

MODELO COMPUTACIONAL DO TRANSPORTE DE PARTÍCULAS EM MALHA BLOCO-REFINADA

Vitor Maciel Vilela Ferreira, vilela.eng@gmail.com
João Marcelo Vedovoto, jmvedovoto@mecanica.ufu.br
Aristeu da Silveira Neto, aristeus@mecanica.ufu.br

Resumo. Vários são os tipos de escoamento que exigem a modelagem do transporte de partículas em domínio euleriano. A partícula computacional pode representar entidades de natureza distintas, como um particulado no escoamento gás-sólido, de natureza física, ou um elemento estocástico no escoamento reativo, de natureza numérica. O modelo computacional deste sistema, precedido pelos modelos físico, matemático e numérico, é determinante na eficiência do algoritmo. Fundamentalmente, requer-se que a estrutura de dados seja capaz de fornecer as informações do referencial lagrangiano, dadas as informações do referencial euleriano, e vice-versa. Este trabalho apresenta o transporte de partículas em malha bloco-refinada; o mapeamento do conjunto de elementos lagrangianos contido em cada volume euleriano é o seu resultado principal. Para gerenciar as informações dos dois referenciais, utiliza-se uma estrutura de dados do tipo hash table multinível. Nesta, as chaves dos primeiros níveis e do último nível são, respectivamente, os índices cartesianos do volume discreto e o identificador da partícula. Os processos de adição e transporte de elementos lagrangianos apresentaram $\Theta(n)$, enquanto o processo de localização do conjunto de elementos de um volume específico apresentou $\Theta(1)$. O modelo computacional mostrou-se eficiente para a malha bloco-refinada e capaz de atender aos requisitos impostos pelos modelos físico e numérico dos escoamentos com transporte de partículas.

Palavras chave: transporte de partícula, hash table, malha bloco-refinada

1. INTRODUÇÃO

Estruturas de dados tem a função de gerenciar a armazenagem e a aquisição de dados na memória ou em disco. Algumas delas são baseadas em *array*, nas quais os dados são ordenados e referenciados pelo seu índice inteiro, outras em *linked list*, cujos dados são ligados através de ponteiros de alocação dinâmica, e ainda em *hash table* (Shaffer, 2013).

Hash table é uma estrutura de dados que, a semelhança de *array*, possui referência direta ao valor armazenado. Esta referência é realizada através de uma chave, que pode ser do tipo inteiro, tipo definido pelo usuário, entre outros. Desta forma, é importante que, dado um conjunto de dados, a cada elemento seja atribuída uma chave, a qual permitirá a armazenagem em uma única posição na memória ou no disco. Esta regra de atribuição de chaves é denominada função *hash* (Shaffer, 2013).

Sistemas possuem dados, e dados são caracterizados por atributos. Quando é necessário que a armazenagem seja realizada com base em um atributo que compartilha o mesmo valor entre um subgrupo de dados, ocorrerá o que se denomina colisão. A colisão é caracterizada pela tentativa de sobrescrever um valor já armazenado pela mesma chave (Shaffer, 2013).

Como exemplo de um sistema no qual a colisão é inerente, tem-se o transporte de partículas lagrangianas entre volumes eulerianos. Nesse caso, um dos atributos chave da partícula que gerará a colisão é o índice do volume euleriano, ao qual várias partículas pertencem. Desta forma, é necessário que exista um gerenciamento através de técnicas denominadas *open hashing* ou *closed hashing* (Shaffer, 2013).

Gerenciar colisões significa atribuir uma nova chave a um dado cuja chave anteriormente a ele associada aponta para uma posição na memória já ocupada por outro dado. Quando esta atribuição de nova chave visa à solução da colisão com o armazenamento do dado na mesma *hash table*, tem-se a técnica *closed hashing*. Se o dado é armazenado em outra *hash table* ou estrutura de dados, tem-se a técnica *open hashing* (Shaffer, 2013).

A *hash table* multinível é formada quando cada elemento da *hash table* contém sua própria *hash table*. Esta estrutura propicia a técnica *open hashing* de gerenciamento de colisões.

O *uthash* foi utilizado no presente trabalho. Desenvolvido no ano de 2006 por Troy D. Hanson, *uthash* é um arquivo *header* que fornece uma estrutura *hash table* para plataformas computacionais desenvolvidas em linguagem C. Ele é baseado em macros e está disponível sob a licença BSD revisada.

2. METODOLOGIA

A *hash table* possui referência direta ao valor armazenado em memória e tem comportamento dinâmico. Estas características são fundamentais para se construir o modelo computacional do transporte de partículas em malha bloco-refinada.

