



1.1 Geração de grid de partículas a partir de sólidos modelados com malha de triângulos

Davi Teodoro Fernandes

davifernandes@usp.br

Tanque de Provas Numérico, Escola Politécnica, Universidade de São Paulo

Av. Prof. Mello Moraes, n. 2231, Cidade Universitária, 05508-970, São Paulo, Brazil.

Liang-Yee Cheng

cheng.yee@poli.usp.br

Depto. de Engenharia de Construção Civil, Escola Politécnica, Universidade de São Paulo

Av. Prof. Almeida Prado, trav. 2, n. 83 Edifício de Engenharia Civil, Cidade Universitária, 05508-070, São Paulo, Brasil.

Rodrigo Carneiro da Silva

rodrigo.carneiro.silva@usp.br

Depto. de Engenharia Naval e Oceânica, Escola Politécnica, Universidade de São Paulo

Av. Prof. Almeida Prado, trav. 2, n. 83 Edifício de Engenharia Civil, Cidade Universitária, 05508-070, São Paulo, Brasil.

Eric Henrique Favero

eric.favero@usp.br

Depto. de Engenharia de Construção Civil, Escola Politécnica, Universidade de São Paulo

Av. Prof. Almeida Prado, trav. 2, n. 83 Edifício de Engenharia Civil, Cidade Universitária, 05508-070, São Paulo, Brasil.

Kazuo Nishimoto

knishimo@usp.br

Depto. de Engenharia Naval e Oceânica, Escola Politécnica, Universidade de São Paulo

Av. Prof. Mello Moraes, n. 2231, Cidade Universitária, 05508-970, São Paulo, Brasil.

Resumo. *Devido a capacidade de simulação de fenômenos altamente não lineares superior aos métodos tradicionais com malha, os métodos baseados em partículas, como o Moving Particle Simulation (MPS), tem recebido muita atenção nos anos recentes. No entanto,*

apesar da relativa facilidade e simplicidade do pré-processamento dos métodos de partículas, que não requerem a geração de malhas indispensáveis nas técnicas convencionais, são altamente ferramentas eficientes e eficazes de criação de grids de partículas. Dentro deste contexto, o aproveitamento dos recursos oferecidos por software para a modelagem geométrica de sólidos, como os aplicativos CAD, para agilizar a geração de grid de partículas de sólidos com geometrias complexas se torna uma alternativa extremamente atraente. Atendendo esta demanda, neste trabalho foi desenvolvido um sistema de pré-processamento que se baseia na conversão de modelos geométricos definido por malha de triângulos em grid de partículas para simulação do MPS. A geração de grids para o MPS compreende a localização das partículas da superfície do sólido e de duas camadas de partículas dummies adjacentes à superfície. Para isso são utilizados modelos descritos por malha de triângulos segundo o formato STL (estereolitografia), que é nativo nos principais sistemas CAD 3D e amplamente usada para prototipagem rápida e arquivos de manufatura auxiliada por computador. Para identificação de pontos interiores à malha de triângulos foi implementado uma adaptação do algoritmo de Möller-Trumbore, que utiliza a técnica de Ray-casting a fim de obter o número de vezes que um raio semi-infinito com origem no ponto cruza as faces do modelo. Como resultado, estruturas de geometrias complexas em escoamentos interno, externo e com superfície livre, como é o caso da dinâmica de multi-corpos flutuantes acoplados aos efeitos de sloshing, podem ser modeladas e simuladas com facilidade usando o método de partículas. A solução proposta foi avaliada com modelos de sólidos de complexidade crescente e utilizada com sucesso para geração de modelos de partículas para plataforma FPSO (Floating Production Storage and Offloading). Por fim, apesar da técnica de pré-processamento desenvolvida neste trabalho estar focada no método MPS, os fundamentos adotados também podem ser estendidos e aplicados a outros métodos de simulação baseados em partículas com características similares.

Palavras-chave: *métodos sem malha, pré-processamento, CFD, Moving Particle Simulation, simulação*

2 INTRODUÇÃO

Na última década, tendo em vista a facilidade de rastrear a superfície livre submetida a grande deformação, fragmentação e junção, e de trabalhar com contornos complexos e móveis, multicorpos, escoamentos multifásicos e multifísicos, o método *Moving Particle Simulation* (MPS) foi extensamente aplicado nos estudos de hidrodinâmica não linear (Shibata et al., 2012; Tsukamoto, Cheng, Nishimoto, 2011), na modelagem de escoamentos com múltiplos fluidos (Shakibaeinia, Jin, 2012), fenômenos multifísicos como acidentes severos em reatores nucleares (Chen et al., 2014; Morita et al., 2011), interação fluido estrutura e hidroelasticidade (Akimoto, 2013; Amaro, Cheng, 2013), além de aplicações na área de computação gráfica (Koshizuka, 2011).

Por ser um método sem malha, que discretiza o domínio em partículas lagrangeanas de dimensões uniformes, o método MPS apresenta a grande vantagem de possuir um custo relativamente baixo de pré-processamento mesmo quando se trata de geometrias muito complexas. No entanto, em problemas reais de engenharia, por exemplo, o escoamento em sistemas de esgotos prediais ou industriais e a hidrodinâmica de plataformas flutuantes, a quantidade de partículas pode chegar na ordem de centenas de milhões ou mais e, assim como os métodos de CFD tradicionais baseados no uso de malha, torna-se imprescindível sistemas amigáveis de pré-processamento para a geração da distribuição inicial de partículas, denominado de *grid*, de modo a aumentar a eficiência e eficácia do processo de simulação.

Dentro deste contexto, este trabalho tem como objetivo um sistema de pré-processamento que permite a interpretação de figuras geométricas tridimensionais de difícil descrição matemática e *grids* de partículas para a simulação usando o método MPS, usando como dados de entrada modelos geométricos criados em *software* de CAD e descritos na forma de malhas de faces triangulares. Além de reduzir o custo de pré-processamento de simulações usando o método MPS, a técnica proposta também irá facilitar o desenvolvimento de métodos híbridos de simulação, que aproveite o melhor dos métodos eulerianos com malhas e dos métodos lagrangeanos baseados em partículas, provendo bases para a geração automática de *grids* de partículas durante a simulação. Vale ressaltar que apesar da técnica de pré-processamento desenvolvida neste trabalho estar focada no método MPS, os fundamentos adotados também podem ser estendidos e aplicados nos métodos de simulação baseados em partículas com características similares.

