

## 6

### Implementação e Resultados

A implementação deste trabalho foi feita inteiramente em GPU. Os algoritmos descritos nas seções anteriores foram desenvolvidos e implementados de forma a otimizar os recursos paralelos existentes na placa de vídeo. O sistema foi programado usando a linguagem GLSL, fazendo uso de funções disponíveis no SDK de CUDA (17), como a função para ordenação e a função para acumular os valores de um vetor. A complexidade de funcionamento das placas de vídeo produzem resultados muito sensíveis à implementação dos algoritmos. Isto faz com que pequenas mudanças na estratégia adotada reflitam significativamente nos resultados. De uma forma geral, cada passo do procedimento descrito corresponde à execução de um *fragment shader*, usando texturas como vetores de dados. Os resultados serão apresentados separando a simulação de grãos apenas com forças normais e a simulação de grãos com forças normais e tangenciais, considerando a formulação do método dos elementos discretos que foi apresentada no Capítulo 3. Foram realizados testes de validação da implementação do modelo, de desempenho do sistema e de aplicação em exemplos de larga escala.

#### 6.1

##### Testes de Validação do Sistema

Para validação deste trabalho, foi utilizado o software conhecido como *PFC3D*, que é desenvolvido pela *Itasca Consulting Group, Inc.* (7). Sua primeira versão foi lançada em 1994 e ele tem por objetivo modelar o movimento e a interação de partículas esféricas através do Método dos Elementos Discretos (MED), como descrito por Cundall e Strack (3). O *PFC3D* é utilizado para simular várias aplicações em engenharia, como por exemplo:

- Fluxo de material em calhas, tubulações e reservatórios
- Mini escavações: fratura, ruptura, fragmentação e fluxo de blocos de rocha
- Compactação de pó em moldes
- Impacto de objetos compostos de partículas ligadas: a quebra dinâmica
- Estudos fundamentais de materiais granulares: rendimento, fluxo, alterações de volume, etc.

- Resposta sísmica e colapso de estruturas compostas por vigas representadas por matrizes de partículas coladas
- Estudos fundamentais de sólidos, representados por conjuntos ligados de partículas: acumulação de danos, emissão acústica e fratura

Os exemplos apresentados neste trabalho foram simulados na versão 3.0 do *PFC3D*, em um PC Intel Core 2 Duo @2.66GHz, com 2GB de memória RAM e sistema operacional Microsoft Windows XP Professional. O sistema desenvolvido neste trabalho, chamado de GPU Grãos, foi testado em um PC Intel Core 2 Duo @2.66GHz, com 2GB de memória RAM, sistema operacional Microsoft Windows XP Professional e placa de vídeo NVidia GeForce 9800 GT, com a versão de driver 178.24 e com memória dedicada de 512 MBytes e 112 processadores.

### 6.1.1

#### Colisão entre duas partículas

Para validar a implementação do modelo, o primeiro teste realizado foi a interação entre duas partículas sobre um plano, onde a única força de campo é a gravidade, como mostra a Figura 6.1. A partícula verde (ou cinza claro na versão preto e branco) foi posicionada em repouso sobre o plano, na coordenada (0.0, 0.0, -6.0). A partícula azul (ou preta na versão preto e branco), considerada a *partícula de teste*, foi liberada da coordenada (0.0, 0.0, 10.0), caindo em queda livre.

#### Simulação apenas com forças normais

A Tabela 6.1 mostra as propriedades das partículas e os dados utilizados na simulação apenas com forças normais.

Propriedades Físicas	Var.	Valor	Unid.
Raio	$R$	0.1	m
Massa Específica	$\rho$	1748.8	$kg/m^3$
Rigidez Normal	$K_n$	$3.636 \times 10^7$	$N/m$
Coef. de Amortecimento	$\alpha$	0.7	-
Passo de Tempo	$\Delta t$	$1.0 \times 10^{-5}$	s

Tabela 6.1: Propriedades das partículas e dados da simulação apenas com forças normais.

Foram coletadas a componente z do vetor posição e a componente z do vetor velocidade da partícula de teste na simulação feita pelo *PFC3D* e pelo GPU Grãos. As curvas de resposta dos dois simuladores podem ser comparadas na Figura 6.2.

No gráfico de posição, Figura 6.2a, a partícula de teste colide com a partícula em repouso aproximadamente na posição  $-5.70$  e volta a subir, retornando até

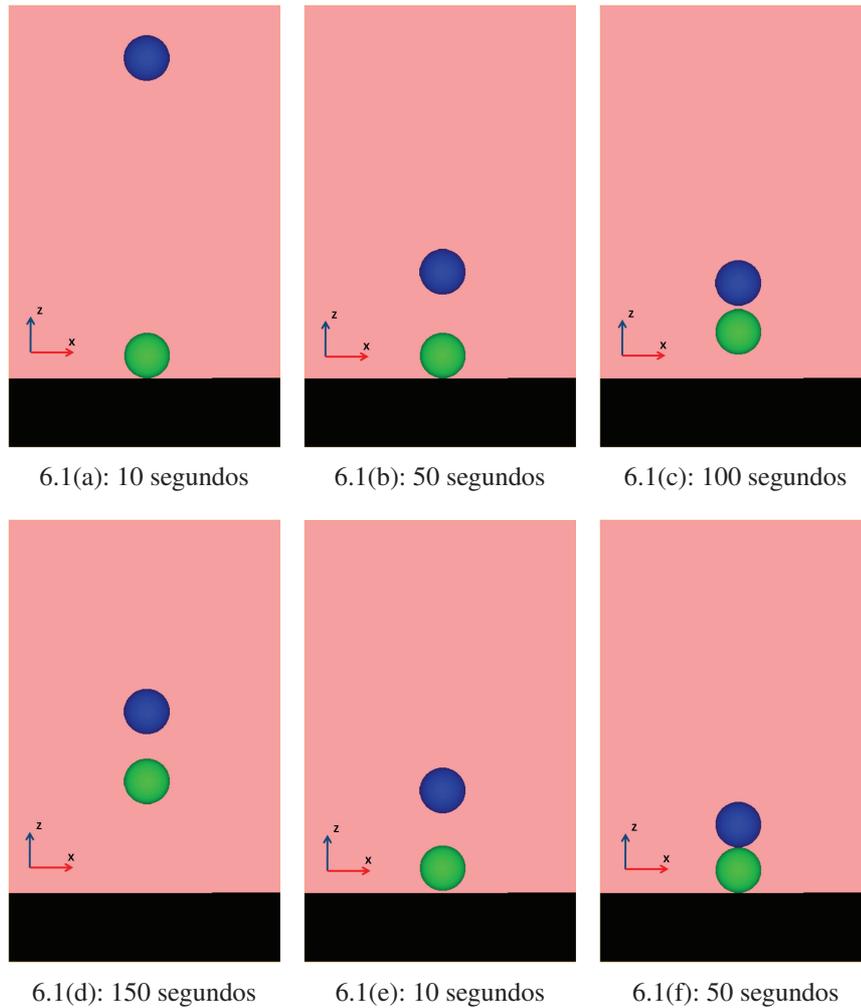
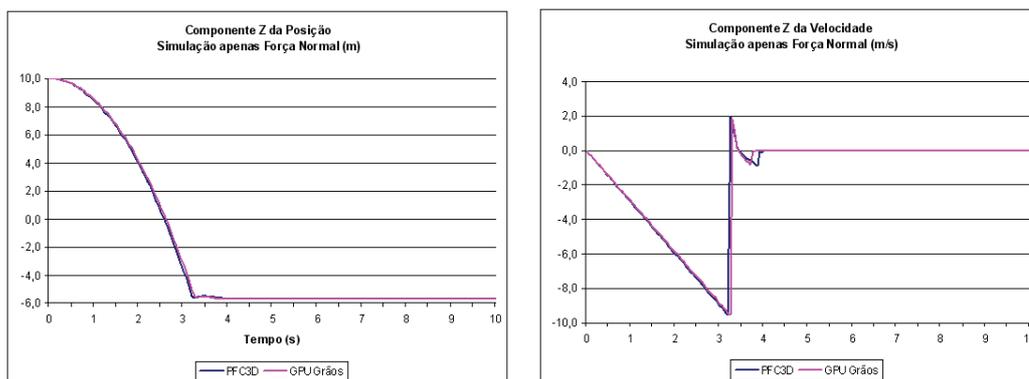


