

Ruan Cardoso Comelli

**FUNDAMENTOS DO MÉTODO DE ELEMENTOS DISCRETOS (DEM) E SUA
APLICAÇÃO À SIMULAÇÃO DA DINÂMICA DE PARTÍCULAS ESFÉRICAS**

Florianópolis

2017

Ruan Cardoso Comelli

**FUNDAMENTOS DO MÉTODO DE ELEMENTOS DISCRETOS (DEM) E SUA
APLICAÇÃO À SIMULAÇÃO DA DINÂMICA DE PARTÍCULAS ESFÉRICAS**

Trabalho apresentado ao Curso de Graduação em Engenharia Mecânica da Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do título de Engenheiro Mecânico.

Universidade Federal de Santa Catarina
Curso de Graduação em Engenharia Mecânica

Orientador: Prof. António Fábio Carvalho da Silva, Dr. Eng.

Florianópolis
2017

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Comelli, Ruan Cardoso

Fundamentos do Método de Elementos Discretos (DEM) e
sua aplicação à simulação da dinâmica de partículas esféricas /
Ruan Cardoso Comelli ; orientador, António Fábio Carvalho
da Silva, 2017.

127 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Centro Tecnológico,
Graduação em Engenharia Mecânica, Florianópolis, 2017.

Inclui referências.

1. Engenharia Mecânica. 2. Método de Elementos
Discretos. 3. Dinâmica de partículas esféricas. 4. Solução de
equações de movimento. I. Silva, António Fábio Carvalho da.
II. Universidade Federal de Santa Catarina. Graduação em
Engenharia Mecânica. III. Título.

Ruan Cardoso Comelli

**FUNDAMENTOS DO MÉTODO DE ELEMENTOS DISCRETOS (DEM) E SUA
APLICAÇÃO À SIMULAÇÃO DA DINÂMICA DE PARTÍCULAS ESFÉRICAS**

Este Trabalho de Conclusão de Curso foi julgado adequado para a obtenção do título de Engenheiro Mecânico e aprovado em sua forma final pela Comissão examinadora e pelo Curso de Graduação em Engenharia Mecânica da Universidade Federal de Santa Catarina.

Florianópolis, 15 de dezembro de 2017.

**Prof. Carlos Enrique Niño Bohórquez,
Dr. Eng.
Coordenador do Curso**

Comissão Examinadora:

**Prof. António Fábio Carvalho da Silva,
Dr. Eng.
Orientador**

Prof. Saulo Güths, Dr. Eng.

Giovani Cerbato, Me. Eng.

À minha família, por tudo o que sou.

Agradecimentos

Agradeço a toda a equipe do SINMEC por fazer deste laboratório um ótimo ambiente de trabalho. Ao professor Maliska por me conceder essa oportunidade. Ao Gustavo e ao Giovani por terem me auxiliado tanto, especialmente no início. À Tatiane por resolver todos os problemas burocráticos, que são sempre os mais difíceis. Ao Felipe, ao Hermínio, à Josi, ao Conrado, ao Vitão, ao Rafa e a todos os demais integrantes do SINMEC pelas conversas e risadas. À Taisa por me ajudar em várias etapas deste trabalho e, pacientemente, responder às minhas perguntas incessantes.

Ao professor Fábio por ter aceito a árdua tarefa de me orientar, pela revisão incansável do texto e pelas discussões, sempre produtivas, que impulsionaram o desenvolvimento deste trabalho.

Aos professores Raphael, Gilles, Melissa e Milton e ao coordenador do Programa Avançado de Matemática (PAM), Mário César Zambaldi, pelo curso de excelência. As habilidades desenvolvidas nesse curso foram fundamentais para a minha formação.

Também não posso deixar de me sentir grato a toda a comunidade on-line de desenvolvedores, programadores e estudantes sempre dispostos a esclarecer qualquer tipo de dúvida. Em especial, à Stack Exchange, sem a qual eu provavelmente não teria aprendido o suficiente sobre programação e L^AT_EX para concluir este trabalho.

A todos os amigos que caminharam comigo pela trajetória da graduação. Dentre eles, gostaria de destacar o Alex, o Henrique, o Sérgio, o Dalla Vecchia e o Guesser por terem compartilhado os momentos de alegria e de sofrimento e por passarem tantas noites acordados comigo fazendo trabalhos. Em especial, ao Guesser por todas as discussões sobre os mais diversos assuntos, pelos lanches sempre pontuais e por, além de ser um grande amigo, participar em todas as etapas deste TCC.

E, acima de tudo, sou imensamente grato à minha família pelo amor e pelo carinho. Às minhas irmãs, Natália e Beatriz, por me ajudarem a exercitar a sabedoria e a paciência e por me ensinarem tanto, cada uma ao seu modo. Ao meu pai, Ricardo, por todas as discussões filosóficas, pelo ótimo gosto musical e por ser meu melhor amigo. À minha mãe, Morgana, pelo apoio incondicional, por me dar forças mesmo nos momentos mais difíceis e pelo seu amor infinito. À Puca e à Poly pela alegria sempre que chego em casa. Aos meus avós, tios e primos por todos os momentos juntos. À Pity e ao meu tio Thiago por haverem me ensinado que os laços familiares se sobrepõem aos laços de sangue e por terem tocado meu coração e ali deixado sua marca em sua passagem pela vida.

