

Manual Científico do PyParticles

Simuldor Numérico de Partículas

Versão: 1.0

Data: 14 de dezembro de 2016

Autor: Bismarck Gomes Souza Júnior

RIO DE JANEIRO - RJ

Sumário

1	Introdução	1
1.1	Escopo do Problema	1
1.2	PyParticles	1
1.3	Objetivos	2
2	Contato entre Partículas	3
2.1	Partículas	3
2.2	Identificação do Contato	4
2.3	Forças no Contato	6
2.4	Busca de Contatos	13
3	Metodologia Numérica e Computacional	15
3.1	Equação do Movimento	15
3.2	Discretização da Equação do Movimento	16
3.3	Fluxograma do Processo de Solução	18
3.4	Modelagem Computacional	19

Lista de Figuras

2.1	Representação geométrica de uma partícula	3
2.2	Representação geométrica de duas partículas	4
2.3	Deformação no contato: representação didática e real	5
2.4	Tipos de contato: pseudo-contato e contato verdadeiro	5
2.5	Forças elásticas entre duas partículas: (a) atração, (b) nula e (c) repulsão	7
2.6	Definição do vetor unitário \hat{n} em função de duas partículas considerando a partícula 0 como referência	7
2.7	Representação das velocidades relativas no plano de contato entre duas partículas	8
2.8	Sistema massa-mola-amortecedor	10
2.9	Força resultante no contato entre duas partículas	12
2.10	Fluxograma para verificar o contato entre duas partículas	13
2.11	Fluxograma da busca de contatos	14
3.1	Forças atuantes em uma partícula	16
3.2	Fluxograma que demonstra que as acelerações de cada partícula de- pende das suas posições	16
3.3	Variação da propriedades cinemáticas no intervalo de tempo Δt_n : (a) aceleração, (b) velocidade e (c) posição	17
3.4	Variação da velocidade no intervalo Δt_n	18
3.5	Fluxograma do processo de solução	19
3.6	Diagrama de classes para as partículas	20
3.7	Diagrama de classes para as fronteiras	20
3.8	Diagrama de classes para os contatos	21
3.9	Diagrama de classes para as forças	21

3.10 Diagrama de classes principal do simulador PyParticles 22

1 Introdução

1.1 Escopo do Problema

Um simulador numérico de partículas é um programa capaz de determinar a configuração geométrica e cinemática de um conjunto de partículas em determinados espaços de tempo.

A partícula, geralmente esférica, é a representação de um material físico desde pequenas escalas até grande escalas como moléculas, grãos, esferas maciças ou planetas. Cada representação possui características e considerações diferentes e devem ser analisadas caso a caso durante a execução do simulador.

Além das partículas existem as paredes ou fronteiras fixas que estabelecem condições de contorno das mais variadas formas e com as quais a aplicabilidade de um simulador de partículas aumenta consideravelmente.

Dada a importância de um simulador deste tipo, foi desenvolvido um simulador numérico de partículas denominado PyParticles.

1.2 PyParticles

O desenvolvimento do PyParticles teve início em outubro de 2016 como parte das exigências para conclusão da disciplina de Modelagem Computacional de Partículas do curso de Pós-Graduação em Engenharia Civil da PUC-Rio.

O PyParticles consiste em um simulador numérico de partículas totalmente desenvolvido em Python 2.7. Para a visualização dos resultados é utilizado um programa externo livre: o ParaView 5.0.1.

O simulador conta hoje com um número ilimitado de partículas esféricas rígidas e fronteiras planas, nas quais atuam forças elásticas amortecidas, gravitacionais e externas. Porém nenhuma comparação com soluções analíticas foi realizada.

O PyParticles está em fase de desenvolvimento, visto que diversas funcionalidades ainda podem ser introduzidas, como:

- Aumento do grau de liberdade do sistema com a inclusão dos ângulos de rotação;
- Métodos de busca mais otimizados;
- Implementação de fronteiras servo-controladas;
- Criação de conjunto de partículas que compatilham as mesmas propriedades;
- Implementação de discretizações mais convencionais;

1.3 Objetivos

O principal objetivo do trabalho é apresentar a formulação numérica utilizada para desenvolver o PyParticles, desde a definição básica das propriedades utilizadas até a formulação matemática da equação de movimento das partículas.

No Capítulo 2 será apresentado a definição de contato, a condição para sua ocorrência e a metodologia necessária para determinar as forças que atuam nos contatos. No final do capítulo será apresentado um método elementar capaz de determinar todos os contatos existentes em um sistema de partículas.

O equacionamento da física do problema será discutido no Capítulo 3, no qual será apresentada a equação de movimento das partículas e a sua discretização temporal e espacial. No final do capítulo será apresentado a modelagem computacional utilizada para desenvolver o PyParticles.

2 *Contato entre Partículas*

Neste capítulo será definida as características de uma partícula, a forma como se identifica um contato, o cálculo das forças no contato e um método de busca de contatos.

2.1 Partículas

No PyParticles as partículas são consideradas esferas rígidas, podendo representar materiais em diferentes escalas como átomos, grãos ou até mesmo bolas de bilhar.

Cada partícula é caracterizada geometricamente pela sua posição, \vec{x} , e seu raio, r ; cinematicamente, pela velocidade, \vec{v} , e pela aceleração, \vec{a} ; e fisicamente, pela rigidez normal, k_n , pela rigidez cisalhante, k_s , pelo fator de amortecimento normal, β_n , e cisalhante, β_s .

Um exemplo da representação geométrica de uma partícula é apresentada na Fig. 2.1, na qual o centro da partícula de raio r_0 está localizado na posição \vec{x}_0 .

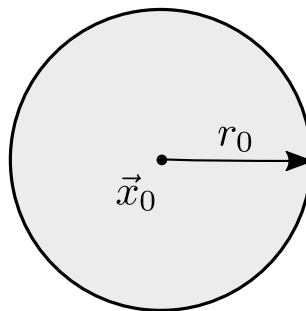


Figura 2.1: Representação geométrica de uma partícula

2.2 Identificação do Contato

A principal interação entre as partículas é dada pelo contato entre elas. Desta forma, é de fundamental importância que seja definida uma condição que represente o contato, considerando apenas as propriedades geométrica das partículas.

