

8 Implementação do método GBPM

Antes de descrever o algoritmo completo, é interessante formar uma visão geral do problema. Por isso, vamos começar com um esboço de implementação para o método GBPM.

8.1 Esboço do algoritmo

A estrutura do algoritmo é resumida nas etapas exibidas na figura (8.1).

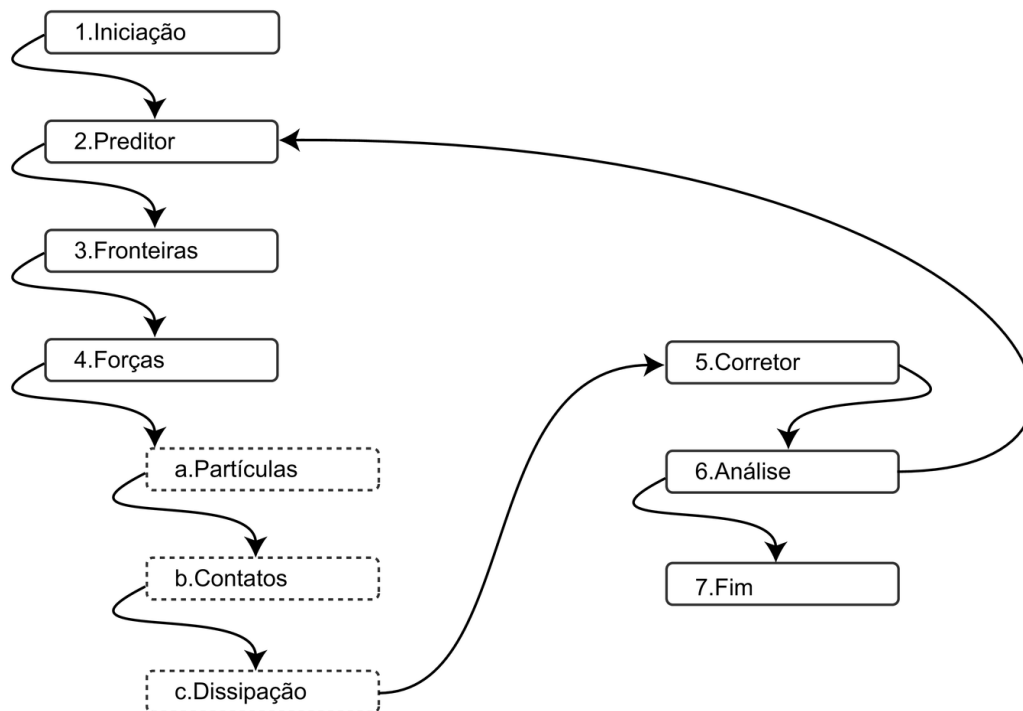


Figura 8.1: Esboço do algoritmo.

1. Iniciação: São definidas as posições e velocidades iniciais das partículas, e também a rede de contatos. Na seção 6.1 é descrito o processo utilizado na construção da geometria do modelo inicial.
2. Preditor: É efetuado o cálculo das aproximações para as coordenadas, e derivadas, em relação ao tempo $t + \Delta t$. Para gerar essas aproximações,

empregamos a expansão em série de Taylor. A seção 7.1 explica esse processo.

