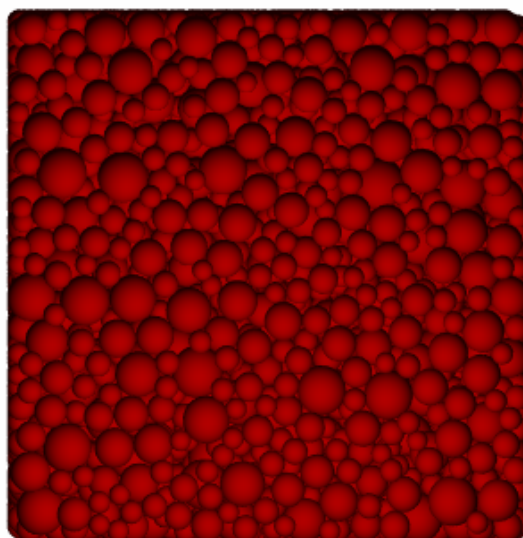


Figura 2. Domínio quadrado composto por partículas de diferentes tamanhos

Alguns *snapshots* com resultados da deposição das partículas são apresentados na Figura 3. Nesta figura, podemos observar que quando duas partículas se tocam, uma rigidez ao cisalhamento é criada no ponto de contato. As forças das partículas circundantes farão com que as duas partículas comecem a deslizar uma após a outra com a rigidez ao cisalhamento resistindo ao movimento. Quando a força de cisalhamento excede a força normal multiplicada pelo coeficiente de atrito, o deslizamento dinâmico começa (a força de cisalhamento máxima é governada pela força normal e o coeficiente de atrito).

Os resultados apresentados na Figura 3 mostram aparentemente que este parece ser um método simples e eficaz para simular a deformação estática em contatos de atrito e deslizamento com atrito dinâmico. Além disto é possível observar também uma maior coesão entre as partículas que se encontram no centro do domínio. Comportamento semelhante desta coesão pode ser visualizado, por exemplo, em análises de solos arenosos e argilosos.