A Fig. 2.2 representa duas esferas de raio r_0 e r_1 cujos os centros estão localizados nas posições \vec{x}_0 e \vec{x}_1 , respectivamente. A distância d entre seus centros geométricos pode ser obtida aplicando-se a norma no vetor delimitado pelos centros, ou seja:

$$d = \|\vec{x}_1 - \vec{x}_0\| \quad (2.1)$$

A distância entre as bordas das partículas, d_b , pode ser obtida em função da distância entre os centros, d , e dos raios, tal que:

$$d_b = d - r_1 - r_0 \quad (2.2)$$

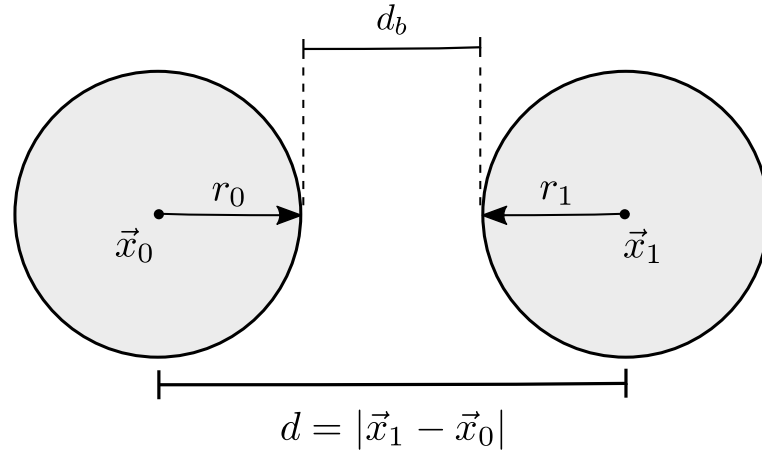


Figura 2.2: Representação geométrica de duas partículas

Como a distância d_b representa a distância entre as partículas, então quando $d_b = 0$, as partículas estarão em contato. Caso as partículas se aproximem ainda mais, representando uma sobreposição hipotética, a distância d entre os centros irá diminuir e, conseqüentemente, $d_b < 0$. Esta sobreposição é uma representação didática para a ligeira deformação que ocorre nos contatos, conforme pode ser visto na Fig. 2.3.

Por outro lado, conforme as partículas se afastam, d_b aumenta e, por isso, a representação matemática para o não contato entre as partículas é: $d_b > 0$. Porém, como será considerado que há uma força de atração entre as partículas quando estas estão próximas, é necessário definir o conceito de pseudo-contato.

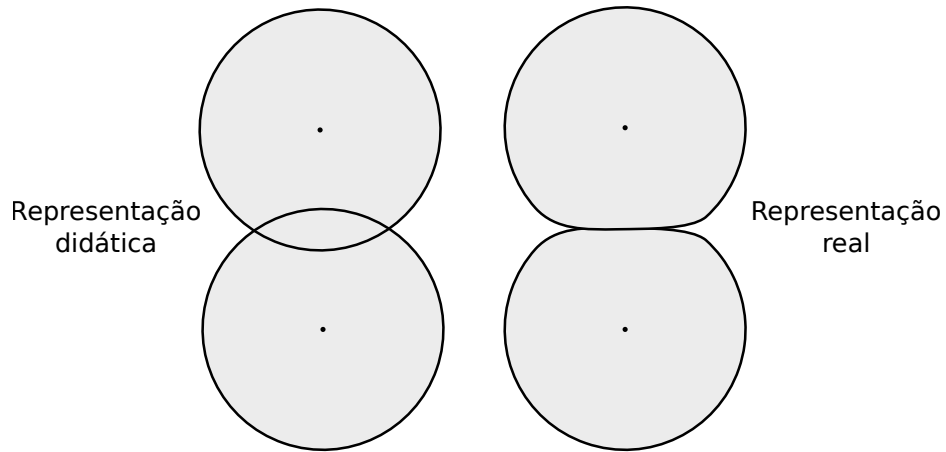


Figura 2.3: Deformação no contato: representação didática e real

Como o contato é a única forma de interação entre as partículas, o conceito de pseudo-contato se aplica quando há uma força de atração entre elas que atua até uma distância máxima estabelecida, $d_{b,max}$. Esta força de atração diminui conforme as partículas se aproximam e atinge o valor nulo quando ocorre o contato. Caso as partículas continuem se aproximando, surgirá uma força de repulsão entre elas que aumenta em magnitude conforme $|d_b|$ aumenta. A Fig. 2.4 ilustra melhor estes conceitos.

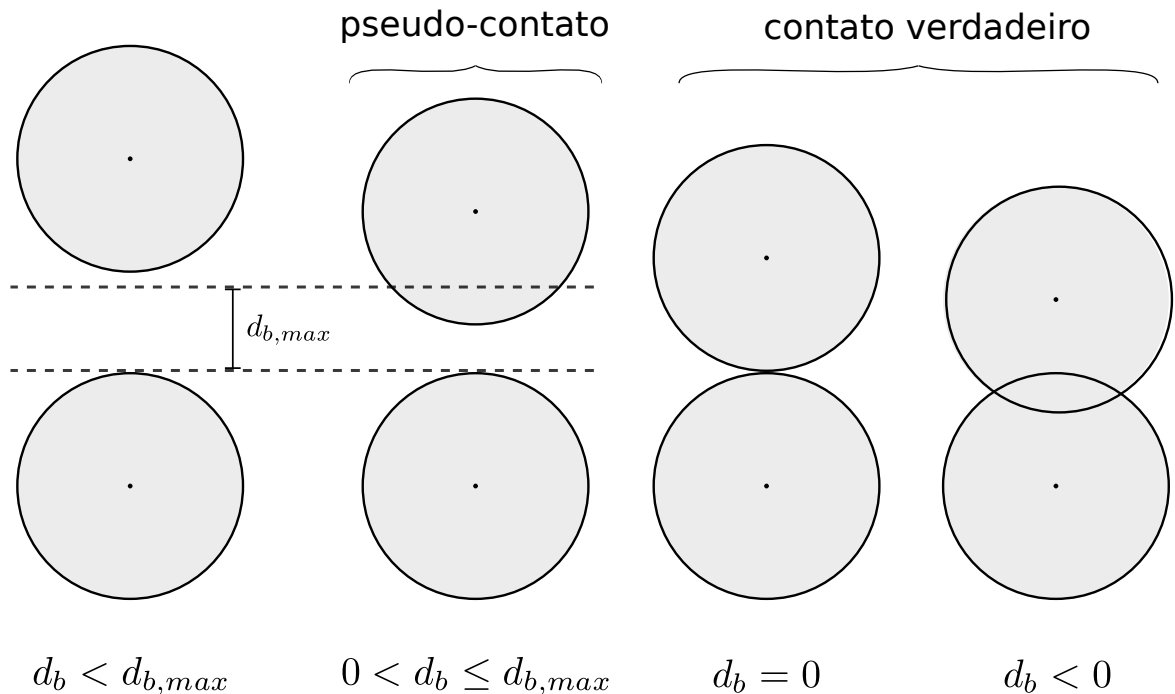


Figura 2.4: Tipos de contato: pseudo-contato e contato verdadeiro

2.3 Forças no Contato

Dependendo do tipo do contato podem existir dois tipos de forças: atração ou repulsão. Por hipótese estas forças serão admitidas elásticas, tal que seja válida a lei de Hooke, e amortecidas, a fim de representar casos mais realísticos. As forças que atuam no contato entre as partículas serão decompostas em forças normais e cisalhantes.