A seguir será feita uma descrição bem sucinta do método *Moving Particle Simulation*, com base no qual o pré-processador apresentado neste trabalho foi desenvolvido, mostrando os pontos essenciais do método relacionados aos requisitos do pré-processamento. Em seguida, as técnicas de geração de *grids* de partículas, assim como exemplos de resultados obtidos, são apresentadas e analisadas.

3 MOVING PARTICLE SIMULATION

O método *Moving Particle Simulation* (MPS), inicialmente denominado *Moving Particle Semi-implicit* para sua versão semi-implícita, é um método lagrangeano sem malha que discretiza o espaço por meio de partículas. Originalmente proposto por Koshizuka et al. (Koshizuka, 1995) para simulação de escoamentos incompressíveis com superfície livre, o método resolve as equações governantes de meios contínuos substituindo os operadores

diferenciais por operadores diferenciais discretos em uma distribuição de nós irregulares (Isshiki, 2011), derivados a partir de um modelo de interação entre partículas baseado numa função peso que pondera a contribuição de todas as partículas interiores a um raio de vizinhança.

O algoritmo de solução do *Moving Particle Simulation* pode ser semi-implícito ou explícito. O algoritmo semi-implícito é baseado na estimativa de velocidade, posição e densidade de partículas, solução do sistema linear de equações de Poisson de pressão e a correção das velocidades e posições das partículas. O algoritmo explícito é baseado no uso da equação de estado para relacionar densidade e pressão.

No MPS, as partículas nada mais são que pontos de observação lagrangeanos. Essencialmente o método lida com partículas uniformemente distribuídos de modo que a densidade de partículas é proporcional à densidade do fluido.

Conforme mostra a Fig. 1, no MPS os tipos básicos de partículas que formam o domínio fluido são: partículas de fluido e partícula de superfície livre. As partículas de superfície livre são aquelas localizadas na interface ar-líquido. Utilizadas para rastrear a superfície livre, elas são facilmente detectadas utilizando critérios que apontam para falta de partículas na sua região de vizinhança. Como condição dinâmica do contorno, a pressão atmosférica é imposta nas partículas de superfície livre. A condição cinemática do contorno é satisfeita pelo próprio deslocamento da partícula de superfície livre.

Por outro lado, normalmente as paredes também são modeladas utilizando partículas. Para isso, a forma mais comum é definir uma camada de partículas de parede, que é a camada de partículas de sólido que pode entrar em contato com as partículas de fluido, e mais duas camadas de partículas *dummy* ou partícula de parede auxiliar, que são utilizadas para garantir o correto cálculo da densidade de partículas na região vizinha às partículas de parede.

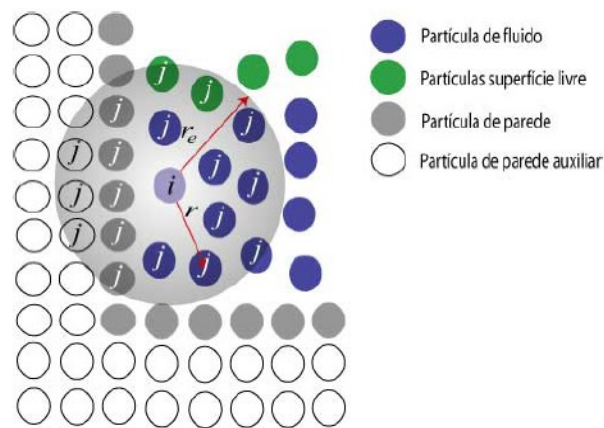


Figura 1. Tipos básicos de partículas de partículas e a região de influência de uma partícula qualquer (Tsukamoto, 2006).

4 TÉCNICAS DE GERAÇÃO DE MODELOS DE PARTÍCULAS

Por ser um método sem malha, o método MPS apresenta uma grande vantagem operacional que é a facilidade de pré-processamento mesmo para casos com geometrias complexas. Isso porque considerando a modelagem do domínio fluido e das condições de contorno de paredes de sólido mostradas na seção anterior, os requisitos essenciais do pré-processamento do MPS são a geração de um *grid* de partículas equidistantes entre si, e a

atribuição correta dos quatro tipos básicos de partículas: de fluido, superfície livre, parede e *dummies*. Vale ressaltar que como as partículas de superfície livre mudam a cada instante conforme a deformação da superfície, elas passam por uma rotina de detecção em todo passo de tempo ao longo da simulação. Sendo assim, na geração do *grid* inicial de partículas para a simulação, não há necessidade de distinguir as partículas de superfície livre das partículas de fluido.

4.1 Geração de *grid* de partículas por imagens de *bitmap*

Para estudos bidimensionais, o trabalho pode ser simplificado ao extremo usando imagens *bitmap* para descrever o *grid* de partículas. Nesta abordagem, cada partícula do *grid* inicial é representada por um pixel da imagem e as coordenadas de um pixel são convertidas para as coordenadas da partícula a ele associada. A coloração do pixel é usada como parâmetro para distinguir os diferentes tipos de partículas. Esta abordagem tem sido intensivamente usada para simulações 2D como aquelas reportadas em Cheng et al. (2011), Pereira et al. (2013).

A extensão desta abordagem para casos 3D é relativamente direta e simples para casos envolvendo geometrias que podem ser geradas por varredura translacional perpendicular à seção, por meio da repetição das figuras que representam as seções transversais. Com alguns esforços adicionais o *grid* de partículas de geometrias um pouco mais complexas também pode ser gerado com base nesta abordagem. No entanto, para uma geometria 3D arbitrária, faz-se necessária outra solução mais eficiente.