3. Fronteiras: Responsável pela movimentação das partículas de fronteira, essas partículas seguem uma trajetória pré-estabelecida. Ver seção 2.3.
4. Forças: Esse é o ponto mais importante do algoritmo. As forças são calculadas, e acumuladas, para na seqüência, serem utilizadas na integração das equações de Newton. Vamos subdividir essa etapa em três passos. E o efeito de forças externas, como a gravidade por exemplo, pode ser adicionada a simulação durante esse passo. Para conseguir isso, basta iterar sobre a lista de partículas e adicionar a força desejada.
 - (a) Partículas: É efetuado o cálculo das forças de interação entre partículas, assim como os torques. É recomendável o auxílio de uma estrutura de dados de busca espacial, para se determinar rapidamente as partículas que estão em contato mecânico. Uma opção para a estrutura de dados é apresentada na seção 7.2. E diferentes modelos de contato podem ser utilizados. A seção 3 mostra os possíveis modelos de contato.
 - (b) Contatos: Determinação das forças devido ao modelo de contato cimentado. Essa etapa é numericamente eficiente, de fato, para determinar essas forças é necessário percorrer a lista que contém a informação dos contatos, calculando e acumulando as forças e torques em cada partícula. Na seção 5.2 encontram-se as fórmulas que modelam as forças e os torques. É também nessa etapa, que podem ocorrer o rompimento dos contatos. O critério para a quebra do contato está descrito na seção 5.3.
 - (c) Dissipação: São calculadas as forças relacionadas a dissipação de energia cinética. Na seção 3.4 é apresentado o modelo de dissipação.
5. Corretor: São utilizadas as forças calculadas na etapa anterior para corrigir as acelerações dos grãos. Veja a seção 7.1.
6. Análise: Logo após o contador de tempo ser incrementado em um passo de tempo $t = t + \Delta t$, são salvas informações relevantes da simulação, e um critério de parada da simulação é testado. Um exemplo de critério de parada é verificar se o tempo de chegou a certo limiar, t_{\max} .
7. Fim: É efetuada a limpeza na memória, e em seguida a finalização do programa.

8.2

Estruturação do código

O design definido para a implementação a seguir, está focado no entendimento e na reutilização de código. No digrama 8.2 são exibidas as estruturas lógicas básicas do programa. E com o auxílio dessas estruturas, o modelo da simulação é organizado na memória do computador.

Em seguida, cada uma dessas estruturas será descrita.

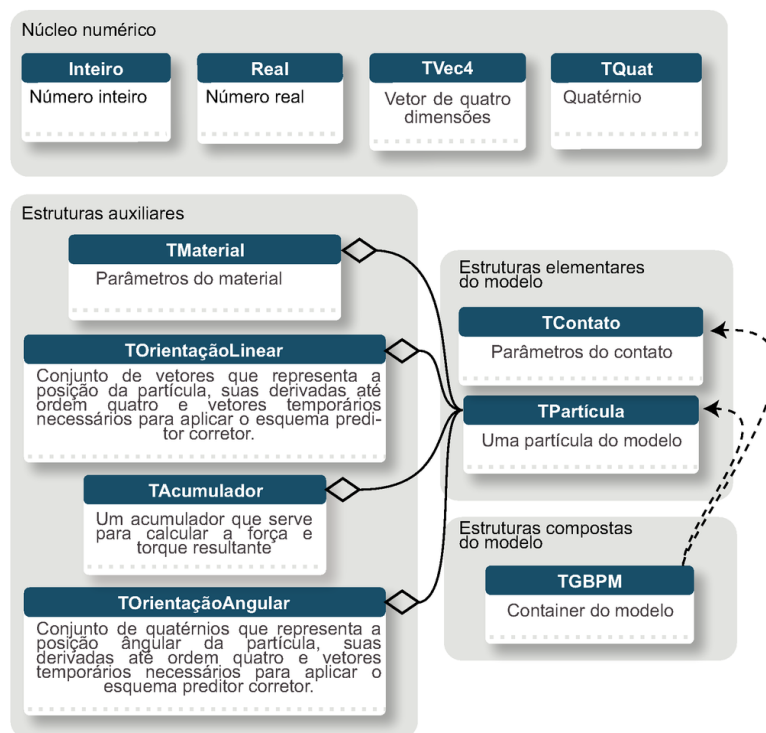


Figura 8.2: Estruturas do programa, o núcleo numérico é encarregado das operações básicas. As estruturas auxiliares são definidas para otimizar o reuso do código, podendo ser substituídas mais tarde. E as estruturas do modelo, que são os elementos que definem a simulação.

8.2.1

Núcleo numérico

O núcleo numérico é o responsável sobre os cálculos aritméticos. Esse núcleo é capaz de realizar diversas operações nos seguintes conjuntos: quatérnios \mathbb{Q} , vetores no \mathbb{R}^4 , Reais \mathbb{R} , e Inteiros \mathbb{Z} . Em especial esse módulo consegue calcular as rotações definidas pelos quatérnios.