2.3.1 Força Elástica

A força elástica em uma mola, de acordo com a lei de Hooke, é diretamente proporcional ao seu deslocamento a partir do ponto de equilíbrio. A força elástica, \vec{F}_e , que atua no contato pode ser decomposta em força normal e cisalhante tal que:

$$\vec{F}_e = \vec{F}_n + \vec{F}_s \quad (2.3)$$

onde \vec{F}_n é a força normal e \vec{F}_s é a força cisalhante. Estas forças podem ser analisadas de forma independente, conforme será apresentado a seguir.

Força Elástica Normal

Considerando que duas partículas se deslocam em um mesmo eixo e que não há forças de amortecimento, então as possíveis interações entre elas é dada conforme a Fig. 2.5.

Seja \hat{n} o vetor unitário que representa a direção que une os eixos de duas partículas tal que o sentido é definido com relação a partícula de referência 0, conforme a Fig. 2.6, então este vetor pode ser escrito como:

$$\hat{n} = \frac{\vec{x}_1 - \vec{x}_0}{\|\vec{x}_1 - \vec{x}_0\|} \quad (2.4)$$

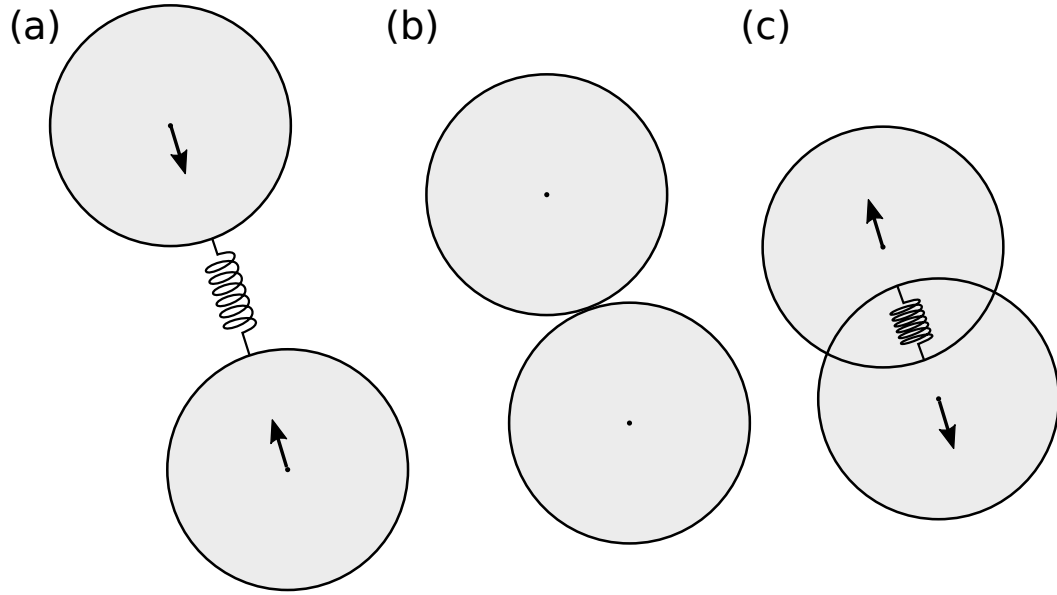


Figura 2.5: Forças elásticas entre duas partículas: (a) atração, (b) nula e (c) repulsão

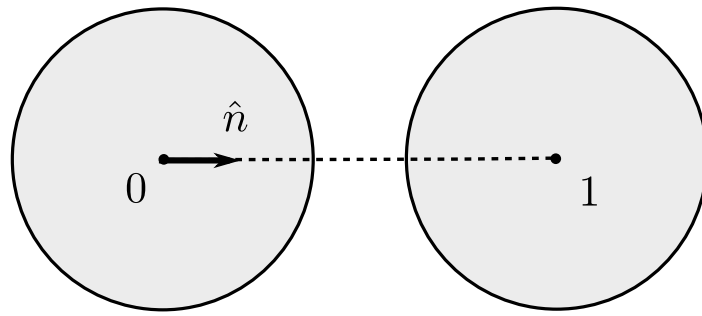


Figura 2.6: Definição do vetor unitário \hat{n} em função de duas partículas considerando a partícula 0 como referência

Assim, a força elástica normal ao contato entre duas partículas pode ser escrita como:

$$\vec{F}_n = \hat{n} \|\vec{F}_n\| \quad (2.5)$$

tal que $\|\vec{F}_n\|$ representa a magnitude da força elástica, que, segundo a lei de Hooke, pode ser expressa como:

$$\|\vec{F}_n\| = \begin{cases} -k_{nr}d_b & , d_b \leq 0 \\ +k_{na}d_b & , 0 < d_b < d_{b,max} \\ 0 & , d_b > d_{b,max} \end{cases} \quad (2.6)$$

Esta equação considera a possibilidade da rigidez das partículas durante o fenô-

meno da repulsão, k_{nr} , seja diferente do parâmetro de atração, k_{na} . Desta forma é possível considerar uma força de cimentação (atração) entre grãos seja menor do que a força de compactação (repulsão).

Força Elástica Cisalhante

A força elástica cisalhante no contato pode ser expressa, de acordo com a lei de Hooke, em função da distância tangencial que, por sua vez, pode ser expressa pelo produto da velocidade cisalhante, \vec{v}_s , com o tempo de contato entre as partículas, Δt_c , ou seja:

$$\vec{F}_s = k_s \vec{v}_s \Delta t_c \quad (2.7)$$

A velocidade cisalhante é a velocidade tangencial relativa entre as duas partículas em contato, conforme a Fig. 2.7. Nesta figura, 0 representa o centro da partícula de referência; 1, o centro da partícula em contato; \vec{v}_0 , a velocidade da partícula 0; \vec{v}_1 , a velocidade da partícula 1; \vec{v}_r , a velocidade relativa entre elas; \vec{v}_n , a velocidade relativa normal ao plano de contato; e \vec{v}_s , a velocidade relativa tangencial ao plano de contato.

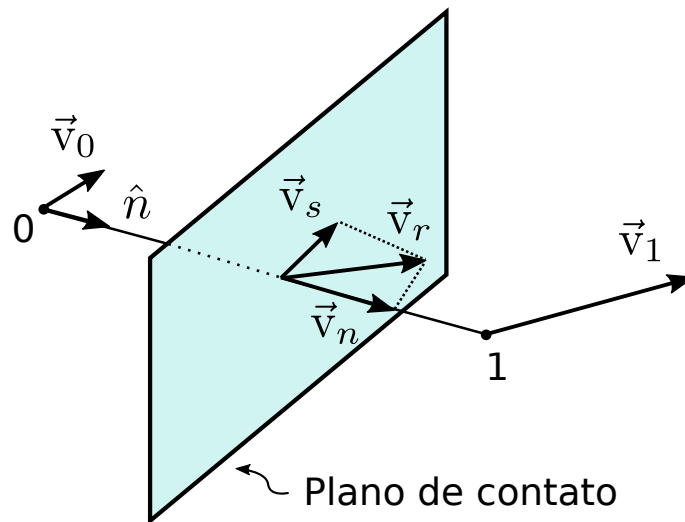


Figura 2.7: Representação das velocidades relativas no plano de contato entre duas partículas