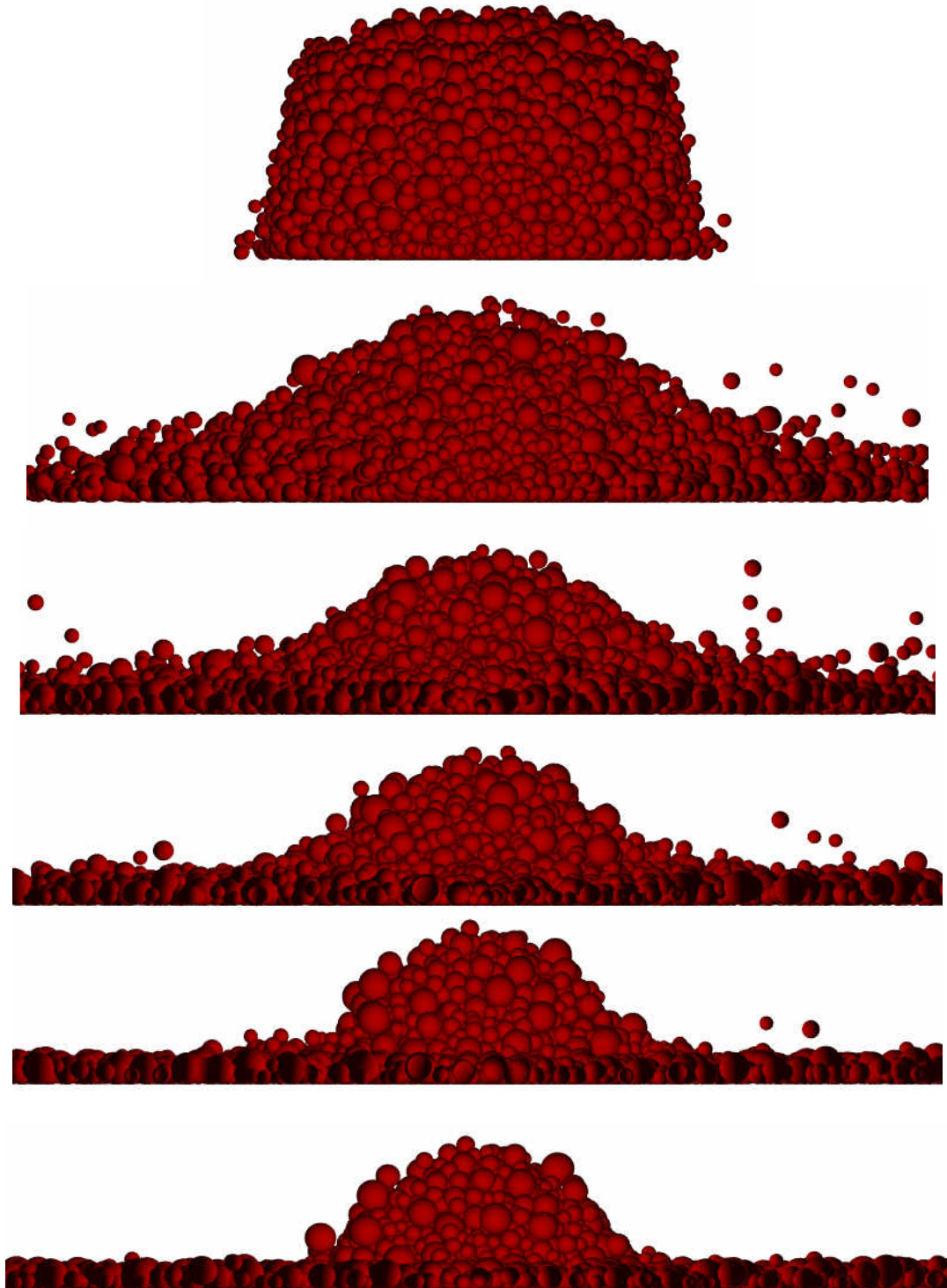


Figura 3. Colapso de um conjunto de partículas em uma base rígida

Considerando este mesmo modelo, para um número fixo de partículas, registramos as velocidades de aceleração numéricas em função do número de processadores n_p existentes na CPU. Neste caso, é possível observar que os resultados numéricos (linha tracejada) se aproximam do resultado teórico esperado (linha tracejada) (Figura 4).

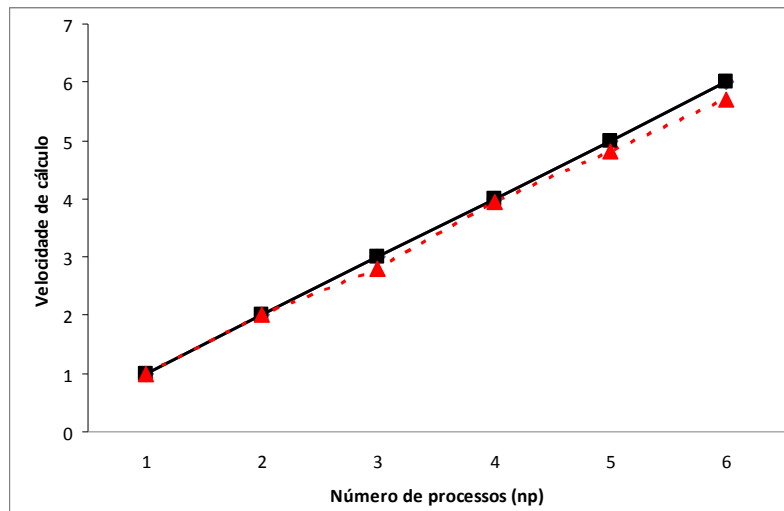


Figura 4. Velocidade de cálculo em função do número de processos n_p

A Figura 4 mostra que a velocidade de cálculo aumenta linearmente em função do aumento do número de processadores. Resultados muito semelhantes aos apresentados na literatura.

4.2 Fluxo de partículas em orifícios

Para avaliar o fluxo de partículas em um determinado orifício, consideramos um cubo quadrado contendo partículas de raios iguais de 0,2 mm (Figura 5).

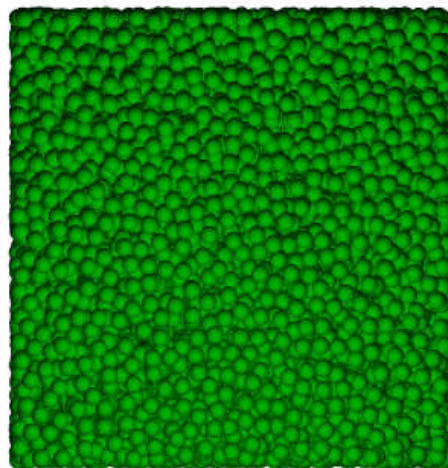


Figura 5. Domínio composto por partículas iguais para análise de fluxo em um orifício



Figura 6. Resultado do fluxo granular em um orifício

Foram incorporadas repulsão elástica e resistência ao atrito entre as partículas que se tocam entre si. O coeficiente de rigidez normal foi considerado de 1000,0 N/m, o tangencial de 100,0 N/m. Coeficiente de atrito 0,5. Coeficiente de aceleração da gravidade $9,81 \text{ m/s}^2$ e viscosidade igual a 0,58. Para as paredes, o coeficiente normal de rigidez é 10000,0 N/m. Considerando a geometria, um quarto do domínio foi modelado com partículas não rotacionais. Os ângulos de fricção e os limites da parede são ajustados a zero e as pressões são registradas no segmento médio para reduzir os efeitos de contorno. Executou-se a simulação para 10^5 interações com um incremento de 10^{-4s} no intervalo de tempo. O programa *POVray* foi usado novamente para gravar instantâneos a cada 10^3 passos de tempo. Diferentemente do primeiro modelo, as partículas apresentam o mesmo raio e foram definidos parâmetros de atrito e viscosidade menores. Assim, é possível observar uma menor coesão dessas partículas ao atravessar o orifício. Esquema idêntico foi usado por Weatherley et al. (2014) para modelar o fluxo em um orifício.