4.2 Geração de *grid* 3D de partículas a partir de primitivas geométricas e operações booleanas regularizadas

Uma das soluções eficientes para a geração do *grid* 3D de partículas é o pré-processador denominado de *Burner*, desenvolvido no Tanque de Provas Numérico da USP (TPN-USP), e inicialmente utilizado no estudo de Cheng et al. (2012).

A essência do algoritmo do *Burner* é verificar, para qualquer ponto (x, y, z) , se ele está contido em uma região. Para a geração de um *grid* de partículas utilizando *Burner*, é necessário especificar uma função booleana, denominada *Region*, que retorna verdadeiro para pontos no interior da região do modelo e falso caso contrário. Por exemplo, uma região esférica pode ser implementada como:

$$sphere(C, r, x) = ((x_1 - C_1)^2 + (x_2 - C_2)^2 + (x_3 - C_3)^2) < r^2.$$

Inicialmente, o algoritmo de geração utiliza a função *Region* para classificar os pontos de um domínio, especificando-os como internos, externos ou sobre o contorno da região. Em seguida, são geradas as partículas de parede, para cada um dos pontos sobre o contorno, e as duas camadas de partículas *dummies*. Estas últimas serão geradas adjacentes ao lado externo, ou interno, da parede conforme selecionado previamente. Para simplificar, a posição das partículas do *grid* são previamente definidas como os nós de uma malha estruturada uniforme com uma resolução ou espaçamento pré-definidas, de modo que o papel do pré-processador *Burner* é a checagem e atribuição de tipo de partícula para cada um dos nós.

Além de objetos de geometrias mais simples, o algoritmo também pode ser aplicado para geração de lista de partículas para objetos com formas mais complexas. Para isso, pode ser utilizado um conjunto de equações que descrevem o contorno do objeto ou que delimitam a *Region* correspondente.

Além disto, o *Burner* também implementa algoritmos, baseados na Geometria de Sólidos Construtivos (*Constructive Solid Geometry*), que permitem a composição de sólidos usando operações booleanas regularizadas, tais como: união, diferença (subtração) e interseção - para combinação de sólidos primitivos. A Fig. 2 fornece um exemplo de um *grid* 3D de partículas gerado pelo *Burner*.

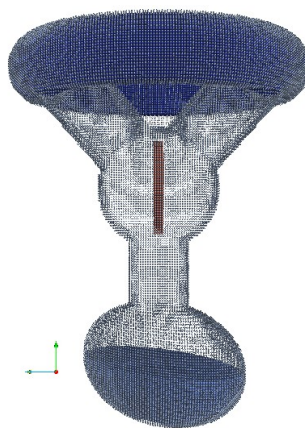


Figura 2. Exemplo de um *grid* 3D de partículas gerado por *Burner*

No entanto, apesar do *Burner* conseguir combinar regiões para gerar o *grid* de objetos complexos, para geometrias mais complexas ainda, como, por exemplo, casco de navios, ou dispositivos mais elaborados, como uma planta industrial, a geração por composição de equações ou primitivas não é fácil nem intuitiva. Sendo assim, para incorporar os recursos e as facilidades proporcionadas por software CAD ao procedimento de pré-processamento de modelos de partículas, foi desenvolvida uma extensão do *Burner* que utiliza como dados de entrada arquivos de formato STL.

4.3 Geração de *grid* 3D de partículas a partir de modelo de malha de triângulos

O formato STL (estereolitografia) é um formato de arquivo nativo para os principais sistemas CAD 3D e pode ser em ASCII (texto) ou binário. Amplamente usada para prototipagem rápida e arquivos de manufatura auxiliada por computador (CAM), um arquivo no formato STL é uma representação por fronteiras (*Boundary Representation*) de um objeto tridimensional, e descreve somente a geometria da superfície do objeto, sem considerar a cor, a textura ou de outros atributos comuns em modelos CAD. A representação do objeto é feita por uma malha triangular não estruturada, representada em um sistema de coordenadas cartesianas tridimensional, por meio da especificação do vetor normal unitário e de três vértices, orientados segundo a regra da mão direita, para cada face triangular. Coordenadas STL devem ser números positivos, não há nenhuma informação de escala e as unidades são definidas arbitrariamente na modelagem dos sólidos.

A parte inicial deste trabalho consiste em adicionar ao *Burner* uma *Region stlmodel(*stlfile)*, que tem como parâmetro de entrada um arquivo STL. Para isso foram implementadas duas novas bibliotecas ao aplicativo, *stlfile* e *CstlFile*, a primeira com intuito de possibilitar a leitura do arquivo e a segunda para criar uma classe que permite armazenamento e acesso às informações obtidas através dessa leitura de modo que ela só precise ser realizada uma única vez.

Depois da leitura da malha de triângulos, que representa a superfície de um objeto modelado usando CAD, é necessário verificar se um ponto qualquer de coordenadas (x, y, z) encontra-se dentro ou fora do modelo tridimensional. Para isso foi usado uma adaptação do algoritmo de Möller-Trumbore (Horvat, D., 2012). Esse algoritmo usa uma técnica conhecida como *Ray-casting*, que consiste em usar o ponto O que está sendo verificado como origem de um raio semi-infinito com direção arbitrária D , a fim de obter o número de vezes que ele intercepta e cruza com as faces triangulares do objeto. Se o número do cruzamento for um número ímpar, o ponto está contido na região, caso o número for zero ou par, o ponto é externo a figura (Fig.3).

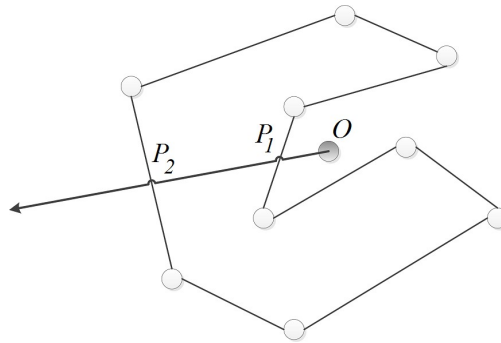


Figura 3. Ilustração da técnica de *Ray-casting*

Para cada raio, a verificação do cruzamento deve ser aplicada a todas as faces do objeto, ou seja, todos os triângulos da malha, para garantir que o número de faces triangulares cruzadas pelo raio semi-infinito seja o correto. Assim, o algoritmo começa recebendo como informação o ponto que está sendo verificado, uma direção para o raio e a lista com os vértices de cada um dos triângulos, (Fig. 4).