*“What is better: to be born good,
or to overcome your evil nature
through great effort?”*

Paarthurnax (Charles Martinet, 2011)

Resumo

Este trabalho apresenta os fundamentos do Método de Elementos Discretos (DEM), uma família de métodos computacionais amplamente aplicados na simulação da dinâmica de sistemas de partículas, e a implementação de uma biblioteca de suporte à geração de simuladores, com sua subsequente validação numérica. Os métodos de elementos discretos têm como propósito a solução das equações de movimento de partículas que interagem entre si por meio de forças de colisão. As partículas são consideradas corpos rígidos aos quais se aplicam as leis de Euler para o movimento, e as forças que atuam entre elas são dadas por modelos matemáticos. A solução das equações de movimento é feita por meio de algoritmos de integração, dentre os quais destaca-se o algoritmo de Gear. Ademais, é detalhado o método numérico obtido a partir da formulação matemática, assim como suas principais etapas: a inicialização do sistema, a solução das equações de movimento, a exportação de dados, a avaliação do critério de parada do laço de simulação e a finalização do programa. Consideram-se, também, ferramentas adicionais como métodos de monitoramento de vizinhança, condições de contorno, restrições de movimento e, de maneira simplificada, o acoplamento do DEM com outras famílias de métodos, como o Método de Elementos Finitos (FEM) e a Fluidodinâmica Computacional (CFD). Com base nisso, foi implementada uma biblioteca computacional de suporte ao desenvolvimento de simuladores DEM, permitindo a inserção de modelos de interação arbitrários e de elementos de geometria e propriedades físicas quaisquer. A implementação foi feita em linguagem C++ aplicando programação orientada a objetos e a técnica de metaprogramação com templates. Por fim, um simulador é desenvolvido e quatro casos de teste são resolvidos numericamente. Os resultados simulados são comparados com as soluções analíticas dos problemas, e, com isso, é atestada a corretude do algoritmo.

Palavras-chave: Método de Elementos Discretos. Dinâmica de partículas esféricas. Solução de equações de movimento. Algoritmo de Gear.

Abstract

This study presents the fundamentals of the Discrete Element Method (DEM), a family of computer methods widely used to simulate particle system dynamics, the implementation of a support library for the generation of simulators and its subsequent numerical validation. Discrete element methods have as purpose the solution to the motion equations of particles that interact according to collision forces. Particles are considered as rigid bodies to which Euler's laws of motion are applied, and the forces acting between them are given by mathematical models. The solution of the motion equations is done by means of integration algorithms, among which stands out Gear algorithm. In addition, the numerical method derived from the mathematical formulation is detailed as well as its main steps: the system initialization, the solution to the motion equations, the data output, the evaluation of the simulation loop stop criterion and the program termination. Furthermore, additional tools are considered, some of which are neighborhood monitoring methods, boundary conditions, motion constraints and, in a summarised manner, the coupling between DEM and other numerical methods such as the Finite Element Method (FEM) and Computational Fluid Dynamics (CFD). On this basis, a computational library was implemented to support the design of DEM simulators, allowing the definition of new interaction models and elements with arbitrary geometry and physical properties. The implementation was done in C++ applying object-oriented programming and template metaprogramming. Finally, a simulator was developed and four test cases were numerically solved. The simulation results were compared to the respective analytical solutions, attesting the algorithm correctness.

Keywords: Discrete Element Method. Spherical particle dynamics. Solution to motion equations. Gear algorithm.

Listas de ilustrações

Figura 1.1 – Comparaçāo entre experimento e simulaçāo de pílulas no interior de um misturador rotacionando a 40 rpm.	28
Figura 1.2 – Comparaçāo entre experimento e simulaçāo da mistura de esferas de vidro no interior de um misturador em função do númerode revoluções do compartimento.	29
Figura 1.3 – Exemplo de associaçāo de partículas simples para a obtenção de geometrias mais complexas.	31
Figura 1.4 – Fluxograma simplificado do Método de Elementos Discretos.	32
Figura 2.1 – Ângulo de rotação de uma partícula com relação à posição de referência em simulações bidimensionais.	42
Figura 2.2 – Representação dos raios, das posições, e das velocidades das partículas, da superposição, dos versores normais, dos vetores radiais, do plano de contato e da reta normal.	47
Figura 2.3 – Velocidade relativa no ponto de contato.	50
Figura 2.4 – Representação esquemática do modelo de amortecimento linear.	53
Figura 2.5 – Representação da colisão entre partículas. Na primeira linha, a avaliação das forças segundo os modelos não corrigidos. Na segunda linha, a colisão com forças limitadas à repulsão.	57
Figura 3.1 – Fluxograma do funcionamento geral do DEM.	69
Figura 3.2 – Condição de parede reflexiva.	78
Figura 3.3 – Restrição ao movimento de subelementos de uma partícula composta pela ação de molas.	79
Figura 3.4 – Processo de fragmentação de uma partícula. O elemento fragmentado é substituído por duas novas partículas com uma elevada superposição.	80
Figura 3.5 – Divisão do domínio da simulação em 16 regiões. Os segmentos de reta indicam a fronteira dos domínios locais dos processadores. Partículas localizadas próximo às fronteiras são representadas em cor preta.	80
Figura 3.6 – Abordagens <i>one-way</i> e <i>two-way</i> para o acoplamento no Método de Elementos Discretos Estendido.	82
Figura 3.7 – Abordagens para o contato entre partículas, representantes do DEM, e corpos contínuos, em que se utiliza o FEM.	83
Figura 4.1 – Componentes do programa implementado.	87
Figura 4.2 – Ilustração do problema do lançamento oblíquo.	89
Figura 4.3 – Solução para a altura da partícula no problema de lançamento oblíquo ¹	90
Figura 4.4 – Erro da simulação com relação à solução numérica para a posição e a velocidade da partícula do problema do lançamento oblíquo.	91