Figura 6.1: Simulação de colisão de 2 partículas sobre o plano



6.2(a): Componente z da posição da partícula de teste durante 10 segundos de simulação.

6.2(b): Componente z da velocidade da partícula de teste durante 10 segundos de simulação.

Figura 6.2: Resultado da partícula de teste na simulação apenas com forças normais.

próximo à posição  $-5.54$ , onde volta a cair, se acomodando sobre a outra partícula, com as duas permanecendo em repouso até o final da simulação. A diferença

entre as curvas dos dois simuladores é muito pequena. Isto mostra que o macro comportamento da implementação do MED na GPU está bem parecido com o comportamento do PFC3D.

Para a velocidade, o gráfico da Figura 6.2b mostra a queda livre da partícula de teste, descrevendo uma reta no sentido de  $z$  negativo. No momento da colisão acontece a inversão no sentido da velocidade, fazendo com que haja uma inversão no sentido de movimento e a partícula suba novamente, se afastando da partícula originalmente em repouso, até que elas colidem novamente e se acomodam uma sobre a outra e sobre o plano, levando a curva de velocidade para zero.

O ponto de máxima velocidade no sentido negativo de  $z$  e o ponto de máxima velocidade no sentido positivo de  $z$  são muito próximos no resultado dos dois simuladores, mostrando novamente a pequena diferença existente entre os macros comportamentos das duas implementações.

### Simulação com forças normais e tangenciais

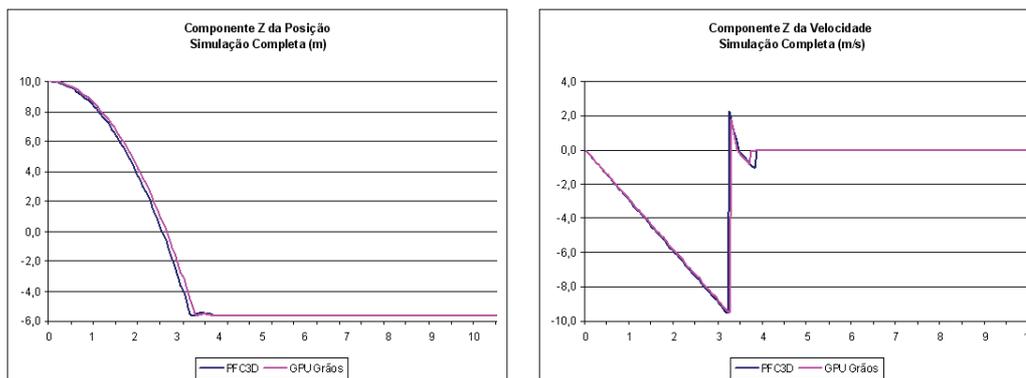
Considerando agora o modelo físico completo de grãos, são apresentados os resultados do mesmo teste de colisão apenas entre duas partículas e o plano. A Tabela 6.2 mostra as propriedades das partículas e os dados utilizados nesta simulação, incluindo dados específicos do modelo completo, que considera a força tangencial entre as partículas. Apesar das partículas estarem posicionadas com os seus centros colineares no eixo  $z$ , o que naturalmente não gera qualquer componente tangencial na força resultante, este teste de validação é necessário pois a implementação do sistema completo requer algoritmos muito diferentes daqueles utilizados pela simulação apenas com forças normais.

Propriedades Físicas	Var.	Valor	Unid.
Raio	$R$	0.1	m
Massa Específica	$\rho$	1748.8	$kg/m^3$
Rigidez Normal	$K_n$	$3.636 \times 10^7$	$N/m$
Rigidez Tangencial	$K_t$	$3.636 \times 10^7$	$N/m$
Coef. de Amortecimento	$\alpha$	0.7	-
Coef. de Atrito	$\mu$	0.5	-
Passo de Tempo	$\Delta t$	$1.0 \times 10^{-5}$	s

Tabela 6.2: Propriedades das partículas e dados da simulação com o modelo físico completo.

A Figura 6.3a, mostra as duas curvas dos dois simuladores, *PFC3D* e GPU Grãos, com a componente  $z$  da posição da partícula de teste. Novamente a diferença existente entre as curvas é muito pequena, demonstrando a semelhança do comportamento nas duas implementações. O mesmo arco de retorno da partícula de teste

é observado para a simulação com o modelo físico completo. O gráfico de velocidade, na Figura 6.3b, também mostra que os pontos de colisão e de máxima velocidade na mudança de sentido são bem próximos nos dois simuladores. Novamente, após a colisão, as duas partículas sobem até uma determinada altura e depois se acomodam, uma sobre a outra, sobre o plano. Os pontos de mínimo e de máximo no resultado de velocidade para a simulação com forças normais e tangenciais também são bem próximos do resultado com apenas forças normais, validando as duas implementações feitas no GPU Grãos.



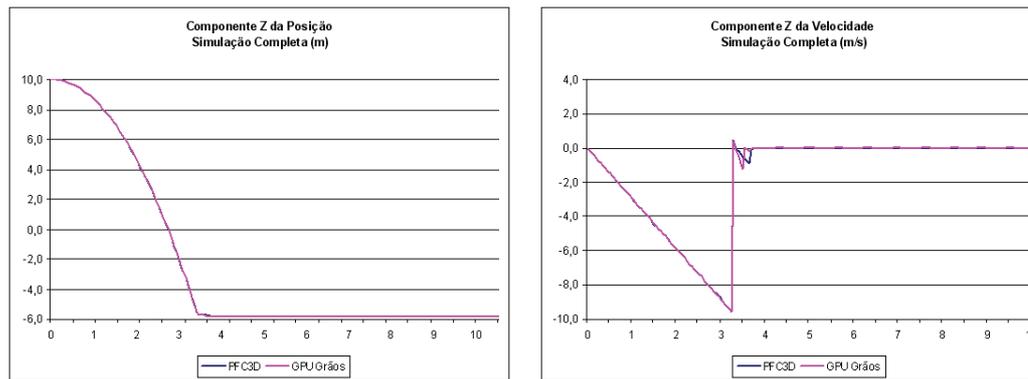
6.3(a): Componente z da posição da partícula de teste com o modelo físico completo. 6.3(b): Componente z da velocidade da partícula de teste com o modelo físico completo.