A velocidade relativa entre as partículas é definida por:

$$\vec{v}_r = \vec{v}_1 - \vec{v}_0 \quad (2.8)$$

e pode ser expressa em função de \vec{v}_n e \vec{v}_s como:

$$\vec{v}_r = \vec{v}_n + \vec{v}_s \quad (2.9)$$

A velocidade \vec{v}_n é a projeção da velocidade \vec{v}_r sobre o eixo \hat{n} , ou seja:

$$\vec{v}_n = \left(\frac{\hat{n} \cdot \vec{v}_r}{\|\vec{v}_r\|^2} \right) \hat{n} \quad (2.10)$$

Por outro lado, a velocidade \vec{v}_s pode ser expressa através da manipulação das Eqs. (2.9) e (2.10), tal que:

$$\vec{v}_s = \vec{v}_r - \left(\frac{\hat{n} \cdot \vec{v}_r}{\|\vec{v}_r\|^2} \right) \hat{n} \quad (2.11)$$

Embora a força elástica cisalhante seja obtida pela Eq. (2.7), há uma limitação desta força imposta pela lei de atrito de Coulomb, tal que:

$$\|\vec{F}_s\| \leq \mu \|\vec{F}_n\| \quad (2.12)$$

onde μ é o coeficiente de atrito do contato. Ou seja, a partir de um momento, independente do tempo de contato, a força cisalhante sempre será a mesma e igual ao valor máximo:

$$\|\vec{F}_s\|_{max} = \mu \|\vec{F}_n\| \quad (2.13)$$

2.3.2 Força de Amortecimento

O modelo de amortecimento utilizado será o viscoelástico, no qual o amortecimento é viscoso tal que sua força é proporcional à velocidade, representando uma oposição ao movimento, ou seja:

$$\vec{F}_d = -c\vec{v} \quad (2.14)$$

onde c é a constante de amortecimento e \vec{v} é a velocidade da massa em relação a um ponto fixo.

Para um sistema massa-mola-amortecedor unidimensional cuja a constante de elasticidade da mola é k e a constante de amortecimento é c , conforme a Fig. 2.8, é válido o equilíbrio de forças:

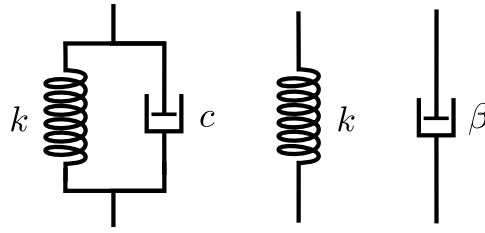


Figura 2.8: Sistema massa-mola-amortecedor

$$m\ddot{x} = -c\dot{x} - kx \quad (2.15)$$

ou ainda:

$$m\ddot{x} + c\dot{x} + kx = 0 \quad (2.16)$$

cujo polinômio característico escrito em função da variável r é dado por:

$$mr^2 + cr + k = 0 \quad (2.17)$$

Quando esta equação possui uma única solução, o sistema é dito criticamente amortecido. Esta hipótese ocorre quando o discriminante da equação é nulo, ou seja:

$$\Delta = c^2 - 4mk = 0 \quad (2.18)$$

Dessa forma, a constante crítica de amortecimento é calculada resolvendo-se a equação, tal que:

$$c_c = 2\sqrt{mk} \quad (2.19)$$

Uma forma alternativa de representar o amortecimento é através do fator de amortecimento, β , definido por:

$$\beta = \frac{c}{c_c} \quad (2.20)$$

Quando $\beta = 1$ o sistema é denominado criticamente amortecido; para $\beta > 1$, super-amortecido; e para $\beta < 1$, sub-amortecido.

Assim, a força de amortecimento escrita em função do fator de amortecimento é dada por:

$$\vec{F}_d = - \left(2\beta \sqrt{mk} \right) \vec{v} \quad (2.21)$$

A extensão para as forças de amortecimento nos contatos entre partículas será realizada através da decomposição desta força nas componentes normal, \vec{F}_{dn} , e cisalhante, \vec{F}_{ds} , tal que:

$$\vec{F}_d = \vec{F}_{dn} + \vec{F}_{ds} \quad (2.22)$$

onde as forças normal e cisalhante serão detalhadas a seguir.

Força de Amortecimento Normal

Analogamente a Eq. (2.21), a força de amortecimento normal pode ser escrita como:

$$\vec{F}_{dn} = - \left(2\beta_n \sqrt{m_c k_n} \right) \vec{v}_n \quad (2.23)$$

onde β_n é o fator de amortecimento normal ao contato; k_n , a constante de rigidez normal da mola; \vec{v}_n , a velocidade relativa entre as partículas normal ao contato, Eq. (2.10); e m_c , a massa equivalente do contato, sendo definida, para o contato entre duas partículas 0 e 1, por:

$$m_c = \frac{m_0 m_1}{m_0 + m_1} \quad (2.24)$$

Força de Amortecimento Cisalhante

Analogamente a Eq. (2.21), a força de amortecimento cisalhante pode ser escrita como:

$$\vec{F}_{ds} = - \left(2\beta_s \sqrt{m_c k_s} \right) \vec{v}_s \quad (2.25)$$

onde β_s é o fator de amortecimento cisalhante ao contato; k_s , a constante de rigidez cisalhante da mola; \vec{v}_s , a velocidade relativa entre as partículas tangencial ao contato, Eq. (2.11); e m_c , a massa equivalente do contato, Eq. (2.24).

2.3.3 Força Resultante no Contato

A força resultante no contato j é a soma das forças elásticas e de amortecimento tanto normais quanto cisalhantes, conforme a Fig. 2.9, tal que:

$$\vec{F}_{c,j} = \vec{F}_n + \vec{F}_{dn} + \vec{F}_s + \vec{F}_{ds} \quad (2.26)$$