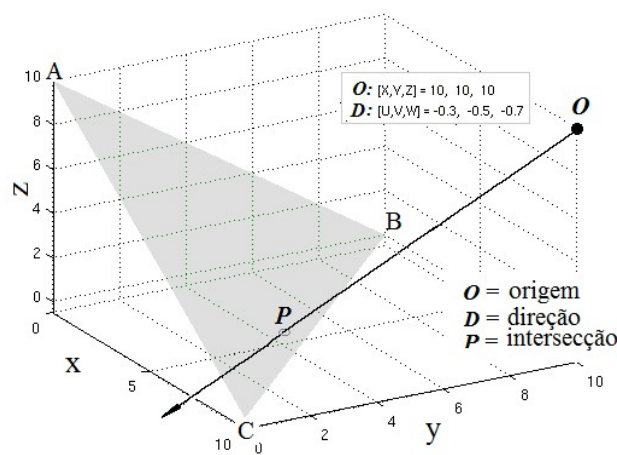


Figura 4. Ilustração do funcionamento do algoritmo de Möller-Trumbore (Horvat, 2012)

Para cada triângulo ABC da malha, define-se primeiro o plano em que o triângulo está contido e o vetor normal N do plano usando o produto vetorial dos vetores associados aos lados com origem num vértice em comum. Em seguida, determina-se o ponto de intersecção entre o raio e o plano do triângulo: se o produto escalar entre o raio e a normal ao plano for nulo, o plano e o raio são paralelos e, portanto, não há intersecção com o triângulo. Caso contrário, é calculada a expressão paramétrica para ponto de intersecção:

$$\mathbf{P} = \mathbf{O} + a \cdot \mathbf{D}, \quad (1)$$

onde a é um número real e $\mathbf{O}$ e $\mathbf{D}$ são, respectivamente, a origem e a direção do raio. Se $a < 0$, o ponto de intersecção pertence à semirreta oposta ao raio semi-infinito e deve ser descartado. Caso contrário, isto é, $a \geq 0$, é utilizado um algoritmo de teste de ponto em polígono, como apresentado em (Haines, 1994), para verificar se $\mathbf{P}$ está no interior do triângulo: se $\mathbf{P}$ se encontra no interior da região delimitada pelas arestas $\mathbf{AB}$, $\mathbf{AC}$, e $\mathbf{CA}$, as seguintes condições devem ser satisfeitas simultaneamente:

$$\begin{aligned} (\mathbf{BA} \times \mathbf{PA}) \cdot \mathbf{N} &\geq 0 \\ (\mathbf{CB} \times \mathbf{PB}) \cdot \mathbf{N} &\geq 0 \\ (\mathbf{AC} \times \mathbf{PC}) \cdot \mathbf{N} &\geq 0 \end{aligned} \quad (2)$$

Da mesma forma que o *burner*, para gerar o grid de partículas para a simulação, a posição das partículas do grid é estabelecida considerando os nós de uma malha estruturada uniforme seguindo uma resolução ou espaçamento pré-definido. Por questão de simetria e outras características geométricas, a precisão da modelagem de um objeto por partículas de sólido geralmente é melhorada quando estas são arranjadas tomando como base o sistema de coordenadas local alinhado com os eixos principais do objeto.

No entanto, em muitos casos de interação fluido-estrutura, o objeto pode estar posicionado de uma forma que seus eixos principais não estejam alinhados com o sistema de coordenadas global, associado ao grid cúbico, que normalmente serve como referência para o alinhamento do arranjo das partículas de fluido. Este problema é particularmente crítico para o desenvolvimento de técnicas híbridas de simulação mencionadas anteriormente, onde os resultados dos métodos com malha, com os objetos móveis numa determinada posição arbitrária, podem ser aplicados como condições iniciais métodos de partículas. Sendo assim, além da geração das partículas dos objetos e as partículas do domínio seguindo seus respectivos eixos de coordenadas, é necessário fazer uma compatibilização entre os dois.

Neste trabalho a compatibilização é feita de uma forma simples, partindo da sobreposição parcial ou total entre as partículas do objeto e as partículas de fluido do domínio. Em primeiro lugar, gera-se o *grid* de partículas de cada um dos objetos, com arranjo alinhado aos eixos do objeto que representam. Em seguida, aplica-se a rotação em cada um dos *grid* de partículas de modo a orientá-los conforme a inclinação dos respectivos objetos em relação ao sistema global. Em seguida, os *grids* são transladados e posicionados corretamente no meio do domínio fluido, já modelado com partículas de fluido. Por fim, faz-se a exclusão de partículas de fluidos da região de intersecção entre os dois.

Para excluir todas as partículas de fluido que ficariam dentro do objeto ou em sobreposição com as paredes, foi usado um método baseado na particularidade dos *grids* usados para esse tipo de simulação: além de uma camada de partículas de parede do objeto, este deve possuir mais duas camadas de partículas *dummies*. O algoritmo verifica a distância entre cada partícula de líquido até as partículas de parede. Se a distância até qualquer uma das partículas de parede for menor que a distância padrão entre partículas, a partícula de líquido é excluída. Caso contrário, verifica se a partícula mais próxima é de parede ou de *dummies*, se for parede, a partícula de líquido é exterior ao objeto, caso contrário ela está na região interna do objeto e é excluída. Dessa forma, temos como resultado um *grid* de partículas de objeto que deve se encaixar bem num *grid* de partículas de fluido, mesmo após o primeiro sofrer alguma rotação.