Figura 4.5 – O problema da esfera quicando.	92
Figura 4.6 – Solução numérica da altura da partícula e comparação entre as soluções numérica e analítica para o coeficiente de restituição no caso conservativo do problema da esfera quicando.	94
Figura 4.7 – Solução para a altura da partícula no caso dissipativo do problema da esfera quicando.	95
Figura 4.8 – Energia cinética e energia mecânica da partícula no caso dissipativo do problema da esfera quicando.	96
Figura 4.9 – Coeficiente de restituição resultante da colisão dissipativa do problema da esfera quicando.	96
Figura 4.10–O problema da colisão entre esferas.	97
Figura 4.11–Solução para a posição horizontal e a velocidade horizontal, para quantidade de movimento linear, para a força de contato, para a energia cinética e para o coeficiente de restituição da simulação conservativa do problema das esferas colidentes.	99
Figura 4.12–Solução da posição, da velocidade, da quantidade de movimento linear, da força de contato horizontal, da energia cinética e do coeficiente de restituição do par de partículas colidentes na simulação dissipativa sem rotação.	101
Figura 4.13–Erro máximo do coeficiente de restituição em função do passo de tempo no caso dissipativo do problema das esferas colidentes sem rotação.	102
Figura 4.14–Solução das posições horizontal e vertical, das velocidades horizontal e vertical, e da velocidade angular das partículas no caso dissipativo do problema da colisão de esferas com rotação.	104
Figura 4.15–Solução numérica das quantidades de movimento linear horizontal, linear vertical e angular para o caso dissipativo do problema das esferas colidentes com rotação.	105
Figura 4.16–Energias rotacional, translacional e cinética e coeficiente de restituição no caso dissipativo do problema das esferas colidentes com rotação.	106
Figura 4.17–O problema da queda com arrasto.	107
Figura 4.18–Soluções para a altura da partícula, sua velocidade vertical, sua energia cinética e sua energia mecânica da partícula no problema de queda com arrasto.	109

Lista de tabelas

Tabela 2.1 – Constantes corretoras para o algoritmo de Gear em função da ordem k da maior derivada considerada e da ordem p da equação diferencial.	64
Tabela 4.1 – Solução do problema do lançamento oblíquo.	89
Tabela 4.2 – Parâmetros para o problema de lançamento oblíquo.	90
Tabela 4.3 – Parâmetros para o caso conservativo do problema da esfera quicando.	93
Tabela 4.4 – Energia mecânica e coeficiente de restituição resultantes do caso conservativo do problema da esfera quicando.	94
Tabela 4.5 – Parâmetros para o caso dissipativo do problema da esfera quicando.	95
Tabela 4.6 – Parâmetros para o caso conservativo do problema da colisão entre esferas.	98
Tabela 4.7 – Coeficiente de restituição resultante do caso conservativo do problema das esferas colidentes.	98
Tabela 4.8 – Parâmetros para o caso dissipativo sem rotação do problema da colisão entre esferas.	100
Tabela 4.9 – Parâmetros para o caso dissipativo do problema da colisão entre esferas considerando rotações.	103
Tabela 4.10–Energia mecânica e coeficiente de restituição resultantes do caso dissipativo do problema da colisão de esferas com rotação.	103
Tabela 4.11–Parâmetros para o problema da queda com arrasto.	108

Lista de listagens

Listagem 3.1 Pseudocódigo para a solução das equações de movimento de partículas esféricas por meio do algoritmo de Gear.	73
Listagem 3.2 Pseudocódigo para a exportação de dados.	75
Listagem 3.3 Pseudocódigo para a condição de parada do laço da simulação. .	75
Listagem 3.4 Pseudocódigo para a aplicação da Terceira Lei de Newton.	77
Listagem B.1 Arquivo de entrada para o problema do lançamento oblíquo. . . .	125

Listas de símbolos

Abreviaturas e siglas

CFD Fluidodinâmica computacional (*Computational Fluid Dynamics*)

DEM Método de Elementos Discretos (*Discrete Element Method*)

EbFVM Método de Volumes Finitos baseado em Elementos (*Element-based Finite Volume Method*)

FEM Método de Elementos Finitos (*Finite Element Method*)

CAD Projeto Assistido por Computador (*Computer Aided Design*)

XDEM Método de Elementos Discretos Estendido (*Extended Discrete Element Method*)

Símbolos Gregos

ω	Vetor velocidade angular	[rad/s]
ε	Coeficiente de restituição	
γ	Constante de amortecimento	[N s/m]
ζ	Elongação	[m]
μ	Coeficiente de atrito	
ν_f	Viscosidade cinemática	[m ² /s]
φ	Função orientação	
ξ_{ij}	Superposição entre os elementos i e j	[m]
ν	Coeficiente de Poisson	
ϱ	Vetor radial	[m]
ρ	Massa específica	[kg/m ³]
κ^t	Constante elástica tangencial	[N/m]

Símbolos Romanos

\mathbf{L}	Vetor quantidade de movimento angular	[kg m ² /s]
\mathcal{B}	Conjunto de elementos de contorno	
\mathcal{T}_{ij}	Plano de contato entre os elementos i e j	
$c_{p,q}$	Constante corretora de ordem q para uma equação de ordem p	
n	Contador	
d	Distância	[m]
C_D	Coeficiente de arrasto	
Δt	Passo de tempo	[s]
k	Constante elástica	[N/m]
Y	Módulo de elasticidade	[Pa]
\mathcal{E}_j	Elemento j	
\mathcal{E}	Conjunto de elementos	
Erro	Erro	
($\mathbf{i}, \mathbf{j}, \mathbf{k}$)	Base canônica do espaço tridimensional	
$\underline{\mathbf{A}}_k$	Matriz de extração de ordem k	
\mathbf{F}	Vetor força	[N]
\mathbf{g}	Vetor gravidade	[m/s ²]
$\underline{\mathbf{I}}$	Matriz identidade	
t	Instante de tempo	[s]
E_C	Energia cinética	[J]
\mathbf{p}	Vetor quantidade de movimento linear	[kg m/s]
m	Massa	[kg]
$\underline{\mathbf{J}}$	Matriz de inércia	[kg m ²]
Erro_{\max}	Erro máximo	
E_M	Energia mecânica	[J]