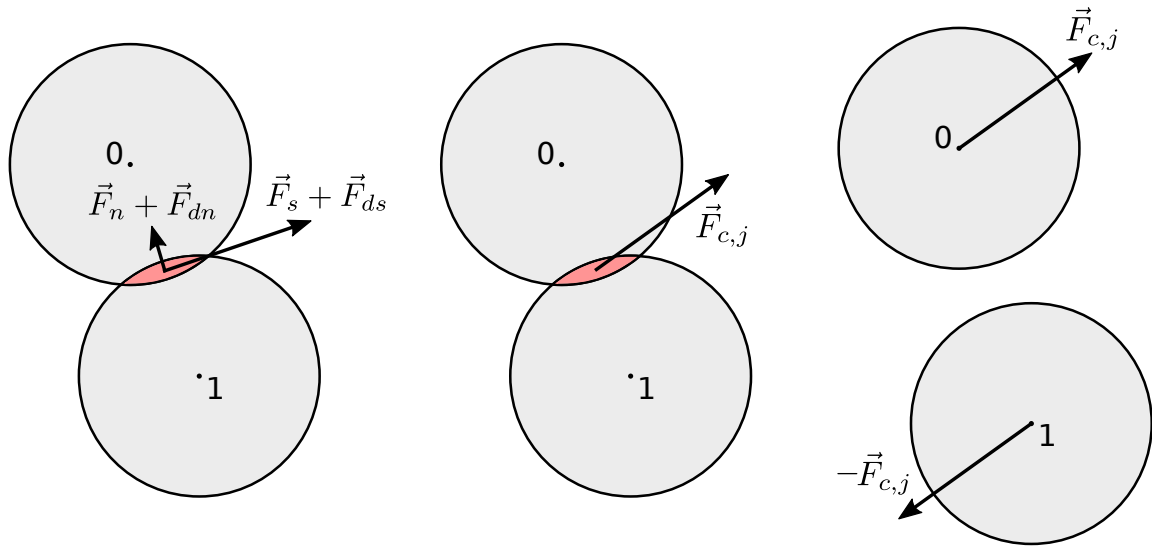


Figura 2.9: Força resultante no contato entre duas partículas

A força resultante no contato será sempre com relação a partícula de referência denotada didaticamente por 0. Durante a simulação numérica, a partícula de referência será aquela com menor índice, admitindo que cada partícula possuirá uma numeração. A força $\vec{F}_{c,j}$ atuará no centro de massa da partícula de referência, enquanto que a força oposta, $-\vec{F}_{c,j}$, atuará na partícula em contato, conforme a Fig. 2.9.

A força $\vec{F}_{c,j}$ representa a força em apenas um dos vários contatos que uma partícula pode ter. A força resultante de todos os contatos na partícula será o somatório de todas as forças que atuam em cada contato, ou seja:

$$\vec{F}_c = \sum_{j \in C} \vec{F}_{c,j} \quad (2.27)$$

onde C representa o conjunto de todos os contatos de uma partícula.

O procedimento para encontrar todos os contatos existentes no sistema será apresentado a seguir.

2.4 Busca de Contatos

A busca de contatos utilizada consiste no método de verificação por força bruta, verificando-se a distância d_b entre todas as partículas em todos os passos de tempo. A implementação de métodos mais robustos será uma sugestão para trabalhos futuros.

O primeiro passo para procurar pelos contatos entre as partículas consiste em definir uma condição para o contato. Para o uso de partículas esféricas, conforme a Fig. 2.4, a condição de contato é dada por:

$$d_b \leq d_{b,max} \quad (2.28)$$

onde $d_{b,max}$ é a distância máxima a partir da qual existirá interações entre as partículas. Este valor deve ser estabelecido conforme a natureza do fenômeno estudado.

A verificação do conatato entre duas partículas i e j é realizada conforme a Fig. 2.10. Caso seja verificado um contato e o mesmo não existir previamente, o contato deverá ser criado. Por outro lado, caso o contato já existir, o mesmo deverá ser atualizado. A atualização do contato é feita basicamente pela atualização do tempo de contato, tal que dado um intervalo de tempo de simulação de Δt , então o tempo de contato deverá ser incrementado Δt , ou seja:

$$\Delta t_c \leftarrow \Delta t_c + \Delta t \quad (2.29)$$

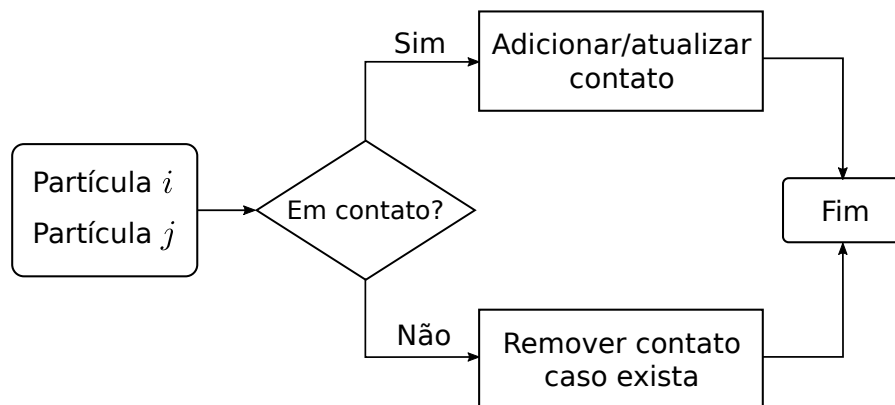


Figura 2.10: Fluxograma para verificar o contato entre duas partículas

Conforme mencionado o método de busca de contato é o de força bruta, na qual a verificação de todas as partículas 2 a 2 será realizado, conforme o fluxograma apresentado na Fig. 2.11, onde N representa o número de partículas; e i e j , os índices de 2 partículas.

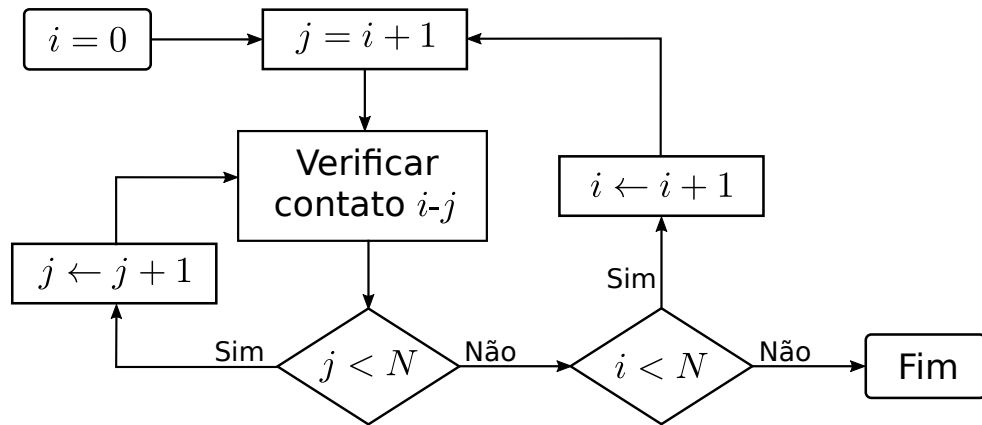


Figura 2.11: Fluxograma da busca de contatos

3 *Metodologia Numérica e Computacional*

Neste capítulo serão apresentadas as metodologias numérica e computacional utilizada para o desenvolvimento do simulador PyParticles.

3.1 Equação do Movimento

De acordo com a segunda lei de Newton, a força resultante que atua sobre um corpo é proporcional ao produto da massa pela aceleração por ele adquirida, ou seja:

$$\vec{F}_R = m\vec{a} \quad (3.1)$$

onde \vec{F}_R é a força resultante, m é a massa e \vec{a} é a aceleração da partícula. Conhecendo-se todas as forças que atuam em um partícula e, conseqüentemente, a força resultante é possível obter uma equação que represente o movimento desta partícula.