É lógico que a técnica não garante que o arranjo das partículas na interface seja ajustado perfeitamente, sem pequenos vazios que podem interferir inicialmente no cálculo do valor local de densidade e provocar algumas instabilidades no cálculo de pressão. No entanto, alguns melhoramentos podem ser feitos para minimizar os efeitos deste problema. Por exemplo, ao invés de usar a superfície do objeto como a interface entre os dois arranjos diferentes, pode-se deslocar esta interface para uma distância finita da superfície do objeto para afastar do objeto as fontes de instabilidade.

5 RESULTADOS

O primeiro modelo testado foi um poliedro de 26 faces (Fig. 5a), ele foi escolhido porque, apesar do formato visualmente simples, é um sólido relativamente trabalhoso de ser definido matematicamente. Para representar a superfícies do poliedro, uma malha triangular gerada por *software* CAD apresenta uma quantidade total de 92 triângulos, conforme ilustrado na Fig. 5a.

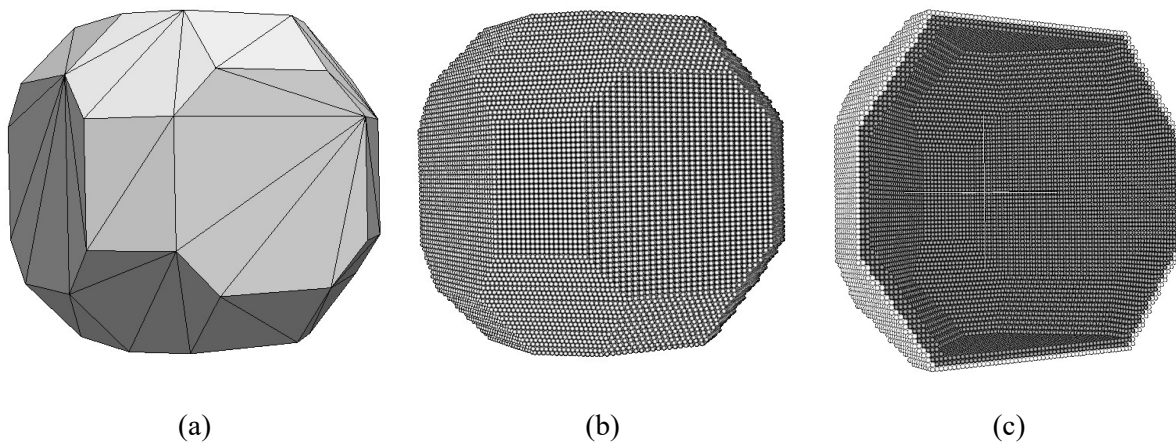


Figura 5. Poliedro de 26 faces com sua malha triangular (a), *grid* de partículas do poliedro (b) e a vista do modelo de partículas em corte (c)

A Fig. 5b mostra que a parte externa do *grid* de partículas gerado pela técnica proposta para representar o poliedro da Fig. 5a. Externamente, na resolução considerada, o *grid* de partículas reproduz muito bem a geometria do objeto convexo fornecida pelo arquivo STL original e a Fig. 5c mostra a perspectiva em corte do poliedro, indicando que as partículas internas (*dummies*) foram geradas corretamente.

O segundo modelo escolhido para o teste foi uma garrafa, que possui uma geometria mais complexa, apresentando concavidades. Para representação deste modelo no formato STL, foi necessária a construção de uma malha com cerca de 1240 triângulos. Na Fig. 6.a o ajuste do modelo de partículas à malha de triângulos é mostrado pela sobreposição de uma vista em corte do modelo de partículas sobreposto à *grid* de triângulos. Na Fig. 6.b as duas vistas complementares em corte mostram o interior da garrafa, enquanto o detalhe apresentado na Fig. 6c mostra a tampa interna presente no gargalo da garrafa. Estas imagens confirmam que também o interior da garrafa foi gerado com sucesso.

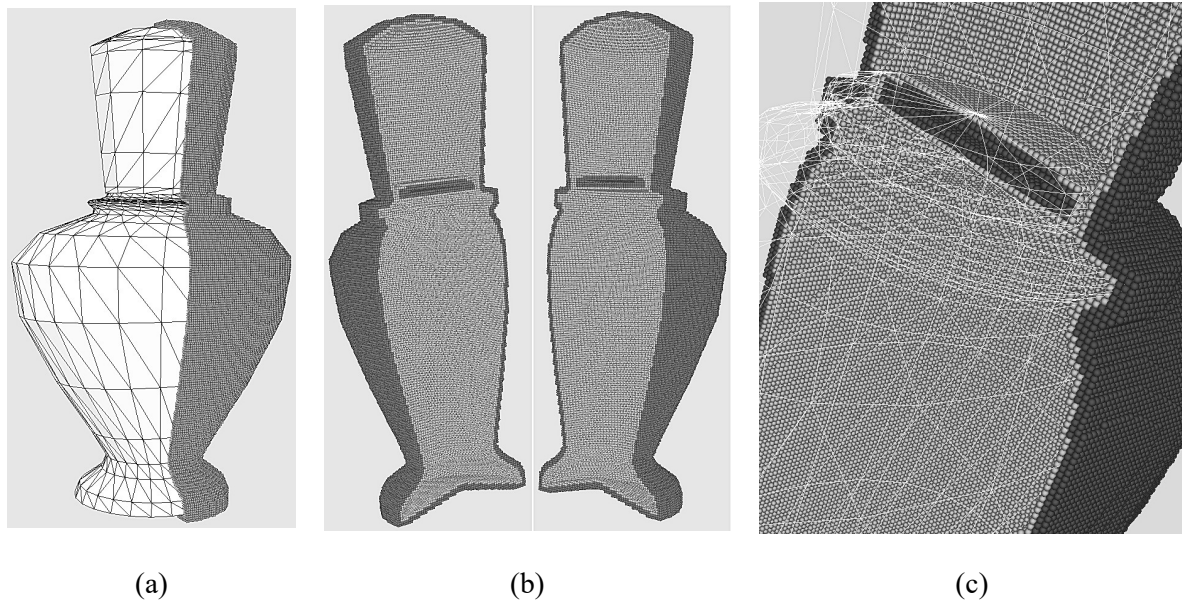


Figura 6. Segundo modelo de testes, garrafa representada por uma malha de 126 faces triangulares (a), vista do modelo de partículas em corte (b) e detalhe do *grid* de partículas da garrafa mostrando a tampa interna no gargalo da garrafa (c)

O terceiro modelo usado para testar a técnica proposta é um caso real, de interesse para engenharia. Trata-se de um casco de uma plataforma FPSO (*Floating Production Storage and Offloading*), apresentado na Fig. 7. Trata-se de um caso que é um dos focos da pesquisa da qual o presente estudo está inserido.