Figura 6.3: Simulação com o modelo físico completo de 2 partículas sobre o plano.

Para verificar e validar a interação tangencial entre as partículas, foi realizado o experimento anterior com apenas duas partículas, porém com o centro da partícula de teste deslocado. Uma partícula permanece em repouso sobre o plano, enquanto que a partícula de teste é liberada, em queda livre, a partir da posição (0.1, 0.1, 10.0). Este experimento foi realizado no *PFC3D* e no GPU Grãos. A comparação entre os resultados dos dois simuladores pode ser observada na Figura 6.4.

O gráfico de posição mostra o mesmo deslocamento da partícula de teste nos dois simuladores, *PFC3D* e GPU Grãos. Os dois apresentam como resultado um pequeno retorno da partícula de teste, após o impacto com a partícula em repouso. No gráfico de velocidade, o valor da máxima velocidade no sentido negativo de z, um pouco antes do impacto, é igual para as duas simulações. Ocorre uma inversão no sentido da velocidade, com um pico positivo em  $0.4m/s$  nos dois simuladores. No momento do retorno da partícula de teste para acomodação sobre o plano, ocorre uma pequena diferença entre os resultados, com o GPU Grãos tendo o segundo impacto um pouco antes do que o *PFC3D*.

Analisando estes resultados, foi possível avaliar que a implementação do GPU Grãos está correta. Pequenas diferenças apresentadas nos testes podem ser



6.4(a): Componente z da posição da partícula de teste deslocada. 6.4(b): Componente z da velocidade da partícula de teste deslocada.

Figura 6.4: Validação do modelo físico completo para duas partículas com colisão tangencial.

creditadas à diferença de precisão na aritmética de ponto flutuante existente entre a CPU e a GPU.

## 6.2 Testes de Desempenho do Sistema

Um dos objetivos deste trabalho é acelerar o processo de simulação numérica de um sistema de partículas, utilizando o hardware gráfico. O trabalho realizado por Venetillo e Celes (23) mostrou a grande capacidade de processamento das GPUs, quando obteve um desempenho de 6 fps com 1024K partículas.

Esta seção apresenta a comparação de desempenho deste trabalho com trabalhos passados, como o de Venetillo e Celes (23), assim como o desempenho do simulador comercial *PFC3D*. O objetivo desta comparação é verificar se os algoritmos de construção da grade uniforme e de construção de contatos, propostos neste trabalho, obtiveram algum avanço em relação aos já existentes.

### 6.2.1 Comparação de Desempenho com Trabalhos Existentes

O trabalho desenvolvido por Venetillo e Celes (23) considera apenas a interação entre as partículas, reposicionando-as a cada interpenetração. Venetillo utiliza a integração numérica de Verlet (5), onde a nova posição da partícula é calculada com base nas duas posições anteriores, sem a necessidade de calcular e armazenar explicitamente a velocidade. O método de construção da grade, adotado por Venetillo e Celes (23), também é diferente do proposto neste trabalho. Sua estratégia utiliza uma estrutura com o tamanho do número de células existentes no domínio da simulação mais o número de partículas que estão sendo simuladas.

Assim, o algoritmo de Venetillo e Celes (23) realiza várias passadas no pipeline gráfico, construindo uma lista encadeada das partículas existentes em cada célula.

A proposta de construção da grade deste trabalho, como apresentada na Seção 5.1.1, realiza apenas uma passada pelo pipeline gráfico para montar os pares  $[célula\ id, partícula\ id]$ . Depois, uma ferramenta conhecida como *CUDPP* (20), desenvolvida em CUDA (17), é utilizada para ordenar esses pares quanto ao id da célula, e por último, é construída uma nova estrutura de endereçamento para consulta nesta grade.

A implementação de Venetillo e Celes (23) e o simulador GPU Grãos foram testados sob as mesmas condições de número de partículas, tamanho do domínio no espaço, raio fixo das partículas, mesmo tamanho de célula, mesmo número de células na grade e mesmo *hardware* gráfico. O experimento realizado é a queda em superfície livre, com o estado inicial apresentado pela Figura 6.5. A implementação de Venetillo foi testada com apenas uma interação de relaxação e mesmo  $\Delta t$ . Os tempos de cada frame, em milissegundo, das duas simulações podem ser observados na Figura 6.6.

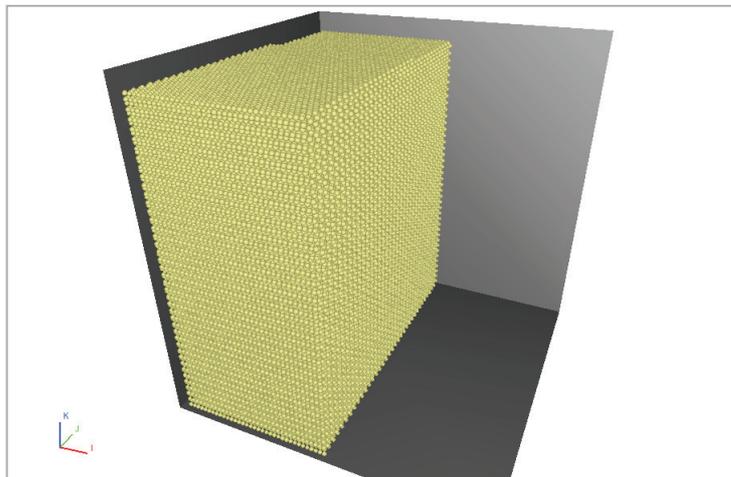


Figura 6.5: Simulação de superfície livre.

O gráfico da Figura 6.6 mostra que o algoritmo de construção da grade proposto neste trabalho conseguiu ser eficaz para modelos com um grande número de partículas. A partir de 200K partículas, o GPU Grãos já começa a ter um desempenho muito superior ao do método adotado por Venetillo e Celes (23). A principal diferença entre as duas implementações de simulação de partículas é a construção da grade uniforme. O resultado do GPU Grãos mostra que o novo algoritmo de construção da grade uniforme contribuiu muito para o melhor desempenho de todo o sistema.