Considerando que no sistema de partículas, atuam sobre cada partícula apenas uma força externa, \vec{F}_{ext} , uma força gravitacional, \vec{F}_g , e a força resultante nos contatos, \vec{F}_c , conforme a Fig. 3.1, então a força resultante na partícula será:

$$\vec{F}_R = \vec{F}_c + \vec{F}_g + \vec{F}_{ext} \quad (3.2)$$

onde a força resultante dos contatos foi devidamente determinada no Capítulo 2 através da Eq. (2.27), e a força externa é estabelecida para cada partícula dependendo-se do fenômeno analisado.

Conhecendo-se as posições das partículas é possível determinar todos os contatos, conforme o fluxograma da Fig. 2.11, e, conseqüentemente, determinar a força resultante em cada partícula através da Eq. (3.2). Finalmente, a partir da Eq. (3.1) é possível obter a aceleração de cada partícula, sendo válido o fluxograma da Fig. 3.2.

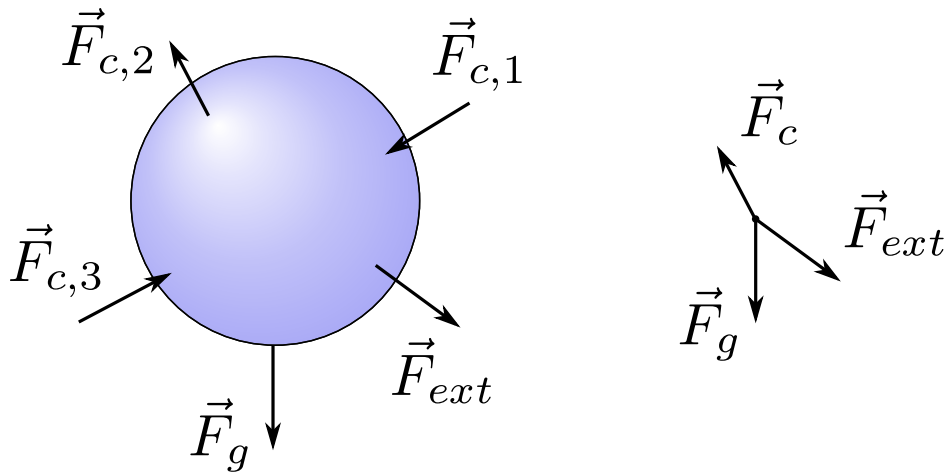


Figura 3.1: Forças atuantes em uma partícula

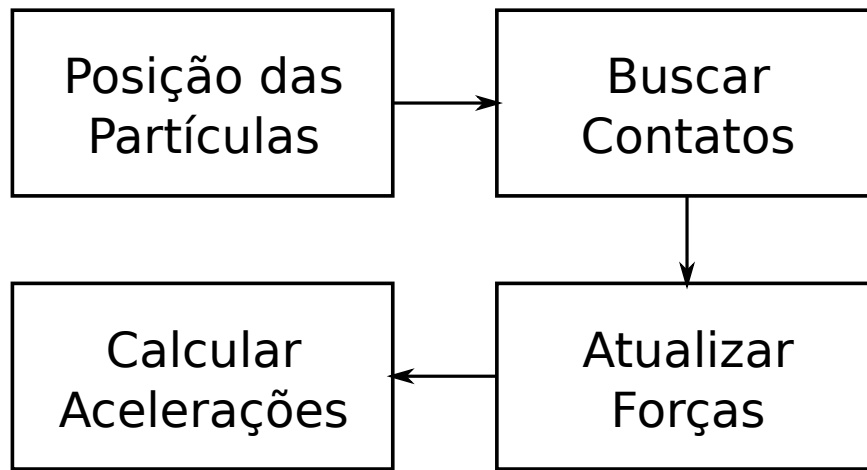


Figura 3.2: Fluxograma que demonstra que as acelerações de cada partícula depende das suas posições

Assim, determinado o vetor aceleração de cada partícula \vec{a} , dado por:

$$\vec{a} = \frac{\vec{F}_R}{m}, \quad (3.3)$$

a equação diferencial que representa o movimento de uma partícula será:

$$\frac{d^2 \vec{x}}{dt^2} = \vec{a} \quad (3.4)$$

3.2 Discretização da Equação do Movimento

Estabelecida a equação diferencial de movimento, é necessário aplicar uma discretização temporal a fim de obter uma relação para a posição da partícula em função

do passo de tempo considerado.

Considerando que no tempo $t = t_n$ a partícula encontra-se na posição $\vec{x}^{(n)}$ com velocidade $\vec{v}^{(n)}$ e aceleração $\vec{a}^{(n)}$, determinada conforme a Eq. (3.4) e admitida constante no intervalo de tempo considerado, então integrando-se a Eq. (3.4) com relação ao tempo de $t = t_n$ até t , tem-se:

$$\int_{t_n}^t \frac{d^2\vec{x}}{dt'^2} dt' = \int_{t_n}^t \vec{a}^{(n)} dt' \quad (3.5)$$

$$\frac{d\vec{x}}{dt} - \vec{v}^{(n)} = (t - t_n) \vec{a}^{(n)} \quad (3.6)$$

Integrando-se novamente com relação ao tempo de $t = t_n$ até t , tem-se:

$$\int_{t_n}^t \frac{d\vec{x}}{dt} dt' = \int_{t_n}^t \vec{v}^{(n)} dt' + \int_{t_n}^t (t' - t_n) \vec{a}^{(n)} dt' \quad (3.7)$$

$$[\vec{x}(t')]_{t_n}^t = [\vec{v}^{(n)} t']_{t_n}^t + \left[\vec{a}^{(n)} \left(\frac{1}{2} t'^2 - t_n t' \right) \right]_{t_n}^t \quad (3.8)$$

$$\vec{x} - \vec{x}^{(n)} = (t - t_n) \vec{v}^{(n)} + \frac{1}{2} (t - t_n)^2 \vec{a}^{(n)} \quad (3.9)$$

ou seja, a posição da partícula i no tempo $t = t_{n+1}$, considerando $\Delta t_n = t - t_n$, será:

$$\vec{x}_i^{(n+1)} = \vec{x}_i^{(n)} + \vec{v}_i^{(n)} \Delta t_n + \frac{1}{2} \vec{a}_i^{(n)} \Delta t_n^2 \quad (3.10)$$

O processo de integração realizado pode ser visto graficamente na Fig. 3.3, que representa um caso unidimensional, para a partícula i , no qual a aceleração é uma função constante, a velocidade uma função linear e a aceleração uma função parabólica.