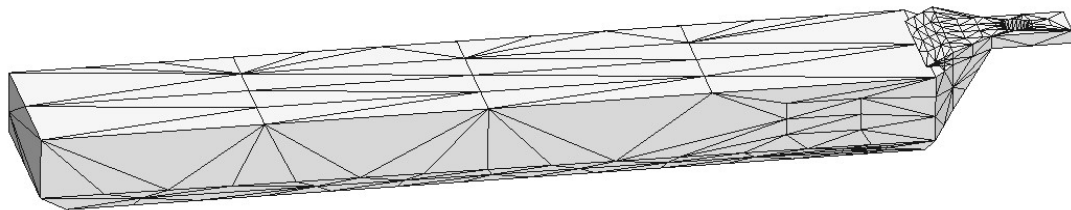


Figura 7. Modelo de um casco de FPSO no formato STL

A Fig. 8 ilustra a parte externa do *grid* de partículas, enquanto o resultado da parte interior é apresentado na Fig. 9. O resultado obtido mostrou que o pré-processador gera com exatidão as partículas do modelo de casco do FPSO, e pode ser usado para simulações de problemas reais.

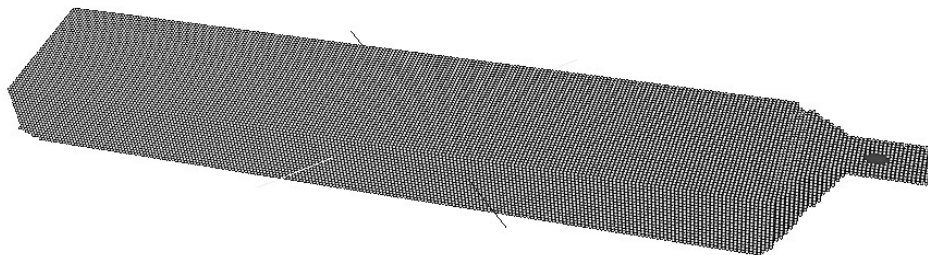


Figura 8. *Grid* de partículas do casco da plataforma FPSO

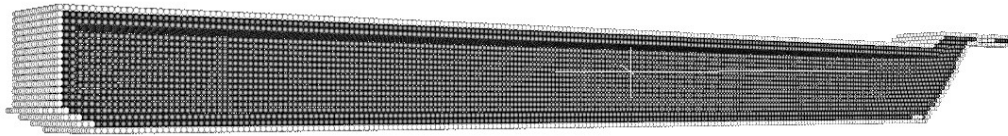


Figura 9. Perspectiva em corte do *grid* de partículas do casco da plataforma FPSO

Após a verificação do *grid* do casco do FPSO, foi feita a verificação do algoritmo de compatibilização. Para isso o modelo de partículas do casco de FPSO foi rotacionado de 26° no eixo XY do sistema global, e foi encaixado em meio a partículas de líquido. A Fig. 10 mostra a vista superior do *grid* da plataforma gerado já inserido no meio fluido.

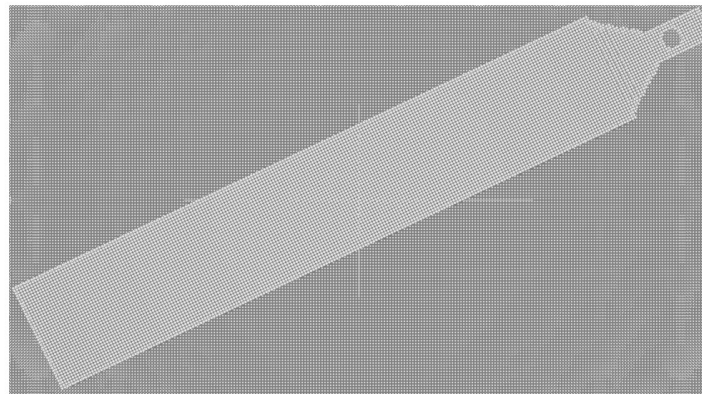


Figura 10. Vista superior do *grid* de partículas do casco da plataforma FPSO juntamente com as partículas do domínio do fluido

A Fig. 11 mostra a vista em corte do detalhe da popa da plataforma destacando o encaixe do casco no meio fluido. Da figura é possível verificar que não existem partículas de fluido na parte interna do casco, e a camada de partículas de parede e as duas camadas de partículas *dummies*, no interior do casco, foram geradas corretamente.

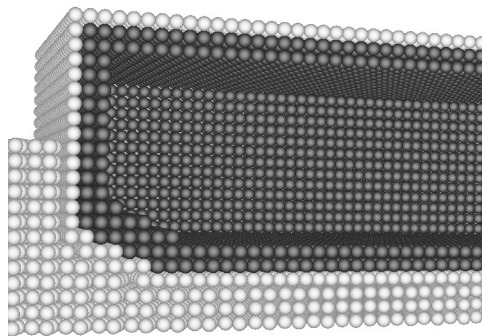


Figura 11. Vista em corte do *grid* de partículas da FPSO mostrando os detalhes da popa e inserido no domínio fluido