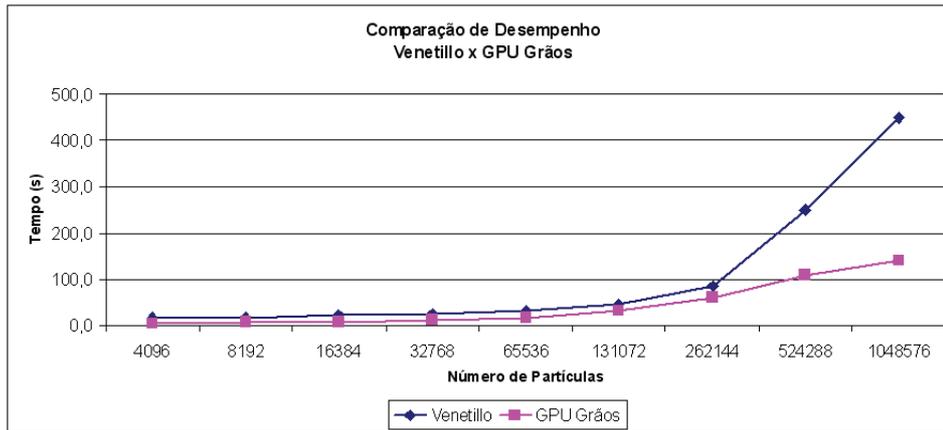


Figura 6.6: Tempo de cada frame das simulações de Venetillo e Celes e do GPU Grãos.

### 6.2.2

#### Comparaç o de Desempenho com o *PFC3D*

Para analisar o desempenho do GPU Gr os, foram feitos experimentos com 150K part culas confinadas em uma caixa, simuladas no *PFC3D*. Um part cula de teste foi liberada para cair em queda livre sobre as part culas adensadas. O adensamento foi estipulado considerando como par metro a energia cin tica das part culas, para o caso do simulador GPU Gr os. No simulador *PFC3D*, o crit rio   a m dia das forç s desbalanceadas. Foram feitos experimentos com a part cula de teste possuindo a metade do raio e o peso sendo de 1.25, 10.0 e 100.0 vezes o peso das outras part culas. A Figura 6.7 mostra o cen rio dos experimentos rodados no simulador GPU Gr os e no *PFC3D*.

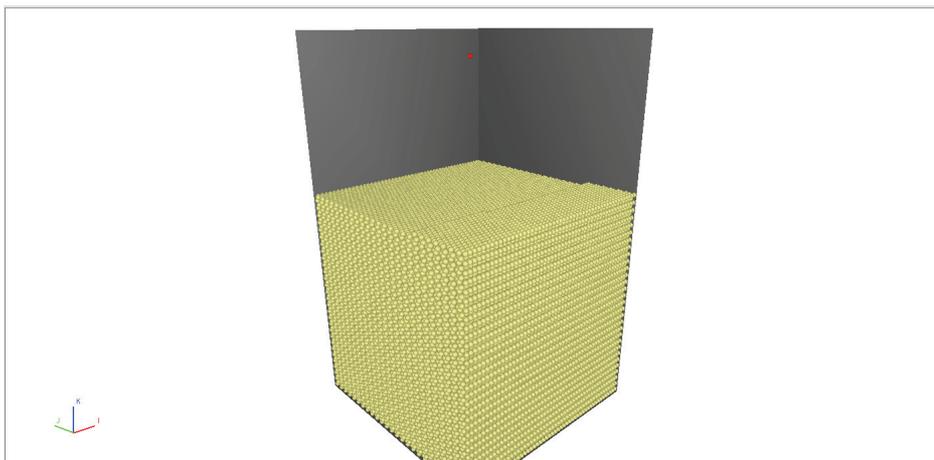


Figura 6.7: Simulaç o de uma part cula de teste em queda livre sobre 150K part culas acomodadas.

A formaç o inicial das part culas que representam o meio   feita para construir uma colmeia. Cada part cula   posicionada de forma muito pr xima das outras,

porém sem contato prévio. A Figura 6.8 mostra com mais detalhe a formação inicial. Tanto do lado direito da figura, olhando para a primeira fileira por exemplo, quanto do lado esquerdo no alto, é possível observar o posicionamento livre de contatos de cada partícula.

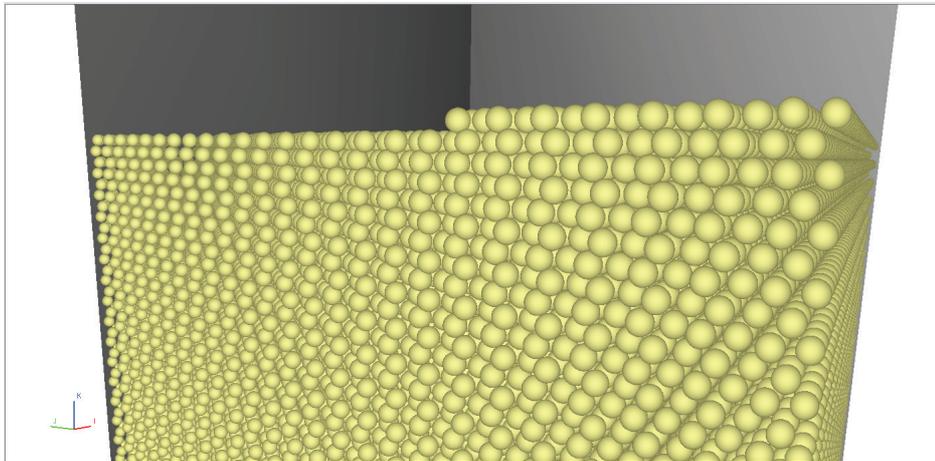


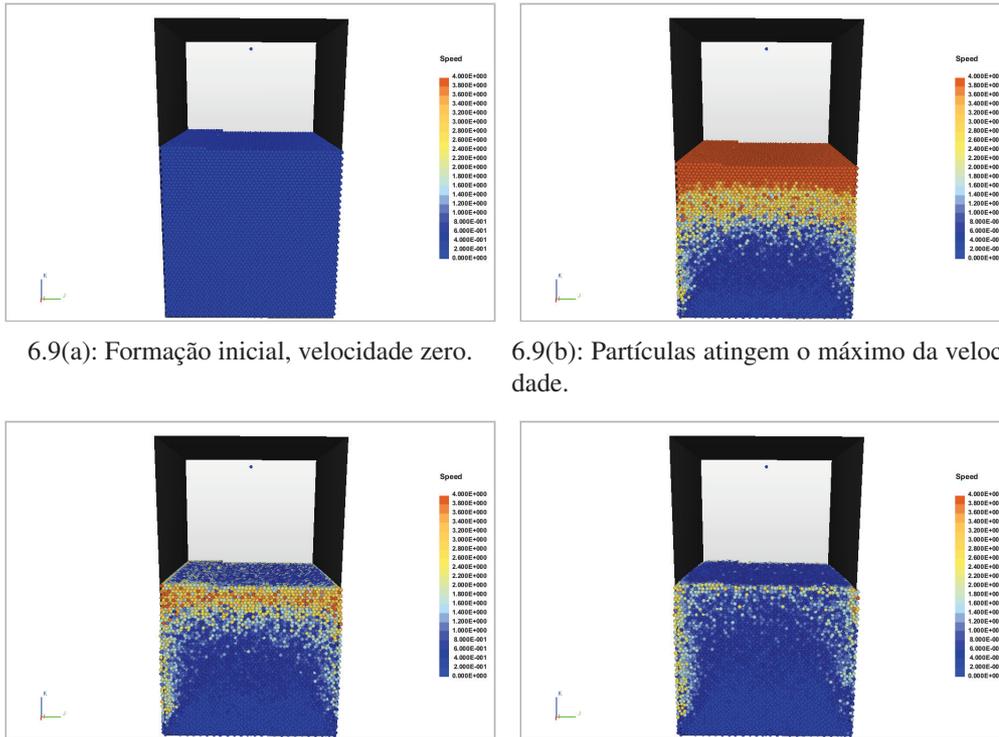
Figura 6.8: Formação inicial em formato de colmeia com 150K partículas.