J	Momento de inércia	[kg m ²]
\mathcal{N}	Vizinhança	
A	Constante dissipativa	
N_{ij}	Reta normal da colisão entre os elementos i e j	
$\hat{\mathbf{e}}^n$	Versor normal	
N	Quantidade	
\mathcal{P}_i	Partícula i	
\mathcal{P}	Conjunto de partículas	
$\hat{\mathbf{n}}$	Versor normal a um plano	
\mathbf{r}	Vetor posição ou ponto de um plano	[m]
E_P	Energia potencial	[J]
R	Raio	[m]
\mathbf{v}	Velocidade relativa no ponto de contato	[m/s]
Re	Número de Reynolds	
E_R	Energia rotacional	[J]
$\underline{\mathbf{A}}$	Matriz de rotação	
\mathcal{S}	Conjunto de partículas esféricas	
$\hat{\mathbf{e}}^t$	Versor tangencial	
k	Ordem de extração	
$\hat{\mathbf{J}}$	Tensor de inércia	[kg m ²]
\boldsymbol{M}	Torque	[N m]
E_T	Energia translacional	[J]
\mathcal{U}	Conjunto universo	

Sobrescritos

\star Direções principais

ext Externo

f Final

i Inicial

n Direção normal

t Direção tangencial

Subscritos

i Elemento ou partícula i

ij Entre os elementos i e j ou de i para j

c Crítico

corr Correção

pr Predição

R Resultante

Sumário

1	INTRODUÇÃO	27
1.1	Motivação	27
1.2	Sistemas de Partículas	29
1.3	O Método de Elementos Discretos	30
1.4	Revisão Bibliográfica	32
1.5	Objetivos e Contribuições	33
1.6	Organização do Trabalho	35
2	MODELO MATEMÁTICO	37
2.1	Sistemas de Partículas	37
2.2	Equações de Movimento	38
2.2.1	Primeira Lei de Euler	39
2.2.2	Segunda Lei de Euler	40
2.2.2.1	Simulação Bidimensional	41
2.2.2.2	Rotação de Partículas Esféricas	43
2.2.2.3	Rotação Geral em Três Dimensões	43
2.2.3	Solução das Equações de Movimento	45
2.3	Modelos de Força de Colisão	46
2.3.1	Parâmetros de Cálculo	46
2.3.2	Modelos de Força Normal	53
2.3.3	Modelos de Força Tangencial	58
2.4	Algoritmo de Gear	60
2.4.1	Etapa de Predição	60
2.4.2	Correção	62
3	O MÉTODO DE ELEMENTOS DISCRETOS	65
3.1	Características Gerais do Método	65
3.1.1	Discretização Temporal	66
3.1.2	Solução das Equações	66
3.1.3	Elementos da Simulação	67
3.1.4	Algoritmo Geral do DEM	68
3.2	Etapas do Algoritmo	68
3.2.1	Inicialização do Sistema de Partículas	68
3.2.2	Solução das Equações	70
3.2.3	Exportação de Dados	74
3.2.4	Condição de Parada	74

3.2.5	Finalização	75
3.3	Monitoramento da Vizinhança	76
3.4	Limitação do Passo de Tempo	77
3.5	Condições de Contorno e Restrições	78
3.6	Outros Aspectos do Método	79
3.7	Acoplamento com Outros Métodos	81
4	IMPLEMENTAÇÃO COMPUTACIONAL E RESULTADOS	85
4.1	Implementação Computacional	85
4.2	Resultados e Validação	87
4.3	Lançamento Oblíquo	88
4.4	Esfera Quicando	91
4.4.1	Caso Conservativo	93
4.4.2	Caso Dissipativo	94
4.5	Colisão entre Esferas	96
4.5.1	Caso Conservativo	97
4.5.2	Caso Dissipativo	100
4.5.2.1	Sem Rotação	100
4.5.2.2	Com Rotação	102
4.6	Queda com Arrasto	106
5	CONCLUSÃO	111
5.1	Sumário	111
5.2	Conclusões	112
5.3	Recomendações para Trabalhos Futuros	114
	REFERÊNCIAS	117
	APÊNDICES	121
	APÊNDICE A – CASO GERAL DO ALGORITMO DE GEAR . . .	123
	APÊNDICE B – ARQUIVO DE ENTRADA PARA O PROBLEMA DO LANÇAMENTO OBLÍQUO	125

1 Introdução

1.1 Motivação

A compreensão acerca do comportamento de sistemas de partículas e de materiais granulados cresceu enormemente nas últimas décadas devido aos esforços da ciência e da engenharia. Tais sistemas são amplamente encontrados em processos industriais e nas ciências naturais (SCHWAGER; PÖSCHEL, 2005) tais como matemática aplicada, física de matéria condensada, geomecânica, agricultura, engenharia química, engenharia civil e engenharia mecânica (AI et al., 2010).

Esses conhecimentos são aplicados na previsão do comportamento mecânico de geomateriais, como rochas, e da propagação de trincas e fraturas em sólidos, na simulação da produção de fármacos, alimentos, detergentes e cosméticos, no desenvolvimento de novos materiais e na modelagem da fragmentação, sedimentação, granulação, escavação, transporte e armazenamento de grãos e do escoamento em máquinas extrusoras (DONZÉ; RICHEFEU; MAGNIER, 2009; GALLAS et al., 1996; MARIGO; STITT, 2015).

Sendo assim, surge a necessidade de se compreender a mecânica das partículas.

A análise de sistemas de partículas, contudo, costuma ser bastante complexa. Partículas podem interagir umas com as outras e com o meio em que estão imersas. As interações podem ser forças de contato, forças de corpo, trocas de calor, trocas de carga elétrica, entre outras. Ainda, a quantidade de partículas presentes no sistema de interesse pode facilmente alcançar a ordem dos milhões (SCHWAGER; PÖSCHEL, 2005). Além dessas dificuldades, os problemas estudados geralmente assumem geometrias distintas, tais quais a mistura de substâncias para a produção de fármacos ou o transporte de carga granular.

Com o propósito de executar tais análises, existem três abordagens principais: a experimental, a analítica e a numérica.