A Figura 1 mostra o conjunto de estruturas de dados que compõem o modelo computacional proposto.

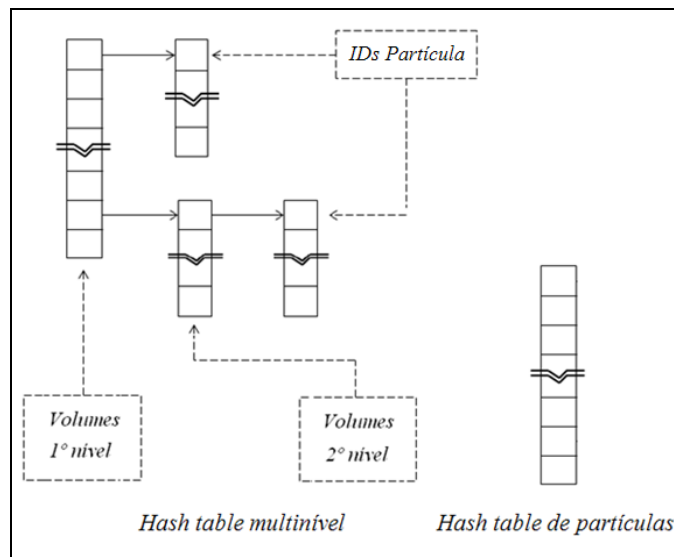


Figura 1 – Modelo computacional composto por duas estruturas de dados do tipo *hash table*. À esquerda, *hash table* multinível responsável pelo mapeamento das partículas. À direita, estrutura que armazena todas as informações dos elementos lagrangianos.

A *hash table* multinível armazena dados do tipo volume, cujos índices cartesianos são suas informações principais, e do tipo identificador de partícula, que possui apenas um valor inteiro. Observa-se que, na estrutura composta, cada volume da primeira *hash table* aponta para outra *hash table*, a qual pode ser constituída por volumes ou identificadores de partícula.

A *hash table* de partículas é uma estrutura de dados independentemente alocada, que utiliza o identificador de partícula da *hash table* multinível como chave e armazena todos os dados associados ao modelo físico do elemento lagrangiano. Desta forma, o custo computacional de inserção e exclusão de dados no processo de transporte visando o mapeamento é aquele relativo a um inteiro, enquanto todas as informações da partícula (e.g., propriedades físicas) mantêm seu endereço na memória inalterado.

O domínio euleriano, onde ocorre o transporte de partículas, é composto por um conjunto de volumes discretos; este conjunto é denominado malha numérica/computacional e pode ser dividido em processos distintos, visando o processamento paralelo.

No presente trabalho, a malha, do tipo bloco-refinada, possui dimensões globais unitárias e é composta por dois *patches* por processo (Figura 2). Os *patches* base e refinado possuem, respectivamente, 2.560 e 12.288 volumes de formato cúbico, com distribuição uniforme no domínio. O particionamento de processos é realizado na posição $x = 0.5$ m.

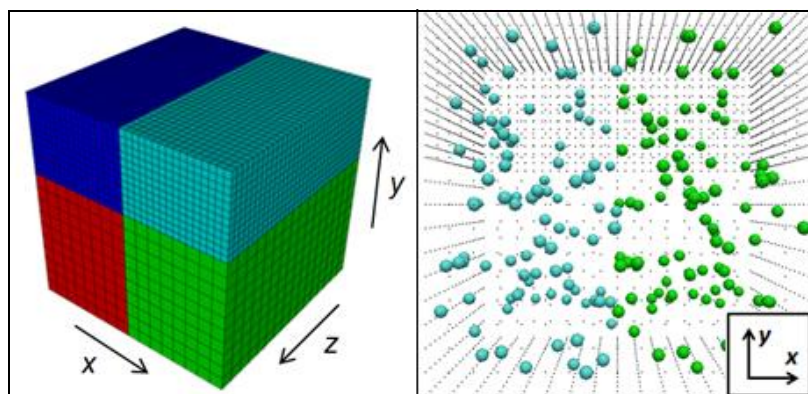


Figura 2 – À esquerda, malha bloco-refinada em dois níveis, com processamento paralelo; as regiões de cor vermelha/azul e verde/ciano representam malhas em processos distintos. À direita, distribuição uniforme de 200 partículas no domínio global; as cores distinguem à qual processo cada partícula pertence.

A posição inicial das partículas é aleatoriamente definida de forma a resultar em uma distribuição uniforme nas três direções cartesianas. Posteriormente, os identificadores das partículas são adicionados na *hash table* multinível, de acordo com o volume discreto que as contém. O total de 100 partículas é gerado para cada processo, conforme ilustra a Figura 2.