A partir da formação inicial, a simulação se inicia com o efeito da gravidade, e as partículas começam a interagir entre si, se acomodando e preenchendo os espaços vazios. Como nas paredes da caixa foram deixados espaços para as partículas, é possível observar o escoamento pelas bordas da caixa, enquanto que o centro se acomoda com velocidade próxima de zero, conforme mostra a sequência da Figura 6.9, com os resultados do simulador GPU Grãos.

A acomodação chega ao fim quando o valor máximo da energia cinética de um determinado passo de integração for de 1% do máximo valor da energia cinética de toda a simulação. Atingido este critério, a partícula de teste é liberada e ganha velocidade durante a queda livre para atingir o meio adensado. A seguir são apresentados os resultados comparativos da partícula de teste com 1.25, 10.0 e 100.0 vezes o peso das outras.

A Figura 6.10 mostra a comparação da componente  $z$  da velocidade da partícula de teste variando o seu peso, nos dois simuladores, apenas com forças normais. Para o simulador GPU Grãos, o valor máximo da velocidade logo após o impacto diminui conforme aumenta o peso da partícula de teste. Para o simulador *PFC3D*, os pesos de 1.25 e 10.0 vezes o peso das demais partículas não resultam em grandes alterações. Na Figura 6.10b, o ponto máximo da velocidade no *PFC3D* é próximo do GPU Grãos. Porém, no caso de 100.0 vezes o peso (Figura 6.10) os pontos já são mais distantes, com o GPU Grãos ficando abaixo de zero.

A diferença entre as curvas dos simuladores pode ser explicada pela aleatoriedade da acomodação das partículas. O meio ao qual as partículas de teste estão se chocando pode ser bastante diferente. Em um dos casos, a partícula de teste pode



6.9(a): Formação inicial, velocidade zero.

6.9(b): Partículas atingem o máximo da velocidade.

6.9(c): Partículas do centro começam a se acodar.

6.9(d): Apenas algumas partículas nas bordas ainda se movimentam.

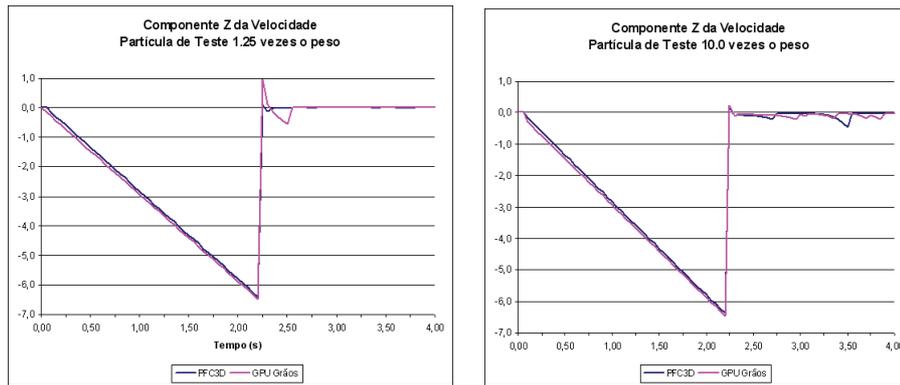
Figura 6.9: Variação de velocidade durante acomodação de 150K partículas.

estar colidindo diretamente com uma outra partícula, enquanto que em outro caso pode estar entrando no meio através de um espaço vazio, e colidindo apenas de forma parcial com as demais. No entanto, o macro comportamento é similar nos dois simuladores.

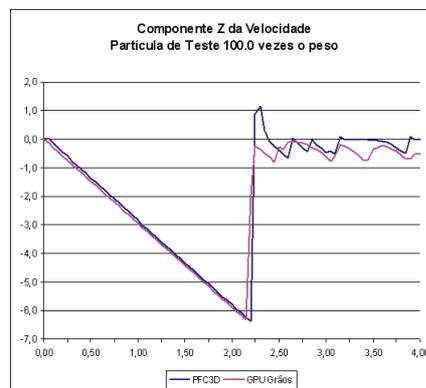
O objetivo principal deste experimento é avaliar o tempo que cada simulador gasta para fazer uma simulação de 4 segundos com 150K partículas, após a acomodação das partículas no ambiente. A Tabela 6.3 mostra o tempo gasto por cada simulador com apenas força normal e com o modelo físico completo, além do ganho de desempenho obtido.

Simulação	PFC3D	Grãos GPU	Ganho
Apenas Força Normal	162.92 min	15.00 min	10.86x
Modelo Completo	307.00 min	43.00 min	7.14x

Tabela 6.3: Tempo de cada simulador para calcular 4s de simulação com 150K partículas.



6.10(a): Partícula de teste com 1,25 vezes o peso das demais  
6.10(b): Partícula de teste com 10,0 vezes o peso das demais



6.10(c): Partícula de teste com 100,0 vezes o peso das demais

Figura 6.10: Resultado da partícula de teste caindo sobre 150k partículas acomodadas.

### 6.3 Teste de Variação de Raio

Para verificar o desempenho do uso da grade uniforme para o caso de variação de raio, o número de partículas se manteve fixo em 150K, em um domínio também fixo com o tamanho da célula sempre igual ao dobro do maior raio entre as partículas. O raio máximo variou até o dobro do valor do raio mínimo, seguindo uma variação uniforme. A Figura 6.11 mostra o resultado desta variação para os dois algoritmos de construção da grade apresentados no Capítulo 4, simulando apenas com forças normais.

O algoritmo que armazena a partícula apenas na célula do seu centro geométrico, *Grade 1-Partícula:1-Célula*, é mais rápido do que a *Grade 1-Partícula:N-Células*. Porém, aumentando o raio máximo até 1.6 do raio mínimo, a simulação ficou 53% mais lenta, além de não ter conseguido viabilizar a simulação para variações de raio maiores do que esta.