A escolha do \mathbb{R}^4 é natural, pois é com o auxílio de uma matriz 4×4 é possível representar uma transformação afim de forma linear.

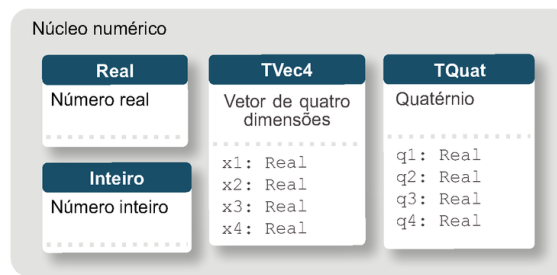


Figura 8.3: Núcleo numérico.

8.2.2

Estruturas auxiliares

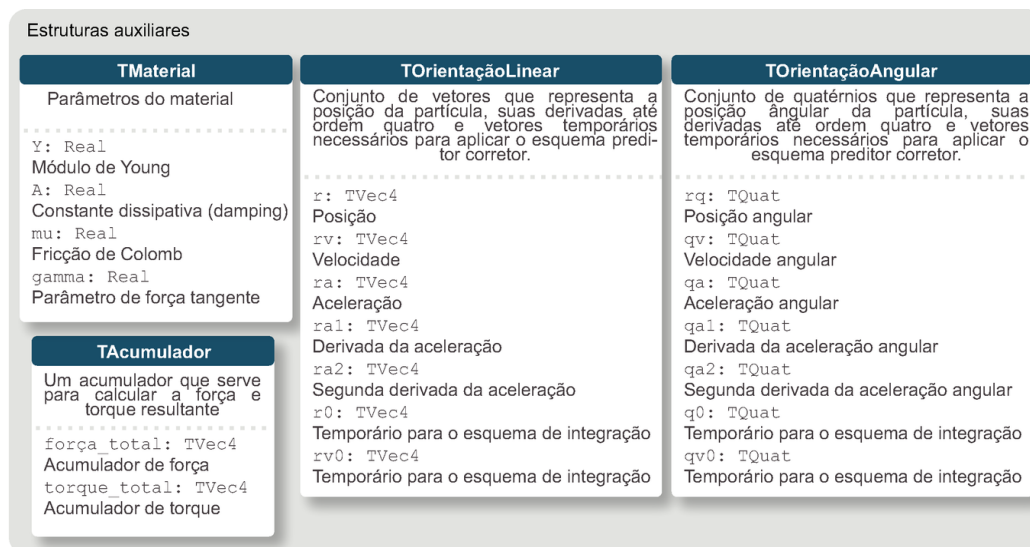


Figura 8.4: Estruturas auxiliares.

Nesse módulo são definidas diversas estruturas que a composição e o estado de um grão. Essas estruturas serão instanciadas em cada partícula do modelo.

TAcumulador: Armazena a força total e o torque total atuando sobre a partícula.

TMaterial: Parâmetros do material da partícula.

TOrientaçãoLinear: Armazena a posição da partícula, e suas derivadas, que são empregadas durante a etapa da integração do movimento da partícula. Essa classe também registra os valores temporários que serão utilizados no esquema de integração.

TOrientaçãoAngular: Idem, serve para armazenar a posição angular.