Por fim, o modelo completo de uma plataforma FPSO foi testado. A Fig. 12 mostra o modelo de sólido juntamente com a malha de triângulos gerados por um *software* de CAD. Um *close-up* no *grid* de partículas da região de convés, com a superestrutura e o *topside* caracterizado por uma grande quantidade de elementos é mostrado na Fig. 13. Da figura, é possível verificar que alguns elementos delgados, como os cilindros verticais, são representados por malhas de triângulos muito pequenos, bem menores que a distância entre partículas adotadas para o *grid*. Mesmo assim, externamente, as

partículas de parede foram geradas corretamente. No entanto, na vista em corte da Fig. 14 a seta aponta para as partículas internas ao casco que apresentaram problemas. Uma investigação mais a fundo sobre a causa do problema mostrou que este foi causado pelo erro em identificar se partícula é interna ou externa decorrente da sobreposição da superfície do casco com os fundos da superestrutura, Fig.15. O efeito da sobreposição de superfícies de sólidos foi avaliado usando um modelo composto por dois paralelepípedos: a Fig. 16.a mostra o erro na geração de partículas, causado pela sobreposição de duas superfícies, enquanto a Fig. 16.b mostra que com o afastamento das superfícies dos paralelepípedos o modelo de partículas é gerado com sucesso. Sendo assim, conclui-se que o problema é devido ao modelo de malha de triângulos, já que na prática os dois sólidos deveriam ser unidos e a operação de união deveria eliminar automaticamente as faces que estão sobrepostas.

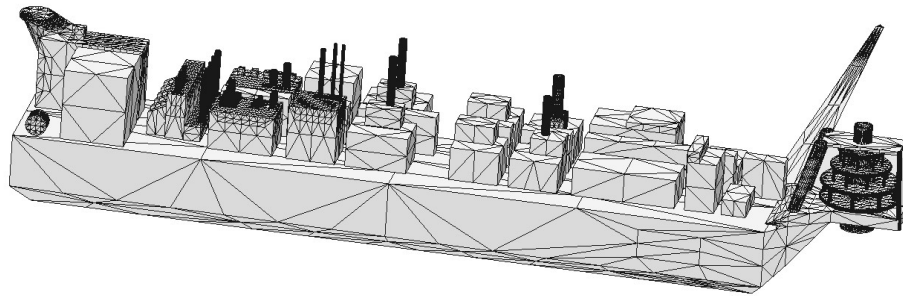


Figura 12. Modelo STL de uma plataforma do tipo FPSO completo

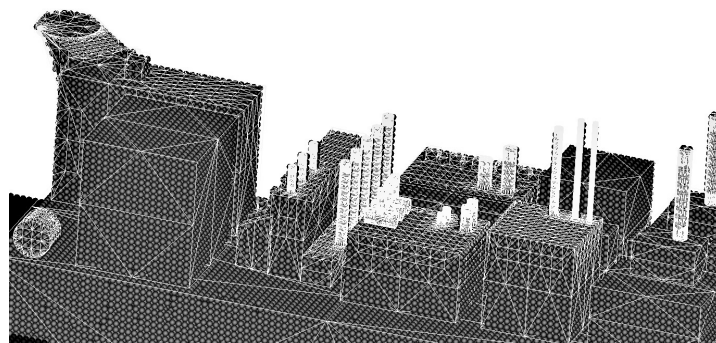


Figura 13. Detalhes do *grid* de partículas da plataforma do tipo FPSO completo

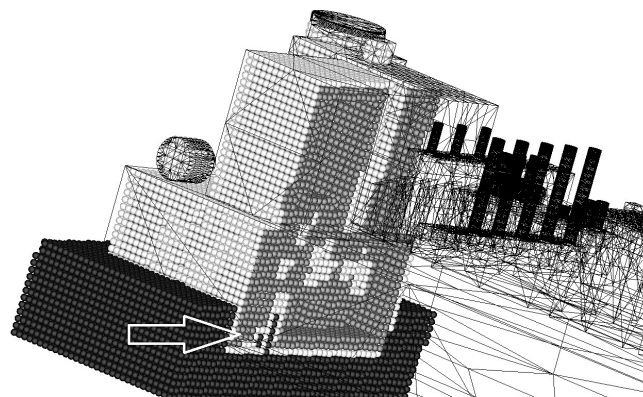


Figura 14. Vista em corte parcial do *Grid* mostrando os detalhes da popa

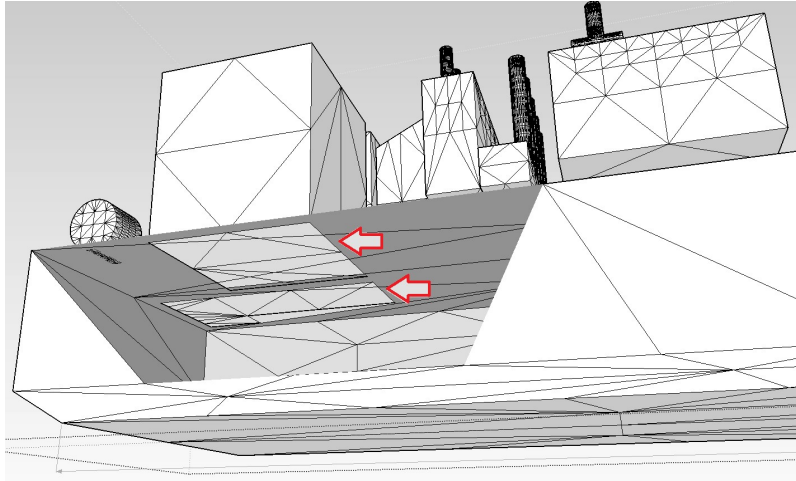


Figura 15. Vista em corte parcial da malha de triângulos; as setas apontam para a sobreposição da superfície do casco com os fundos dos containers