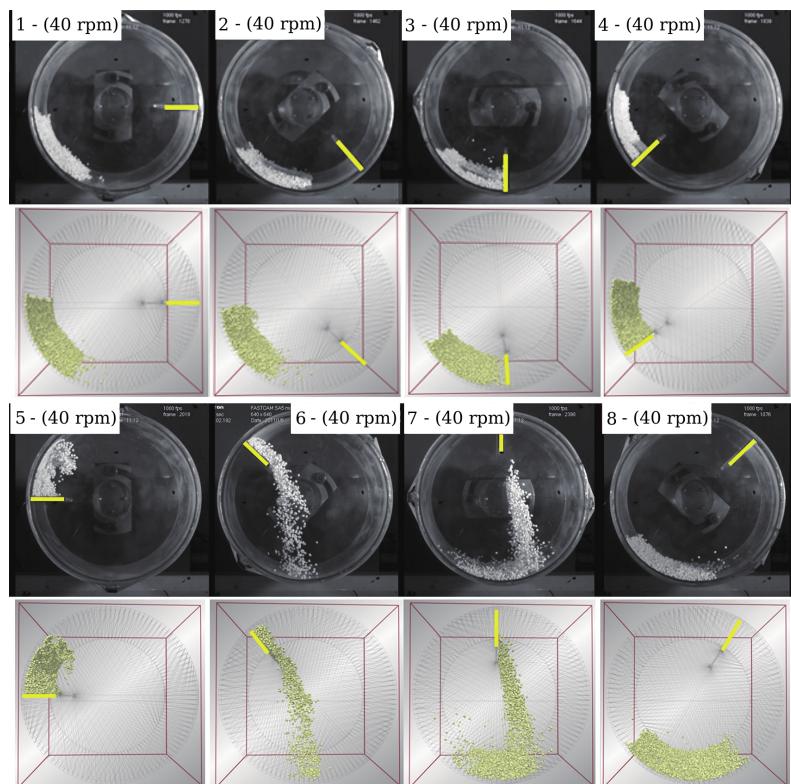
No método experimental, procura-se caracterizar os fenômenos físicos através da sua reprodução em ambientes controlados. A grande vantagem desse método é o fato de lidar com a configuração real do sistema estudado, e dessa análise resultam hipóteses, equações e correlações para as diversas grandezas físicas envolvidas. Essa abordagem, no entanto, resulta em altíssimo custo e muitas vezes é limitada por questões de segurança ou pela dificuldade de reprodução do sistema real. A experimentação é aplicada para validar modelos físicos e matemáticos (MALISKA, 2004).

A abordagem analítica, por sua vez, busca a obtenção de soluções analíticas para

o problema. A vantagem de se obterem soluções em forma fechada é seu baixíssimo custo de computação quando comparado aos outros métodos. O método analítico, porém, frequentemente depende de hipóteses simplificativas e de geometrias e condições de contorno simples, o que acaba por reduzir a aplicabilidade de seus resultados. Geralmente, utilizam-se soluções analíticas para validar métodos numéricos e auxiliar na busca de algoritmos mais robustos ([MALISKA, 2004](#)).

A fim de conciliar exatidão e eficiência, os métodos numéricos despontam como uma poderosa ferramenta capaz de resolver problemas complexos, com geometrias complicadas e grande número de partículas, alcançando, apesar disso, uma elevada rapidez e um baixíssimo custo ([MALISKA, 2004](#)).

Figura 1.1 – Comparação entre experimento e simulação de pílulas no interior de um misturador rotacionando a 40 rpm.



Fonte: Adaptado de [Marigo e Stitt \(2015\)](#).

As figuras 1.1 e 1.2 ilustram a capacidade dos métodos numéricos. A primeira delas é uma comparação feita por [Marigo e Stitt \(2015\)](#) entre pílulas movendo-se no interior de um tambor rotativo experimentalmente e em uma simulação. A segunda figura representa uma comparação feita por [Rantanen e Khinast \(2015\)](#) entre a mistura de esferas de vidro em um misturador e uma simulação correspondente. As esferas são idênticas com exceção de sua cor, o que permite uma melhor avaliação dos resultados. Como pode ser observado, embora as simulações não reproduzam *exatamente* a posição das partículas, elas representam com grande fidelidade o sistema estudado.

Figura 1.2 – Comparação entre experimento e simulação da mistura de esferas de vidro no interior de um misturador em função do número de revoluções do compartimento.



Fonte: Adaptado de Rantanen e Khinast (2015).

Sendo assim, o estudo de sistemas de partículas conta principalmente com a simulação numérica como método de análise.

1.2 Sistemas de Partículas

A definição de partícula não é um consenso entre autores, cada um adotando o conceito mais adequado à sua aplicação. Segundo Sampaio (2017), comprehende-se por *partícula* um objeto ao qual são atribuídas propriedades físicas, mas cuja estrutura interna é desconsiderada. Cada partícula possui também uma função posição, que descreve a posição de seu centro de massa em função do tempo, e uma função orientação, que determina a posição dos seus eixos principais, sendo que essas duas definições são tomadas com relação a um sistema de referência fixo no espaço.

Por sua vez, um *sistema de partículas* é um conjunto de partículas que interagem entre si. Conforme citado na [seção 1.1](#), essas interações podem ser de diversas naturezas, das quais se destacam as forças de colisão.

O aspecto dinâmico dos sistemas de partículas é determinado pelas leis de Euler para o movimento:

$$\mathbf{F}_R = m \cdot \ddot{\mathbf{r}}, \quad (1.1)$$

$$\mathbf{M}_R = \hat{\mathbf{J}} \cdot \dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (\hat{\mathbf{J}} \cdot \boldsymbol{\omega}), \quad (1.2)$$

em que \mathbf{r} , $\boldsymbol{\omega}$, m e $\hat{\mathbf{J}}$ são, respectivamente, a função posição, a função velocidade angular, a massa e o tensor momento de inércia de uma partícula qualquer, enquanto \mathbf{F}_R e \mathbf{M}_R são o vetor força resultante e o vetor torque resultante sobre a mesma.

É na solução das equações (1.1) e (1.2) que consiste a análise acerca dos sistemas de partículas. Essas equações, porém, são de difícil resolução já que a força e o torque resultantes sobre a partícula são oriundos de sua interação com os demais elementos do sistema e são, portanto, funções das propriedades físicas, posições, velocidades, orientações e velocidades angulares dos interagentes.