8.2.3

Estruturas elementares do modelo

Estruturas elementares do modelo	
TPartícula	TContato
Uma partícula do modelo	Parâmetros do contato
identificador: Inteiro Identificador da partícula, cada grão possui um diferente tipo: Inteiro Um inteiro que define se a partícula é de bordo, normal ou está desabilitada raio: Real Raio da partícula massa: Real Massa do grão material: TMaterial Parâmetros do material da partícula acumulador: TAcumulador Força e torque totais atuando na partícula orientação_linear: TOrientaçãoLinear Posições e derivadas orientação_angular: TOrientaçãoAngular Posições angulares e derivadas	identificador_p1: Inteiro Identificador da primeira partícula do contato identificador_p2: Inteiro Identificador da segunda partícula do contato ponto1: TVec4 Ponto de contato na superfície da primeira partícula, definido em coordenadas locais da primeira partícula ponto2: TVec4 Ponto de contato na superfície da segunda partícula, definido em coordenadas locais da segunda partícula k: Real Parâmetro elástico do contato sigma: Real Parâmetro de resistência elástica do contato A_critico: Real Valor do amortecimento crítico calculado quando o contato é inicializado, podendo também ser parametrizado.

Figura 8.5: Estruturas elementares do modelo.

Estruturas de partícula e contato geométrico. Essas estruturas definem os elementos básicos utilizados na modelagem, isso é, são utilizadas para guardar as informações de cada grão e de cada contato do modelo.

TPartícula: Informação sobre a geometria e parâmetros físicos da partícula. Essa estrutura conta com um número identificador único (**identificador**) que é utilizado mais tarde para definir os contatos. Outro campo importante é o **tipo** que define se a partícula é de bordo, normal ou está morta (não afeta a simulação).

TContato: Informação sobre um único contato. Define a geometria e os parâmetros que definem o contato.

8.2.4

Estruturas compostas do modelo

Esse módulo contém a estrutura principal do programa. No contexto de “*design patterns*” a classe TGBPM é um “*Singleton*”, para mais informações sobre “*design patterns*” é recomendado o livro (28).

TGBPM: Guarda toda a informação da simulação. Essa classe possui dois containeres, um do tipo vetor onde cada elemento é um objeto **TPartícula**, definindo cada grão da simulação. O segundo container é uma lista onde cada elemento é um objeto **TContato**, dessa vez definindo cada contato do modelo.

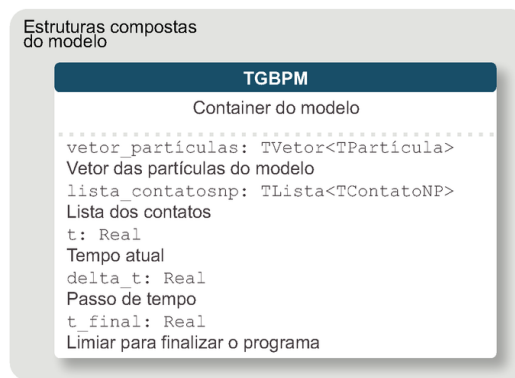


Figura 8.6: Estruturas compostas do modelo.

8.3

Algoritmo completo

8.3.1

Notação adotada

Durante esse capítulo, serão aplicadas as seguintes regras de notação:

Uma linha sob uma variável, representa uma variável local do programa.

Como exemplo \underline{F} .

A seta para a esquerda, representa atribuição de valor. Por exemplo, $\underline{F} \leftarrow x$ representa que a variável local F recebe o valor x .

Optamos por utilizar os nomes das palavras chave de programação (**for**, **while**, **return**) em negrito e na língua inglesa.

Para acessar o campo de um objeto, será utilizado o nome do campo abaixo do nome da instância do objeto. Por exemplo, P_{raio} representa o valor raio da instância P .

8.3.2

Descrição dos algoritmos

O algoritmo 1, descreve o algoritmo completo da simulação. A primeira função é a **Iniciação()**, cuja função é iniciar o modelo, essa função é descrita na seção 6.1.

Depois de iniciado, o programa entra no laço principal controlado pela função **Análise()** cujo pseudocódigo se encontra no algoritmo 2. Essa função tem dois objetivos: o primeiro é salvar informações, e resultados, sobre o estado atual da simulação. O segundo objetivo, é verificar o critério de parada do programa. Em particular, o laço é finalizado quando o tempo chegar a um dado tempo limite.