3. RESULTADOS

A cada passo temporal da solução numérica é realizado um mapeamento que relaciona a posição dos elementos lagrangianos e os índices dos volumes discretos.

Às partículas é imposta uma velocidade predominante na direção x , as quais se movimentam na malha bloco-refinada, paralela, de 14.848 volumes. A condição de parede é adotada para os contornos dos dois processos.

A Figura 3 ilustra a posição das partículas em seções xy para seis instantes da simulação. A coloração destas é dada por uma função tangente hiperbólica.

O modelo computacional foi capaz de apresentar o mapa de partículas corretamente, além de garantir o processo de inserção e exclusão de elementos a cada vez que estes mudam de volume. Como esperado, no instante final as partículas tem seu movimento restringido pela condição de contorno do processo.

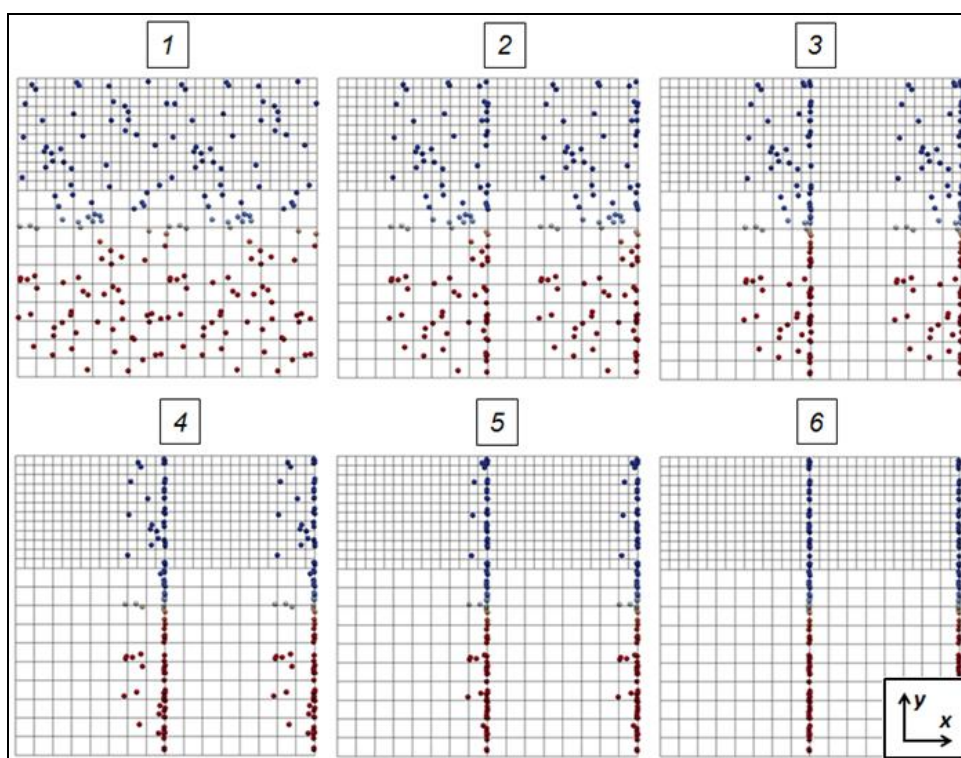


Figura 3 – Transporte de partículas em malha bloco-refinada com mapeamento lagrangiano em domínio euleriano; seções xy em seis instantes da simulação.

Duas questões surgem quando o transporte de partículas em domínio euleriano composto por volumes discretos é observado: Dado um volume, quais partículas estão geometricamente contidas nele? E, dada uma partícula, ao qual volume ela pertence? A resposta à primeira questão é dada pelo modelo computacional baseado em *hash table*, através do processo de mapeamento apresentado. A segunda questão é solucionada através de um algoritmo de busca.

Com relação às operações envolvendo a estrutura de dados, os processos de adição e transporte de elementos lagrangianos apresentaram $\Theta(n)$, sendo n o número de elementos; enquanto os processos de localização do conjunto de elementos contidos em um volume específico e do volume ao qual uma dada partícula pertence apresentaram $\Theta(1)$.

4. AGRADECIMENTOS

Os autores agradecem ao CNPq, à CAPES e à FAPEMIG pelo apoio financeiro.

5. REFERÊNCIAS

Shaffer, A. C., 2013, “Data Structures and Algorithm Analysis”, Edition 3.2, Dover Publications.

6. RESPONSABILIDADE PELAS INFORMAÇÕES

Os autores são os únicos responsáveis pelas informações incluídas neste trabalho.