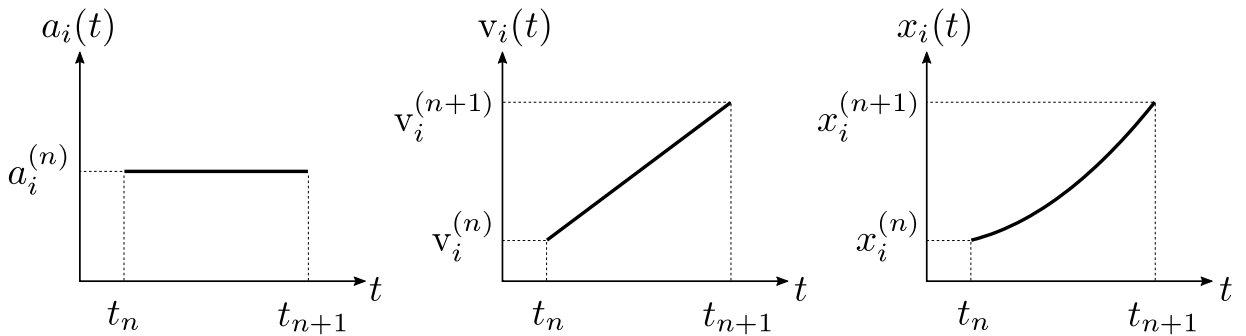


Figura 3.3: Variação das propriedades cinemáticas no intervalo de tempo Δt_n : (a) aceleração, (b) velocidade e (c) posição

Determinada as posições de todas as partículas no tempo $t = t_{n+1}$ através da Eq. (3.10), é possível obter a aceleração de todas as partículas, conforme o fluxograma da Fig. 3.2.

Embora tenha-se considerado que a aceleração não varie no intervalo Δt_n , a nova configuração geométrica das partículas garante que há uma aceleração no tempo $t = t_{n+1}$ diferente da aceleração no tempo $t = t_n$. Essa variação da aceleração, embora não seja considerada no cálculo de $\vec{x}^{(n+1)}$, será considerada no cálculo da velocidade em $t = t^{(n+1)}$, tal que:

$$\vec{v}^{(n+1)} = \vec{v}^{(n)} + \Delta \vec{v}^{(n+1)} \quad (3.11)$$

onde $\Delta \vec{v}^{(n+1)}$ representa a variação da velocidade no intervalo de tempo Δt_n e, considerando uma variação linear da aceleração, tem igual valor a área A , conforme a Fig. 3.4, ou seja:

$$\Delta \vec{v}^{(n+1)} = \int_{t_n}^{t_{n+1}} \vec{a}(t') dt' = A = \frac{1}{2} \Delta t_n (\vec{a}_i^{(n)} + \vec{a}_i^{(n+1)}) \quad (3.12)$$

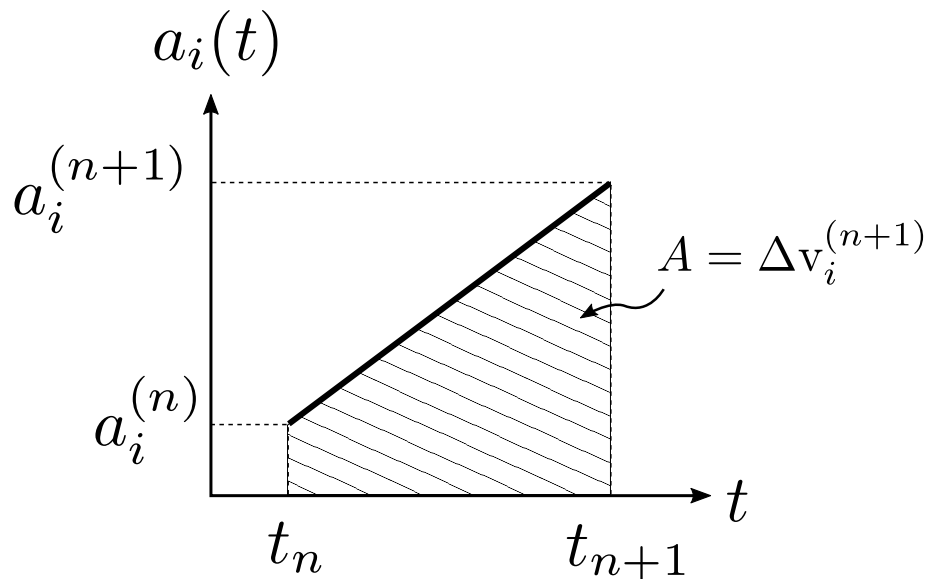


Figura 3.4: Variação da velocidade no intervalo Δt_n

Um fluxograma que resume todo o processo de solução será apresentado a seguir.

3.3 Fluxograma do Processo de Solução

Conforme visto na seção anterior, dado um tempo qualquer $t = t_n$ no qual são conhecidas as posições, as velocidades e as acelerações de todas as partículas, é

possível determinar as novas posições das partículas após um intervalo de tempo infinitesimal Δt_n através da Eq. (3.10). As acelerações das partículas no novo tempo $t = t_{n+1}$ podem ser determinadas através da Eq. (3.3), já as velocidades podem ser calculadas pela Eq. (3.11).

O fluxograma do processo de solução desde o tempo inicial $t = 0$ até o tempo estabelecido t_{max} está apresentado na Fig. 3.5. Nesta figura é possível notar que a aceleração no primeiro passo de tempo é calculada em função das posições e das velocidades das partículas definidas na condição inicial do problema e que há um processo iterativo até que o tempo discretizado, t_n , seja maior do que o tempo final estabelecido, t_{max} .

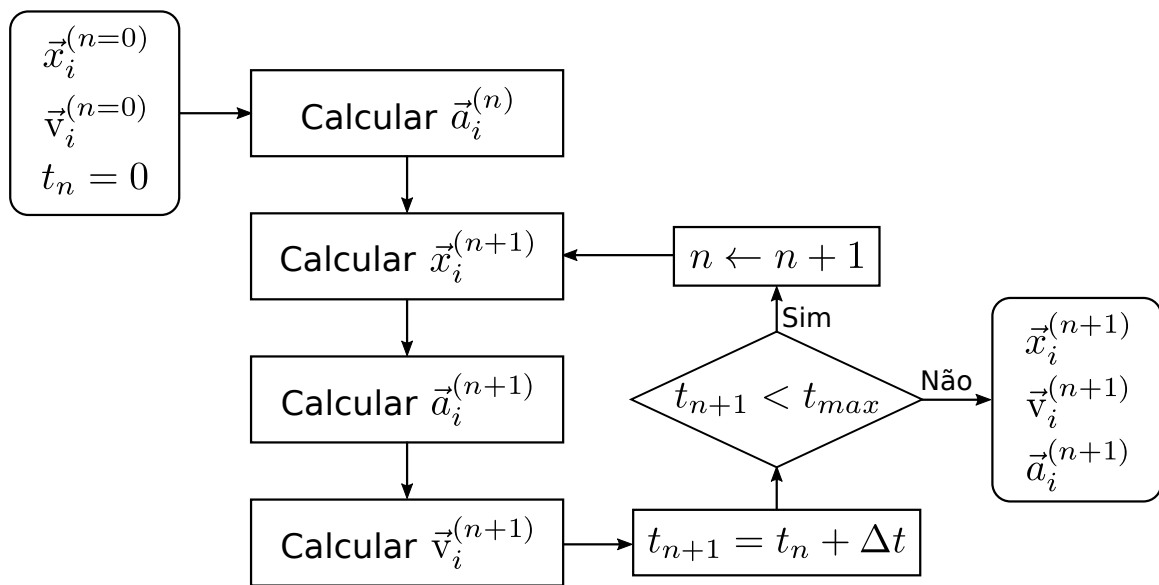


Figura 3.5: Fluxograma do processo de solução

3.4 Modelagem Computacional

Para representar a modelagem computacional utilizada serão apresentados alguns diagramas UML de classes que representam as estruturas criadas no simulador de partículas PyParticles.