Algorithm 1 Simulação

```

Inicialização()
while Análise() do
  Preditor()
  Fronteiras()
  Forças()
  Corretor()
   $t \leftarrow t + \Delta t$ 
end while

```

Algorithm 2 Análise()

```

if  $t > t_{\text{final}}$  then
  return false
else
  GravaInformações()
  return true
end if

```

A próxima etapa é a função `Preditor()`. Essa função aplica a série de Taylor para prever os valores das posições e velocidades dos grãos, no tempo $t + \Delta t$. Os detalhes dessa função são derivados das seções 3 e 7.1.

Algorithm 3 Preditor()

```

 $\underline{c}_1 \leftarrow \Delta t$ 
 $\underline{c}_2 \leftarrow \underline{c}_1 \Delta t / 2$ 
 $\underline{c}_3 \leftarrow \underline{c}_2 \Delta t / 3$ 
 $\underline{c}_4 \leftarrow \underline{c}_4 \Delta t / 4$ 
for  $P \in \text{Partículas}$  do
   $P_r \leftarrow P_r + \underline{c}_1 P_{rv} + \underline{c}_2 P_{ra} + \underline{c}_3 P_{ra1} + \underline{c}_4 P_{ra2}$ 
   $P_{rv} \leftarrow P_{rv} + \underline{c}_1 P_{ra} + \underline{c}_2 P_{ra1} + \underline{c}_3 P_{ra2}$ 
   $P_{ra} \leftarrow P_{ra} + \underline{c}_1 P_{ra1} + \underline{c}_2 P_{ra2}$ 
   $P_{ra1} \leftarrow P_{ra1} + \underline{c}_1 P_{ra2}$ 
   $P_q \leftarrow P_q + \underline{c}_1 P_{qv} + \underline{c}_2 P_{qa} + \underline{c}_3 P_{qa1} + \underline{c}_4 P_{qa2}$ 
   $P_{qv} \leftarrow P_{qv} + \underline{c}_1 P_{qa} + \underline{c}_2 P_{qa1} + \underline{c}_3 P_{qa2}$ 
   $P_{qa} \leftarrow P_{qa} + \underline{c}_1 P_{qa1} + \underline{c}_2 P_{qa2}$ 
   $P_{qa1} \leftarrow P_{qa1} + \underline{c}_1 P_{qa2}$ 
end for

```

O cálculo das forças é executado no procedimento `Forças()`. Veja o algoritmo 4. Em primeiro lugar, os valores do acumulador de força e torque, das partículas, são zerados, em seguida, o cálculo da força definida no contato partícula-partícula é efetuado.

O segundo “**for**” do laço duplo utiliza a estrutura de dados descrita na seção 7.2. Essa estrutura é utilizada para encontrar, de forma eficiente, os grãos em contato mecânico com determinada partícula.

Dentro do laço duplo, as componentes normais e tangentes da força de contato são calculadas, conforme os modelos descritos na seção 4.3.1. E o vetor braço do momento de cada força é também calculado. Com isso, o torque gerado é definido. E os valores de força e torque são acumulados.

Na ultima etapa do procedimento, são efetuados os cálculos relativos ao modelo da rede de contatos, com a função `ModeloContato()`. A implementação dessa função é diretamente baseada na modelagem força de contato, descrita na seção 5.2. Mais uma vez, é encontrado o vetor braço da força e os valores do torque. Finalmente, esses valores são acumulados.

A função `Corretor()`, itera sobre as partículas, determinando o valor do resíduo definido na equação (7-1), tanto para a aceleração linear $\underline{P_{ra}^{res}}$, quanto para a aceleração angular $\underline{P_{qa}^{res}}$. Depois, a fórmula de correção é aplicada. A fórmula é definida na equação (7-2). O símbolo \otimes , representa a operação de multiplicação elemento a elemento, e é definida a seguir.