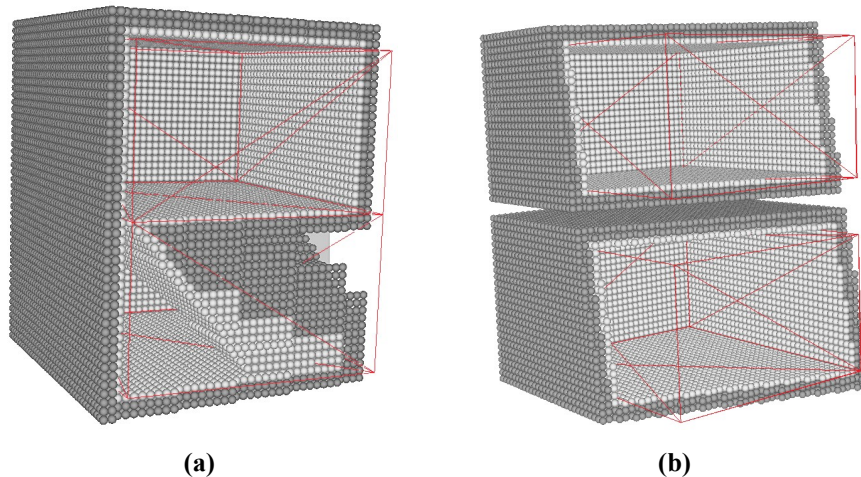


Figura 16. Vista em corte de modelos de partículas gerados a partir de malha de triângulos representando dois paralelepípedos: com sobreposição de superfícies (a) e sem sobreposição de superfícies (b).

6 CONSIDERAÇÕES FINAIS

Este trabalho apresenta as técnicas para pré-processamento do método de partículas *Moving Particle Simulation*, focando, nesta etapa, na geração do *grid* de partículas a partir de um modelo geométrico de sólido representado por uma malha de faces triangulares.

Os resultados obtidos com o caso mais simples de um poliedro até o caso mais complexo da geometria completa de uma plataforma FPSO validam os algoritmos desenvolvidos nesse trabalho, mostrando a confiabilidade e o potencial da técnica e a viabilidade da sua aplicação na modelagem e simulação de problemas reais de engenharia. Adicionalmente, o encaixe de um *grid* de objeto num *grid* de partículas de fluido foi testado.

Ao adotar o formato STL para dados de entrada, a técnica permite o uso dos modelos geométricos criados em *software* de CAD. Como resultado, conseguiu-se uma redução considerável no custo de geração de *grid* de partículas de sólidos de geometria complexa ou de difícil descrição matemática. Este trabalho também contribui para viabilizar o acoplamento entre o método MPS com outros métodos baseados em malha, que demanda a geração automática do *grid* de partículas do sólido.

Por fim, apesar da técnica de pré-processamento estar focada ao método *Moving Particle Simulation*, os fundamentos adotados no seu desenvolvimento também podem ser estendidos e aplicados aos métodos de partículas similares.

AGRADECIMENTOS

O autor Rodrigo Carneiro da Silva agradece o CNPq pela bolsa IC concedida para o desenvolvimento deste trabalho.

O autor Eric Henrique Favero agradece o Banco Santander pela bolsa IC concedida para o desenvolvimento do software de pré-processamento *Burner*.

REFERÊNCIAS BIBLIOGRÁFICAS

Akimoto, H., 2013. Numerical simulation of the flow around a planing body by MPS method. *Ocean Engineering*, v. 64, p. 72–79.

Amaro Junior, R. A., Cheng, L. Y., 2013. Brittle fracture and hydroelastic simulations based on moving particle simulation. *Computer Modeling in Engineering & Sciences*, v. 95, n. 2, p. 87–118.

Chen, R., Oka, Y., Li, G., Matsuura, T. 2014. Numerical investigation on melt freezing behavior in a tube by MPS method. *Nuclear Engineering and Design*, v. 273, p. 440–448.

Cheng, L. Y., Gomes, D. V., Nishimoto, K., 2011. 2D Simulation of leakage and damaged stability of oil carrier by MPS method. *Boletim Técnico da Petrobrás*, Vol. 54, No. 1.

Cheng, L. Y., Oliveira, L. H., Favero, E. H., 2012. Particle-based numerical analysis of drainage flow inside building system. *The Proceedings of the 38th International Symposium CIB W062 on Water Supply and Drainage for Buildings*. p.227 – 238.

Haines, E., "Point in polygon strategies" *Graphics Gems IV*, ed. Paul Heckbert, Academic Press, pp. 24-46, 1994.

Horvat, D., 2012. Ray-casting point-in-polyhedron test. *Proceedings of CESC2012: The 16th Central European Seminar on Computer Graphics*.

Koshizuka, S., 2011. Current achievements and future perspectives on particle simulation technologies for Fluid Dynamics and heat transfer. *Journal of Nuclear Science and Technology*, v. 48, n. 2, p. 155–168.

Koshizuka, S. and Oka, Y., 1996. Moving-particle semi-implicit method for fragmentation of incompressible fluid. *Nuclear Science and Engineering*, 123, pp.421-434

Koshizuka, S., Tamako, H., & Oka, Y., 1995. A particle method for incompressible viscous flow with fluid fragmentation. *Journal of Computational Fluid Dynamics*, pp. 29-46.

Pereira, D. S., Melo, P. S. S., Kurokawa, F. A., Cheng, L. Y., 2013. Uma modelagem numérica do escoamento do concreto fresco baseado no método Moving Particle Semi-implicit (MPS). *The Proceedings of the XXXIV Iberian Latin American Congress on Computational Methods in Engineering (CILAMCE2013)*.

Shakibaeinia, A., Jin, Y.-C., 2012. MPS mesh-free particle method for multiphase flows. *Computer Methods in Applied Mechanics and Engineering*, v. 229-232, p. 13–26.

Shibata, K. Koshizuka, S., Sakai, M., Tanizawa, K., 2012. Lagrangian simulations of ship-wave interactions in rough seas. *Ocean Engineering*, v. 42, p. 13–25.

Tsukamoto, M. M., 2006. Desenvolvimento do método de partículas na representação de corpos flutuantes em ondas altamente não-lineares. Dissertação de mestrado. EPUSP/São Paulo, Brasil.

Tsukamoto, M. M., Cheng, L.-Y., Nishimoto, K., 2011. Analytical and numerical study of the effects of an elastically-linked body on sloshing. *Computers & Fluids*, v. 49, n. 1, p. 1–21.