O algoritmo de *Grade 1-Partícula:N-Células*, onde cada partícula é ar-

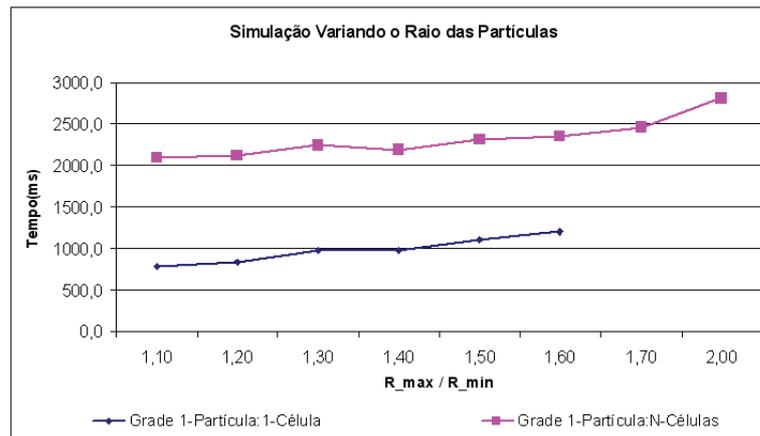


Figura 6.11: Simulação com variação de raio de 150K partículas.

mazenada em todas as células que ocupa, é mais lento em valores absolutos. Porém, foi possível aumentar o raio máximo até o dobro do raio mínimo, perdendo apenas 35% de desempenho. Mesmo sendo menos eficiente para simulação física, este algoritmo mostra uma nova abordagem que pode ser um bom caminho para tratar elementos com dimensões de diferentes ordens de grandeza.

## 6.4

### Exemplo Superfície Livre

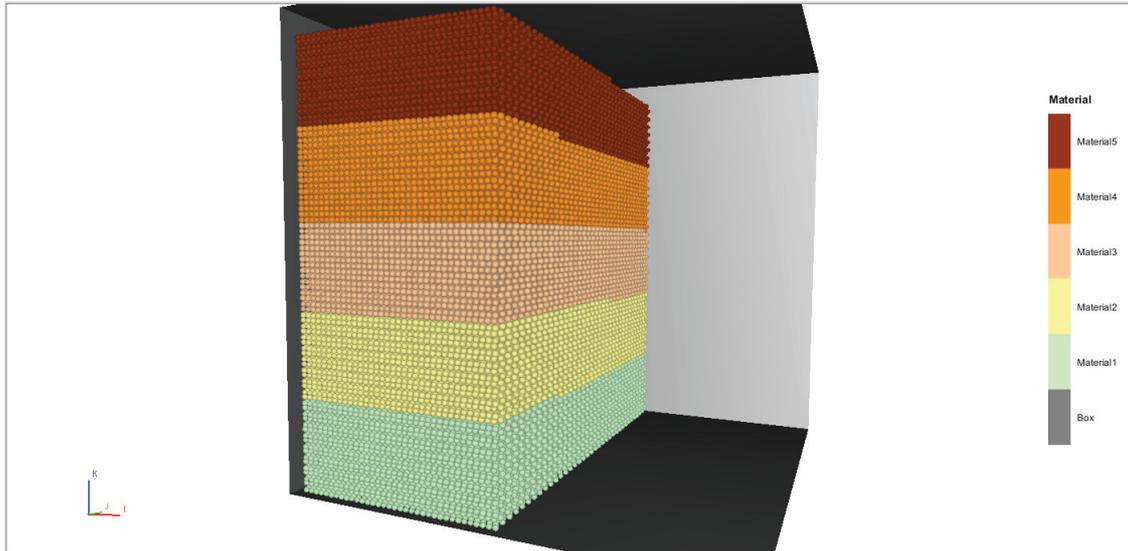
Para exemplificar o comportamento do GPU Grãos, a formação original em colmeia com uma das superfícies livres foi simulada com 500K partículas, de raio e materiais diferentes. A massa específica das partículas foi dividida em 5 camadas, com a camada mais acima sendo a de maior densidade. O raio das partículas varia entre 0.08 e 0.10. A Tabela 6.4 mostra os dados utilizados na simulação de superfície livre.

Propriedades Físicas	Var.	Valor	Unid.
Raio	$R$	0.08 - 0.10	m
Massa Específica	$\rho$	100.0/250.0/500.0/750.0/1000.0	$kg/m^3$
Rigidez Normal	$K_n$	$1.0 \times 10^6$	$N/m$
Rigidez Tangencial	$K_t$	$1.0 \times 10^6$	$N/m$
Coef. de Amortecimento	$\alpha$	0.7	-
Coef. de Atrito	$\mu$	0.5	-
Passo de Tempo	$\Delta t$	$2.0 \times 10^{-5}$	s

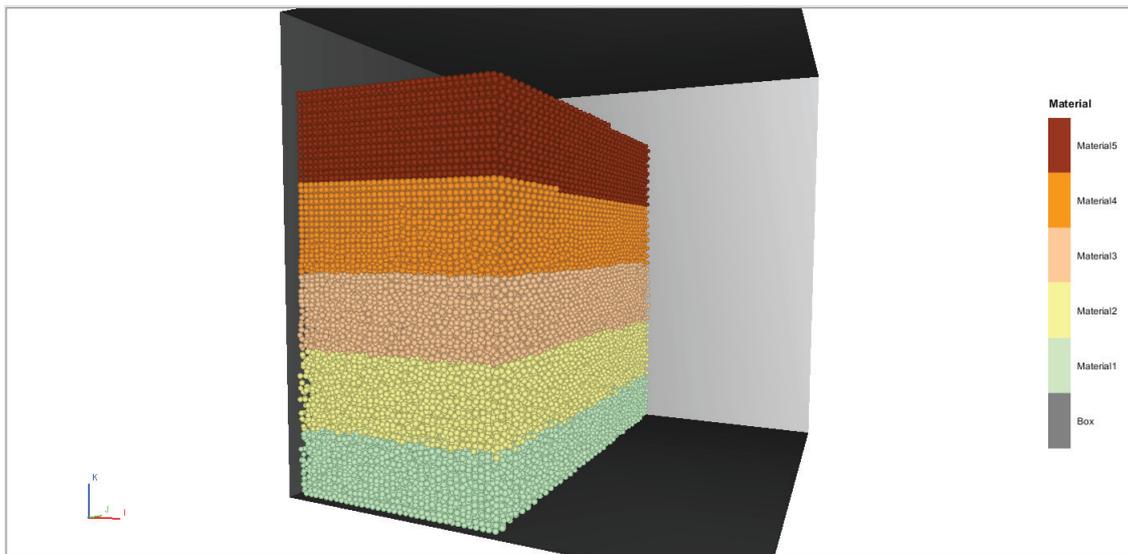
Tabela 6.4: Propriedades das partículas e dados da simulação de superfície livre.

A sequência de imagens nas Figuras 6.12, 6.13 e 6.14 mostra a acomodação das partículas. A força de atrito e o elevado coeficiente de amortecimento fazem com que as partículas deslizem de forma suave, como uma erosão, sem um grande

espalhamento sobre o plano. A camada mais acima, por ser mais densa, envolve pelas laterais as camadas inferiores. A formação final mostra uma inclinação parecida com um monte de areia, com partículas acomodadas ao centro, porém com deslizamentos nas áreas livres.