Definição 8.1 *Sejam dois quatérnios, $P = [p_1, p_2, p_3, p_4]$ e $Q = [q_1, q_2, q_3, q_4]$, a operação de multiplicação elemento a elemento é definida da seguinte forma:*

$$\begin{aligned} \otimes : \mathbb{Q} \times \mathbb{Q} &\mapsto \mathbb{Q} \\ (P, Q) &\mapsto P \otimes Q \equiv [p_1q_1, p_2q_2, p_3q_3, p_4q_4] \end{aligned}$$

Durante o passo corretor, a função `AceleraçãoAngular()` é utilizada. Nessa função, a aceleração angular é corrigida, em termos de quatérnios, com os valores da velocidade angular da partícula P_{qv} , e os componentes do tensor de inércia I_{ii} . O procedimento para efetuar esses cálculos estão descritos nas seções 6 e 6.2.

Algorithm 4 Forças()

```

for  $P \in$  Partículas do
   $P_{\text{torque\_total}} \leftarrow 0$ 
   $P_{\text{força\_total}} \leftarrow 0$ 
end for
ForçasExternas()
for  $P^i \in$  Partículas do
  for  $P^j \in$  Partículas em contato mecânico com  $P^i$  do
     $\overline{F}_{ij}^n \leftarrow$  ModeloForçaNormal()
     $\overline{F}_{ij}^t \leftarrow$  ModeloForçaTangente()
     $\overline{F}_{ij} \leftarrow \overline{F}_{ij}^n \overline{e}_{ij}^n + \overline{F}_{ij}^t \overline{e}_{ij}^t$ 
     $\underline{P}_{\text{braço}}^j \leftarrow \frac{P_{\text{raio}}^i + P_{\text{raio}}^j}{P_{\text{raio}}^j} (P_r^i - P_r^j)$ 
     $\underline{P}_{\text{braço}}^i \leftarrow \frac{P_{\text{raio}}^i + P_{\text{raio}}^j}{P_{\text{raio}}^i} (P_r^j - P_r^i)$ 
     $\overline{M}_{ij} \leftarrow \overline{F}_{ij} \times \underline{P}_{\text{braço}}^j$ 
     $\overline{M}_{ji} \leftarrow -\overline{F}_{ij} \times \underline{P}_{\text{braço}}^i$ 
     $P_{\text{torque\_total}}^i \leftarrow P_{\text{torque\_total}}^i + \overline{M}_{ji}$ 
     $P_{\text{torque\_total}}^j \leftarrow P_{\text{torque\_total}}^j + \overline{M}_{ij}$ 
     $P_{\text{força\_total}}^j \leftarrow P_{\text{força\_total}}^j + \overline{F}_{ij}$ 
     $P_{\text{força\_total}}^i \leftarrow P_{\text{força\_total}}^i - \overline{F}_{ij}$ 
  end for
end for
for  $C \in$  Contatos do
   $P^i \equiv$  Primeira partícula de  $C$ 
   $P^j \equiv$  Segunda partícula de  $C$ 
   $\overline{F}_{ij} \leftarrow$  ModeloContatoNP()
   $\underline{P}_{\text{braço}}^j \leftarrow \frac{P_{\text{raio}}^i + P_{\text{raio}}^j}{P_{\text{raio}}^j} (P_r^i - P_r^j)$ 
   $\underline{P}_{\text{braço}}^i \leftarrow \frac{P_{\text{raio}}^i + P_{\text{raio}}^j}{P_{\text{raio}}^i} (P_r^j - P_r^i)$ 
   $\overline{M}_{ij} \leftarrow \overline{F}_{ij} \times \underline{P}_{\text{braço}}^j$ 
   $\overline{M}_{ji} \leftarrow -\overline{F}_{ij} \times \underline{P}_{\text{braço}}^i$ 
   $P_{\text{torque\_total}}^i \leftarrow P_{\text{torque\_total}}^i + \overline{M}_{ji}$ 
   $P_{\text{torque\_total}}^j \leftarrow P_{\text{torque\_total}}^j + \overline{M}_{ij}$ 
   $P_{\text{força\_total}}^j \leftarrow P_{\text{força\_total}}^j + \overline{F}_{ij}$ 
   $P_{\text{força\_total}}^i \leftarrow P_{\text{força\_total}}^i - \overline{F}_{ij}$ 
end for