Em vista disso, é necessário o uso de métodos para a resolução dessas equações, e nessa situação destaca-se o Método de Elementos Discretos.

1.3 O Método de Elementos Discretos

Segundo [Sampaio \(2017\)](#), o Método de Elementos Discretos, ou DEM (*Discrete Element Method*), refere-se a uma família de métodos computacionais que consideram deslocamentos e rotações de corpos discretos e reconhecem contatos entre elementos automaticamente à medida que os cálculos são executados. Devido a essa característica, o DEM representa um conjunto de técnicas apropriadas para a simulação do comportamento dinâmico de sistemas de partículas.

De acordo com [Cundall e Strack \(1979\)](#), os cálculos são executados explicitamente, alternando entre a aplicação das leis de Euler para a determinação da posição e da orientação das partículas e a aplicação de relações força-deslocamento para a computação das forças e torques atuantes.

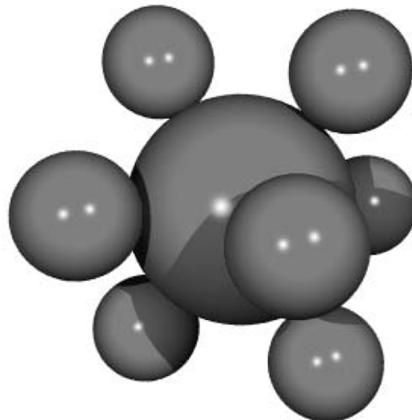
Geralmente, algoritmos de elementos discretos levam em conta a hipótese de partículas rígidas. Uma consequência disso é que, se uma partícula está em contato com outras duas, as interações ocorrem de maneira independente. Com isso, os contatos podem ser computados para cada par de partículas interagentes como se estivessem isoladas das demais.

Cada elemento da simulação pertence a um *tipo*. O tipo de um elemento é usado para determinar a sua geometria e o seu comportamento na simulação. É comum, por exemplo, definir-se um tipo de partícula *fixa*. Partículas desse tipo representam elementos de fronteira, com posição constante e conhecida desde o início da simulação.

Por sua vez, a geometria das partículas determina quais os modelos de interação que podem ser aplicados. Em geral, o uso de formas geométricas simples permite a aplicação de modelos também simples, resultando em um custo computacional menor quando comparado às geometrias mais complexas. Certamente, a geometria de partícula necessária depende do sistema que se deseja estudar e do grau de representatividade requisitado. A forma mais simples é a esférica. Para a obtenção de formas mais complexas, [Schwager e Pöschel \(2005\)](#) apresentam uma técnica de associação de partículas, resultando em partículas compostas, como aquela representada na [figura 1.3](#). [Schwager e Pöschel](#)

(2005) também tratam de partículas poligonais, enquanto Sampaio (2017) demonstra a utilização de superelipsoides.

Figura 1.3 – Exemplo de associação de partículas simples para a obtenção de geometrias mais complexas.



Fonte: Schwager e Pöschel (2005, p. 84).

O Método de Elementos Discretos é intimamente ligado à Dinâmica Molecular, outra família de métodos computacionais. A distinção que geralmente se faz é que o DEM considera geometrias arbitrariamente complexas, exigindo um tratamento da identificação dos pontos de contato e do cálculo das rotações mais adequado.

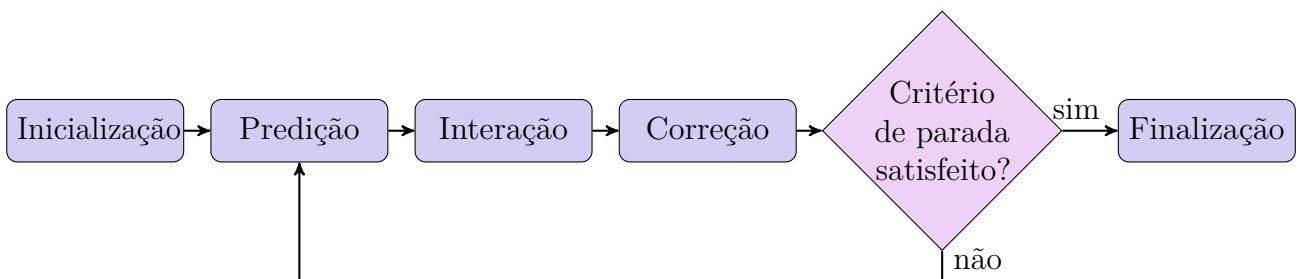
Segundo Schwager e Pöschel (2005), algoritmos do DEM, de forma simplificada, são compostos por algumas etapas principais, sendo elas:

1. *Inicialização*: Nesta etapa, os elementos da simulação são construídos, cada qual com tipo, geometria, propriedades físicas, posição e velocidade iniciais. Também são determinados os modelos de interação considerados e o critério de parada. Esses itens são geralmente definidos por um arquivo de entrada.
2. *Predição*: O cálculo de forças e torques atuantes sobre cada partícula depende da posição, da velocidade, da orientação e da velocidade angular dos elementos da simulação, que são desconhecidas no instante futuro. Sendo assim, é necessária uma estimativa, ou *predição*, para os valores dessas variáveis. Essa estimativa é feita na etapa de predição.
3. *Interação*: A etapa de interação se divide em duas subetapas:
 - a) Determinação dos pares de interação: Nem todas as partículas interagem em todos os instantes. Para o caso das forças de colisão, é necessário identificar todos os pares que estão em contato. Essa etapa costuma ser a mais custosa do método, e várias técnicas são propostas para minimizar esse custo.

- b) Computação das forças e torques entre os elementos da simulação com base nas coordenadas previstas na etapa 2.
4. *Correção*: Nesta etapa, valores corrigidos para as coordenadas (e as respectivas derivadas) das partículas são obtidos através da resolução das equações (1.1) e (1.2).
 5. *Finalização*: Caso o critério de parada previamente estabelecido seja atingido, o programa deve terminar sua execução. Caso contrário, a simulação deve avançar o instante de tempo e retornar ao passo 2.