Adicionalmente, para verificar a eficiência da paralelização, definiu-se um número fixo de partículas no domínio. Registrou-se o tempo computacional sendo comparadas as respectivas velocidades com diferentes números de processadores n_p . Esta variável especifica o número de processos paralelos MPI usados na simulação (Figura 7). Neste caso, foram considerados seis núcleos de CPU físicos.

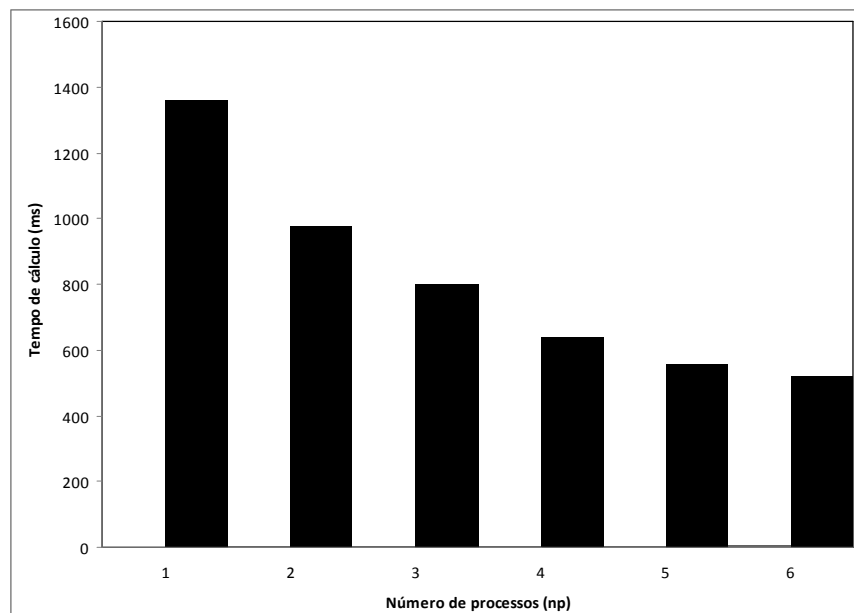


Figura 7. Tempo de cálculo em função do número de processos paralelos

Como esperado, a Figura 7 mostra que o tempo de cálculo diminui em função do aumento do número de processadores e, conseqüentemente, de processos paralelos. Resultados semelhantes também foram apresentados na literatura.



5 CONCLUSÕES

Neste trabalho, foi realizada uma implementação do Método dos Elementos Discretos, mais comumente chamado de método de partículas. Foi utilizado processamento paralelo em uma CPU com vários núcleos. O código foi testado em um computador pessoal considerando no cálculo até seis processadores. A proposta do trabalho foi modelar um problema granular de deposição de material e também o fluxo granular e um determinado orifício. Para esses dois problemas procurou-se apresentar o esquema geral do método de partículas. Nos dois modelos, foi possível verificar a eficiência da programação paralela. Entendemos que os resultados apresentados são bastante promissores e estão de acordo com os descritos na literatura. Investigações futuras precisam ser realizadas para verificar a efetividade de uma CPU com maior quantidade de processadores e processos paralelos.

AGRADECIMENTOS

Os autores agradecem a Comissão Organizadora do XIII Simpósio de Mecânica Computacional pela oportunidade.

REFERÊNCIAS

- Cundall, P. A., Strack, O. D. L., 1979. A discrete numerical model for granular assemblies. *Geotechnique*, v. 29, p. 47–65.
- Lukas, T., D'albano, G. G. S., Munjiza, A. A., 2014. Space decomposition based parallelization solutions for the combined finite-discrete element method in 2D. *Journal of Rock Mechanics and Geotechnical Engineering*. v. 6, p. 607-615.
- Munjiza, A. A., Knight, E. E., Rougier, E., 2012. *Computational Mechanics of Discontinua*. Wiley Series in Computational Mechanics. Chichester, UK: Wiley.
- O'Sullivan, C., 2011. *Particulate Discrete Element Modelling: a Geomechanics Perspective*. New York: Spon Press.
- Shigeto, Y., Sakai, M., 2011. Parallel computing of discrete element method on multi-core processors. *Particuology*, v.9, p. 398–405.
- Wang, L., Li, S., Zhang, G., Ma, Z. Zhang, L., 2013. A GPU-Based Parallel Procedure for Nonlinear Analysis of Complex Structures Using a Coupled FEM/DEM Approach. *Mathematical Problems in Engineering*. Article ID 618980.
- Weatherley, D. K., Boros, V. E., Hancock, W. R., Abe, S., 2014. *ESyS-particle Tutorial and User Guide Version 2.3.1*. The University of Queensland. Australia.