```

Algorithm 5 Corretor()

$$\underline{c}_0 \leftarrow \frac{19}{90} \frac{\Delta t^2}{2}$$

$$\underline{c}_1 \leftarrow \frac{3}{4} \frac{1!}{(\Delta t)^1} \frac{\Delta t^2}{2}$$

$$\underline{c}_2 \leftarrow 1 \frac{2!}{(\Delta t)^2} \frac{\Delta t^2}{2}$$

$$\underline{c}_3 \leftarrow \frac{1}{2} \frac{3!}{(\Delta t)^3} \frac{\Delta t^2}{2}$$

$$\underline{c}_4 \leftarrow \frac{1}{12} \frac{4!}{(\Delta t)^4} \frac{\Delta t^2}{2}$$

for $P \in$ Partículas **do**

$$\underline{P}_{ra}^{cor} \leftarrow \frac{(P_{força_total})}{P_{massa}}$$

$$\underline{P}_{ra}^{res} \leftarrow P_{ra} - \underline{P}_{ra}^{cor}$$

$$\underline{P}_r \leftarrow P_r + \underline{c}_0 \underline{P}_r \otimes \underline{P}_{ra}^{res}$$

$$\underline{P}_{rv} \leftarrow P_{rv} + \underline{c}_1 \underline{P}_{rv} \otimes \underline{P}_{ra}^{res}$$

$$\underline{P}_{ra} \leftarrow P_{ra} + \underline{c}_2 \underline{P}_{ra} \otimes \underline{P}_{ra}^{res}$$

$$\underline{P}_{ra1} \leftarrow P_{ra1} + \underline{c}_3 \underline{P}_{ra1} \otimes \underline{P}_{ra}^{res}$$

$$\underline{P}_{ra2} \leftarrow P_{ra2} + \underline{c}_4 \underline{P}_{ra2} \otimes \underline{P}_{ra}^{res}$$

end for

for $P \in$ Partículas **do**

$$\underline{P}_{qa}^{cor} \leftarrow \text{Aceleração}\hat{\text{Angular}}()$$

$$\underline{P}_{qa}^{res} \leftarrow P_{qa} - \underline{P}_{qa}^{cor}$$

$$\underline{P}_q \leftarrow P_q + \underline{c}_0 \underline{P}_q \otimes \underline{P}_{qa}^{res}$$

$$\underline{P}_{qv} \leftarrow P_{qv} + \underline{c}_1 \underline{P}_{qv} \otimes \underline{P}_{qa}^{res}$$

$$\underline{P}_{qa} \leftarrow P_{qa} + \underline{c}_2 \underline{P}_{qa} \otimes \underline{P}_{qa}^{res}$$

$$\underline{P}_{qa1} \leftarrow P_{qa1} + \underline{c}_3 \underline{P}_{qa1} \otimes \underline{P}_{qa}^{res}$$

$$\underline{P}_{qa2} \leftarrow P_{qa2} + \underline{c}_4 \underline{P}_{qa2} \otimes \underline{P}_{qa}^{res}$$

end for

Algorithm 6 Aceleração $\hat{\text{Angular}}()$

$$\underline{\omega} \leftarrow -\frac{1}{2} P_{qv} P_{qv}^* // \text{velocidade angular}$$

$$\underline{T}_{qa}[1] \leftarrow \frac{1}{I_{11}} P_{torque_total}[1] + (I_{22} - I_{33}) \underline{\omega}[2] \underline{\omega}[3]$$

$$\underline{T}_{qa}[2] \leftarrow \frac{1}{I_{22}} P_{torque_total}[2] + (I_{33} - I_{11}) \underline{\omega}[1] \underline{\omega}[3]$$

$$\underline{T}_{qa}[3] \leftarrow \frac{1}{I_{33}} P_{torque_total}[3] + (I_{22} - I_{11}) \underline{\omega}[1] \underline{\omega}[2]$$

$$\underline{T}_{qa}[4] \leftarrow -2 \sum_{i=1}^4 (P_{qv}[i])^2$$

return $\frac{1}{2} \underline{P}_q \underline{T}_{qa}$