Essas etapas podem ser representadas como na figura 1.4. Outros passos podem ser introduzidos no algoritmo, tais quais exportação de dados, iterações em cada passo de tempo e acoplamento com outros simuladores.

Figura 1.4 – Fluxograma simplificado do Método de Elementos Discretos.



Fonte: do Autor.

Também é possível acoplar o DEM a outras famílias de métodos, como FEM (Método de Elementos Finitos) e CFD (Fluidodinâmica Computacional). Nesses casos, além das equações de movimento para as partículas, são consideradas equações de conservação de massa, de energia, de carga elétrica, equações de estado, entre outras. Com isso, é possível computar transferência de calor, arrasto causado por fluidos e distribuição de tensões em sólidos. [Donzé, Richefeu e Magnier \(2009\)](#) e [Bićanić \(2007\)](#) apresentam alguns exemplos das diversas extensões que podem ser feitas ao DEM. [Chu e Yu \(2008\)](#) explicam e aplicam um modelo para interação entre um fluido e partículas com foco no arrasto, enquanto [Vargas e McCarthy \(2002\)](#) e [Simsek et al. \(2009\)](#) demonstram a aplicação de modelos de troca de calor. O acoplamento entre DEM e outras famílias é conhecido como XDEM, ou EDEM, o Método de Elementos Discretos Estendido.

1.4 Revisão Bibliográfica

A base matemática para os Métodos de Elementos Discretos foi fundada por Isaac Newton com a divulgação de suas leis de movimento. Entretanto, devido às dificuldades

mencionadas na [seção 1.1](#), como a quantidade de partículas e a natureza das equações diferenciais, a análise de sistemas de partículas só se tornou factível com o advento dos computadores.

O Método de Elementos Discretos foi primeiramente apresentado por [Cundall \(1971 apud SAMPAIO, 2017\)](#). Uma descrição do método pode ser encontrada em [Cundall e Strack \(1979\)](#), em que se consideram atrito e amortecimento na colisão das partículas. A presença de atrito provoca rotações, que devem ser solucionadas em conjunto com as posições. Nesse algoritmo, as velocidades eram preditas como constantes durante um passo de tempo.

O método foi ainda estendido com os trabalhos de [Haff e Werner \(1986\)](#), [Walton \(1983 apud HAFF; WERNER, 1986\)](#), [Gallas, Herrmann e Sokolowski \(1992\)](#), além de vários outros.

Dentre as principais referências sobre o DEM, destacam-se [Allen e Tildesley \(1987\)](#), [Schwager e Pöschel \(2005\)](#) e [Bićanić \(2007\)](#).

Além das publicações acadêmicas, existem diversos programas comerciais e de código aberto projetados para simulação com o DEM, cada qual com seus objetivos, vantagens e desvantagens. Dentre os simuladores comerciais, destacam-se o Rocky DEM, o EDEM e o PFD (*Particle Flow Code*). Dentre os principais programas de código aberto estão o Yade, o LIGGGHTS e o KRATOS. Algumas das características desses simuladores são o processamento em paralelo, a capacidade de customização, pós-processamento, geração de scripts, acoplamento entre diferentes físicas, uso de geometrias CAD e acoplamento com outros simuladores.

1.5 Objetivos e Contribuições

O SINMEC - Laboratório de Simulação Numérica em Mecânica dos Fluidos e Transferência de Calor, do Departamento de Engenharia Mecânica da Universidade Federal de Santa Catarina, tem como principal linha de pesquisa a aplicação do método de volumes finitos baseado em elementos (EbFVM) com malhas não estruturadas para a solução de problemas de mecânica de fluidos e transferência de calor.

Devido à importância do tema e à possibilidade de se estudar o acoplamento entre a dinâmica de partículas e os fenômenos de transporte, surgiu o interesse do laboratório em iniciar uma linha de pesquisa acerca do Método de Elementos Discretos Estendido utilizando o EbFVM para a solução de escoamentos. Para tanto, porém, é necessária, primeiramente, uma compreensão acerca do DEM.

Sendo assim, este trabalho objetiva iniciar essa linha de pesquisa no SINMEC, desenvolvendo um simulador que seja de fácil extensão, permitindo que novos modelos

de interação, tipos de partículas e algoritmos de solução sejam introduzidos à medida que novos estudos sejam feitos. Para que isso se tornasse possível, este trabalho foi subdividido em três partes: primeiramente, implementar uma biblioteca contendo toda a lógica do funcionamento dos algoritmos DEM de forma generalizada, sem particularizar suas características; na segunda parte, gerar um simulador que, tomando a biblioteca como base, especifique os tipos de partícula, as interações e os métodos de solução; e, por fim, executar simulações no simulador e validar a implementação através da análise dos resultados e da comparação com soluções analíticas.

Com o propósito de melhor guiar as atividades, são propostos os seguintes objetivos específicos:

1. Identificar e detalhar os aspectos fundamentais do Método de Elementos Discretos, como o modelo matemático considerado e os algoritmos aplicados.
2. Desenvolver uma biblioteca computacional para a simulação da dinâmica de partículas através do DEM. Essa biblioteca deve possuir as seguintes características:
 - a) Suporte para diferentes modelos de colisão, inclusive para modelos estudados e implementados por um usuário posterior;
 - b) Capacidade para um número arbitrário de partículas, as quais podem assumir diferentes formas geométricas e possuir várias propriedades físicas;
 - c) Permitir a inserção de diferentes tipos de interação entre as partículas. Essas interações podem ter naturezas distintas, como a transferência de calor e a troca de cargas elétricas;
 - d) Disponibilidade de utilização de diversos tipos de condição de contorno;
3. Implementar um simulador que, com base na biblioteca desenvolvida, seja capaz de executar simulações que aceitem, no mínimo:
 - a) Partículas esféricas;
 - b) Paredes fixas planas;
 - c) Os principais modelos de força de colisão, conforme explicado na [seção 2.3](#);
 - d) O algoritmo de Gear, empregando a extrapolação de Taylor, de acordo com o [capítulo 2](#);
 - e) Formulações em três dimensões;
4. Validar numericamente a implementação por meio da comparação de resultados de simulações com soluções analíticas para o problema ou através da análise dos resultados.