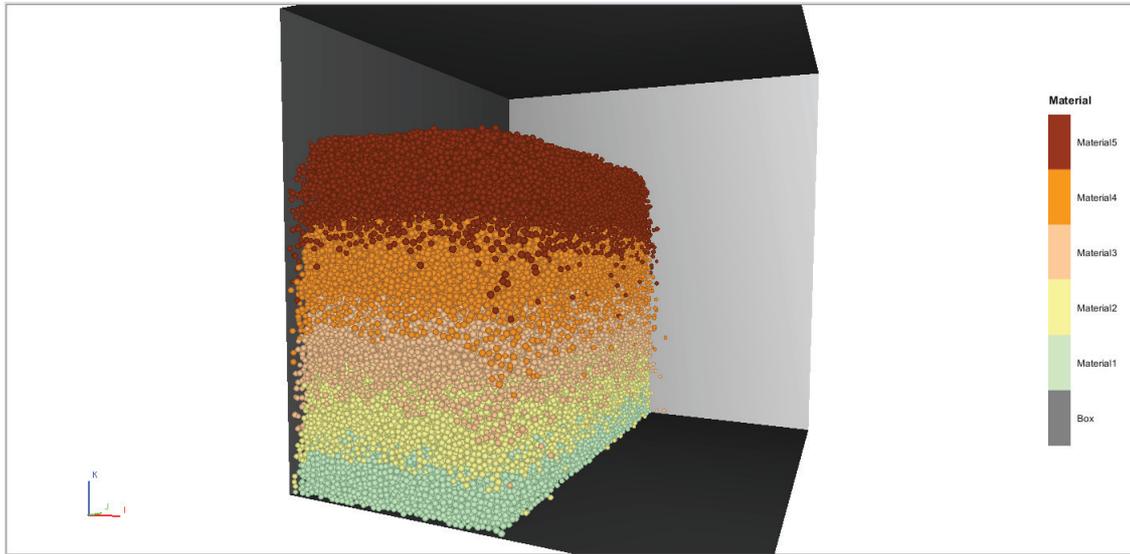


6.12(a): Formação inicial

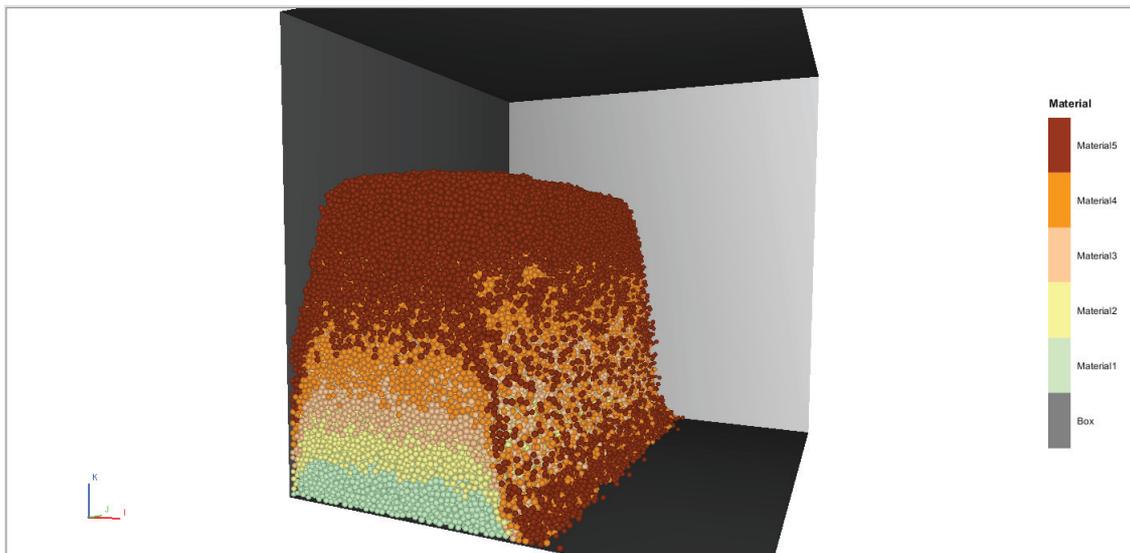


6.12(b): Algumas camadas já em colisão

Figura 6.12: Simulação de superfície livre com camadas de diferentes materiais.