Para representar as partículas criou-se uma classe *Partcile* que armazena todas as funcionalidades e características que uma partícula deve ter. Além desta classe, existe a classe *Particles* que representa o conjunto de partículas e a classe *Particles_ByPass* que é um tipo especial de partícula. Estas classes e suas interações estão representadas na Fig. 3.6.

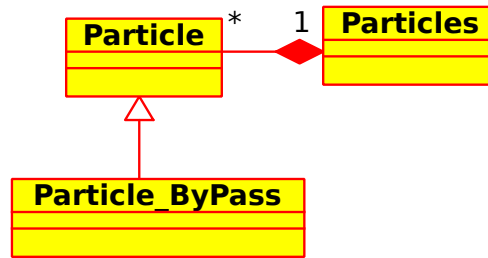


Figura 3.6: Diagrama de classes para as partículas

A classe *Particle_Bypass* representa um tipo de partícula que não é influenciada por outras partículas, porém influencia as demais. Desta forma, é possível simular uma compressão de partículas “normais” colocando-se partículas “*by pass*” ao redor e fazendo-se com que elas movam-se em direção às partículas “normais”.

Além das partículas, existem as classes de fronteira: *Boundary*. O conjunto de fronteiras é representado pela classe *Boundaries* e um tipo especial de fronteira é a plana, denominada por *Boundary_Plane*. Estas classes e suas interações estão representadas na Fig. 3.7.

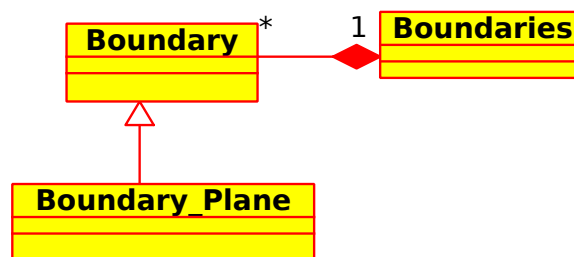


Figura 3.7: Diagrama de classes para as fronteiras

Para representar os contatos de forma genérica foi criada a classe *Contact*, porém para os contatos entre partículas foi criada a classe *Contact_Particle* e para o contato partícula-fronteira foi criada a classe *Contact_Boundary*. A classe *Contacts* representa o conjunto de contatos. Estas classes e suas interações estão representadas na Fig. 3.8.

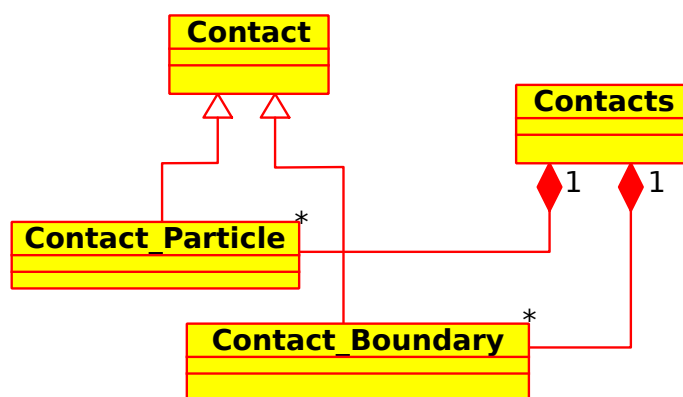


Figura 3.8: Diagrama de classes para os contatos

Para representar as forças de forma genérica foi criada a classe *Force* e, de forma específica, as classes: *Force_Constant*, para representar forças constante; *Force_Shear*, para forças elásticas cisalhantes; *Force_Attractive*, para forças elásticas normais de atração; e *Force_Repulsive*, para forças elásticas normais de repulsão. Estas classes e suas interações estão representadas na Fig. 3.9.

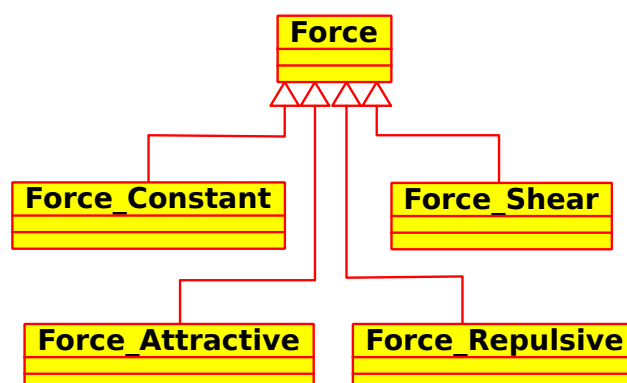


Figura 3.9: Diagrama de classes para as forças

A interação entre o simulador (classe *PyParticles*) e os arquivos de entrada e saída é realizada pela classe *IOControl*. Já a visualização do resultado é realizada pela classe *PyParticle_View* através de um encapsulamento do programa de visualização gráfica ParaView 5.0.1. A relação entre estas classes e as apresentas anteriormente está exposta na Fig. 3.10.

O código desenvolvido pode ser encontrado no site [github.com](https://github.com/bismarckjunior/PyParticles) pelo caminho:

[www.github.com/bismarckjunior/PyParticles](https://github.com/bismarckjunior/PyParticles)

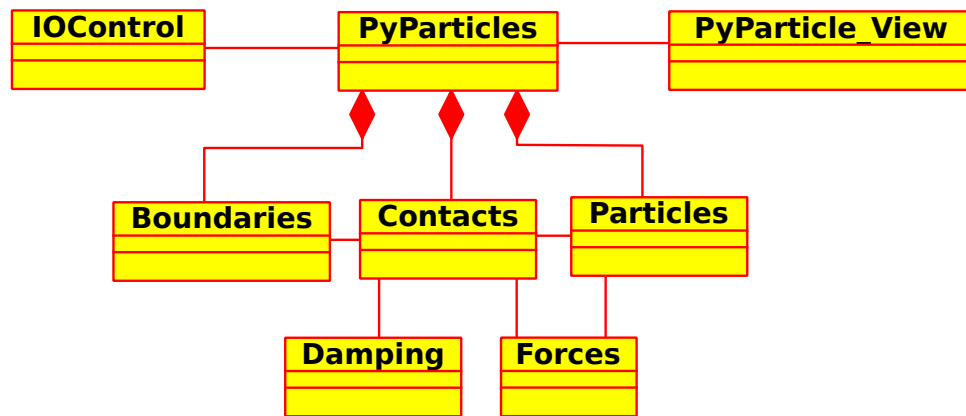


Figura 3.10: Diagrama de classes principal do simulador PyParticles