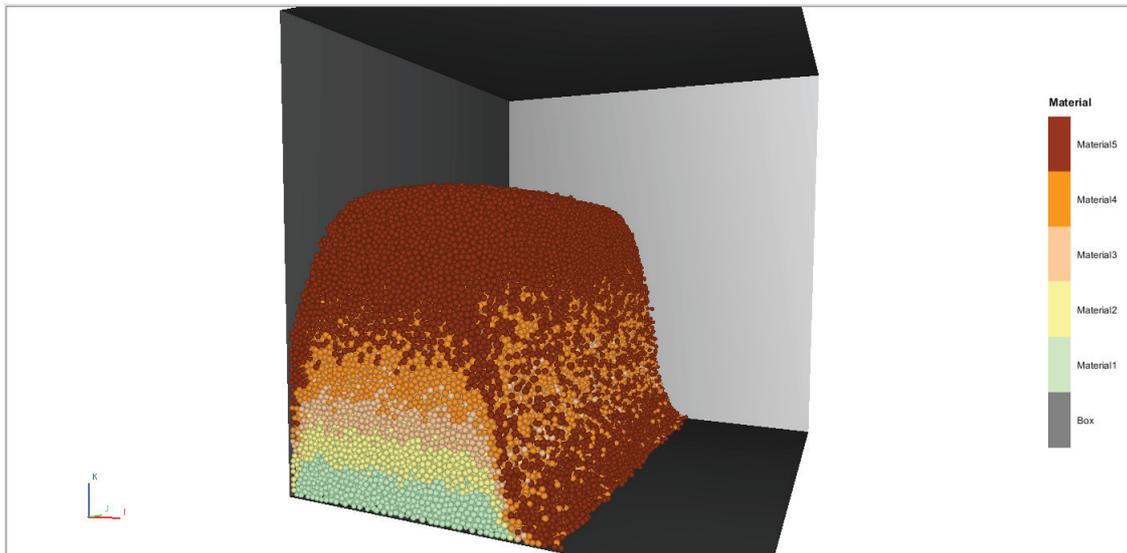


6.13(a): Deslizamento pelas laterais

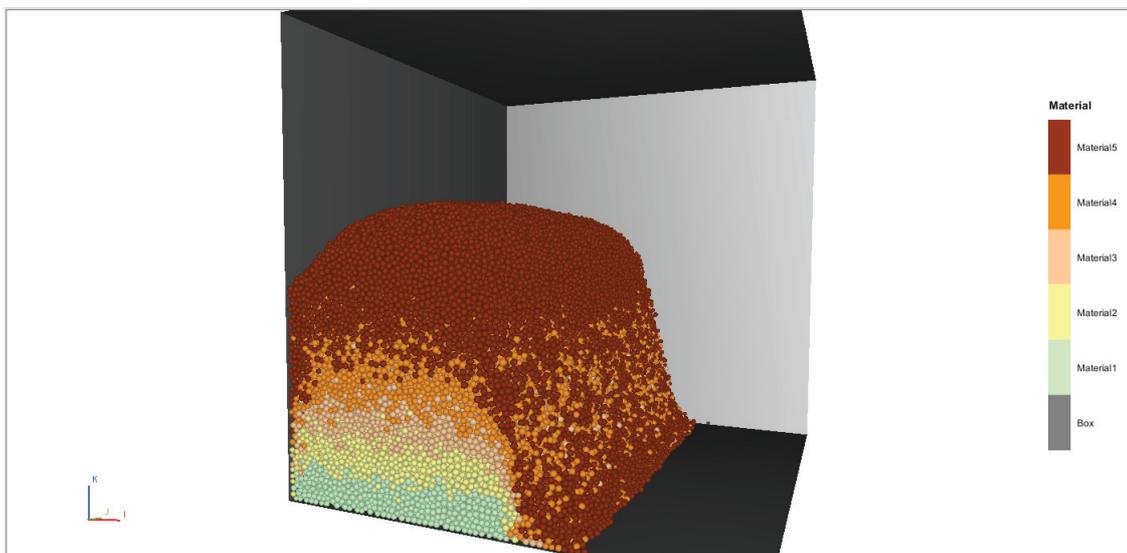


6.13(b): Deslizamento pela superfície livre

Figura 6.13: Simulação de superfície livre com camadas de diferentes materiais.



6.14(a): Penetração da camada mais densa nas camadas inferiores



6.14(b): Acomodação como um monte de areia

Figura 6.14: Simulação de superfície livre com camadas de diferentes materiais.

## 6.5

**Exemplo Ângulo de Resposta**

Outra aplicação da simulação de Grãos é a realização de um experimento com partículas posicionadas dentro de um funil e caindo por uma abertura na parte inferior, como mostra a Figura 6.15.

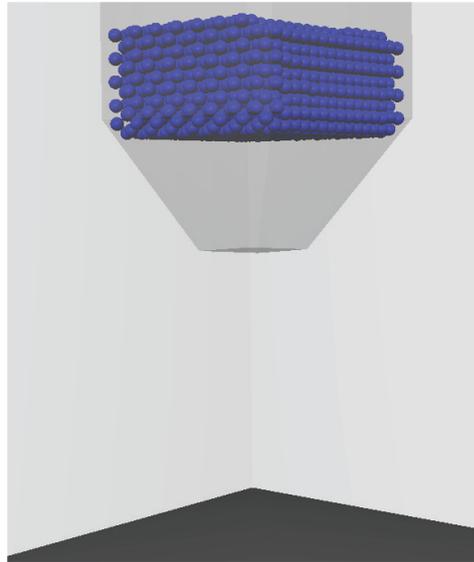


Figura 6.15: Simulação de 2000 partículas caindo por um funil.

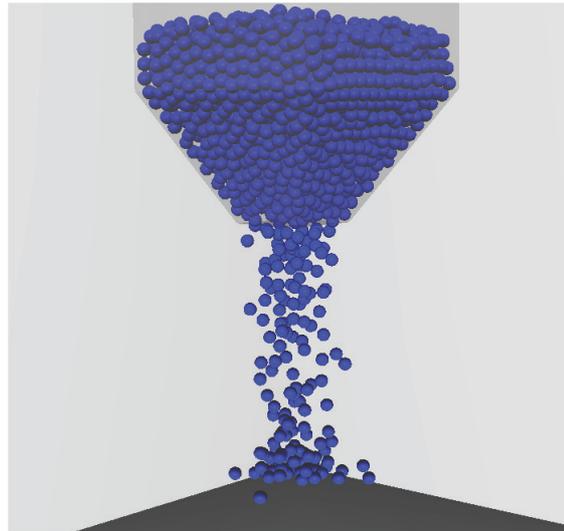
Foram simuladas 2000 partículas com as características apresentadas na Tabela 6.5.

Propriedades Físicas	Var.	Valor	Unid.
Raio	$R$	0.25	m
Massa Específica	$\rho$	1000.0	$kg/m^3$
Rigidez Normal	$K_n$	$1.0 \times 10^8$	$N/m$
Rigidez Tangencial	$K_t$	$1.0 \times 10^8$	$N/m$
Coef. de Amortecimento	$\alpha$	0.7	-
Coef. de Atrito	$\mu$	0.577	-
Passo de Tempo	$\Delta t$	$7.0 \times 10^{-5}$	s

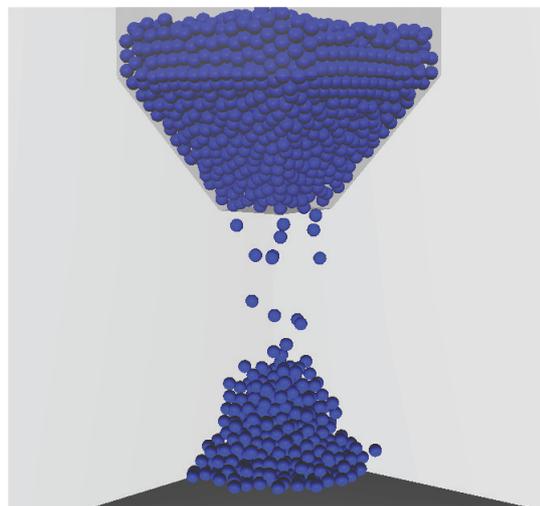
Tabela 6.5: Propriedades das partículas e dados da simulação de ângulo de resposta.

A Figura 6.16 mostra a queda das partículas pelo funil, com a pilha sendo formada sobre o plano. O atrito entre as partículas faz com que não haja espalhamento, formando um ângulo de inclinação na configuração final de acomodação.

As partículas localizadas no centro do funil sofre imediata queda, enquanto que as partículas em contato com as paredes inclinadas caem lentamente por causa do atrito com o plano. A Figura 6.17 mostra o detalhe das partículas em contato com o plano, e o centro do funil vazio.



6.16(a): Queda das partículas pelo funil



6.16(b): Acomodação formando uma pilha, com ângulo de inclinação.

Figura 6.16: Simulação de pilha com partículas passando por um funil.

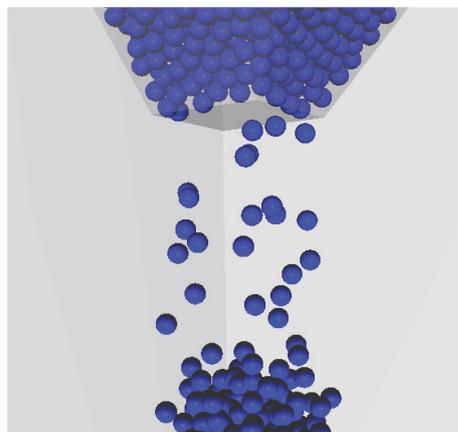


Figura 6.17: Partículas caindo lentamente por causa do atrito com as paredes do funil.