1.6 Organização do Trabalho

Para melhor cobrir todos os assuntos relativos ao cumprimento dos objetivos, além do capítulo de introdução, este trabalho é dividido nos seguintes capítulos:

Capítulo 2 - Modelo Matemático: Nesse capítulo, são abordados os aspectos teóricos e matemáticos concernentes ao DEM. Nele, são descritas as equações de movimento que devem ser solucionadas, alguns dos principais modelos para as forças de colisão e o algoritmo de Gear;

Capítulo 3 - O Método de Elementos Discretos: Nesse capítulo, é descrito o DEM, demonstrando como os conceitos apresentados no [capítulo 2](#) são aplicados e transformados em algoritmos.

Capítulo 4 - Implementação Computacional e Resultados: Esse capítulo evidencia a implementação da biblioteca e do simulador em linguagem C++. Nesse capítulo, ainda, resultados de simulação são apresentados, confrontados com soluções analíticas e analisados. Com isso, a implementação é validada.

Capítulo 5 - Conclusão: O último capítulo é dedicado a sumarizar o trabalho, reafirmando os principais pontos explorados ao longo do texto e atestando que os objetivos propostos foram alcançados. Também são apresentadas algumas sugestões para trabalhos futuros.

2 Modelo Matemático

Segundo Maliska (2004), a solução numérica de qualquer problema físico requer a sua prévia modelagem matemática. Um modelo matemático é uma representação de um sistema real através de equações. Essas equações são obtidas ao se fazerem hipóteses sobre o comportamento do sistema estudado, e a representatividade do modelo depende das simplificações feitas nesse processo.

Este capítulo é dedicado aos aspectos matemáticos pertinentes ao Método de Elementos Discretos. São abordados os principais conceitos referentes aos sistemas de partículas, e apresentadas as equações de movimento que governam o problema. Os principais modelos de força de colisão são então expostos e, por fim, é apresentado o algoritmo de Gear, responsável por resolver as equações.

2.1 Sistemas de Partículas

As partículas são os elementos fundamentais do Método de Elementos Discretos. De acordo com a definição apresentada por Sampaio (2017), partículas são consideradas corpos rígidos. Por serem indeformáveis, fica implícita a hipótese de que não ocorrem deformações plásticas nesses elementos, e então eles preservam a sua geometria mesmo durante e após colisões.

A princípio, o DEM suporta geometrias arbitrárias para seus elementos. Entretanto, devido a limitações de modelos físicos e de avaliação das colisões, em geral consideram-se formatos básicos como esferas, superelipsoides e poliedros. Geometrias mais complexas podem ser geradas pela associação de geometrias mais simples em partículas compostas (SAMPAIO, 2017; SCHWAGER; PÖSCHEL, 2005).

Um sistema de partículas é, por sua vez, um conjunto de partículas que interagem entre si. Considera-se que o sistema possua um referencial global fixo centrado no ponto O a partir do qual as demais grandezas vetoriais são definidas.

A cada partícula está associada uma função posição que determina, vetorialmente, a posição de seu centro de massa com relação à origem do sistema. A orientação de um elemento é a orientação do seu sistema de eixos principais com relação ao sistema global. Com isso, cada partícula possui seis graus de liberdade: três translações e três rotações (SAMPAIO, 2017). Os elementos discretos ainda contêm diversas *propriedades*, as quais são determinantes para as suas interações.

As translações e as rotações das partículas são determinadas através da solução das equações de movimento, detalhadas na seção 2.2. Essas equações dependem das interações

que ocorrem entre as partículas. Em um sistema qualquer, as interações podem ser de diversas naturezas, tais quais colisões mecânicas, troca de cargas elétricas, transferência de calor, e outras. No entanto o DEM considera, a princípio, apenas as colisões e as forças de campo, como a gravidade.

Embora a quantidade de elementos em um sistema seja arbitrária, ela é sempre finita. Denotando por $N_{\mathcal{P}}$ essa quantidade, é possível enumerarem-se esses elementos e definir-se o conjunto de partículas do sistema como

$$\mathcal{P} = \{\mathcal{P}_i \mid 1 \leq i \leq N_{\mathcal{P}}\},$$

sendo \mathcal{P}_i a i -ésima partícula no sistema.

Cada partícula ainda possui uma *vizinhança*, que é o conjunto de elementos que podem interagir com ela. Denota-se por \mathcal{N}_i ou $\mathcal{N}(\mathcal{P}_i)$ a vizinhança da partícula i . A determinação da vizinhança de uma partícula é essencial para o cálculo das forças e dos torques que atuam sobre ela, e alguns dos métodos para se construir esse conjunto são apresentados na [seção 3.3](#).

Escrevendo como \mathbf{F}_{ji} a força e \mathbf{M}_{ji} o torque aplicados pela partícula \mathcal{P}_j sobre a partícula \mathcal{P}_i , tem-se que a força resultante \mathbf{F}_i e o torque resultante \mathbf{M}_i sobre \mathcal{P}_i são dados pelo princípio da superposição como

$$\mathbf{F}_i = \sum_{\mathcal{P}_j \in \mathcal{N}_i} \mathbf{F}_{ji} + \mathbf{F}_i^{\text{ext}}, \quad (2.1)$$

$$\mathbf{M}_i = \sum_{\mathcal{P}_j \in \mathcal{N}_i} \mathbf{M}_{ji} + \mathbf{M}_i^{\text{ext}}, \quad (2.2)$$

em que $\mathbf{F}_i^{\text{ext}}$ e $\mathbf{M}_i^{\text{ext}}$ são a força e o torque *externos* atuantes sobre \mathcal{P}_i , isto é, resultantes da interação entre a i -ésima partícula com elementos exteriores ao sistema.

Uma importante hipótese que se faz no Método de Elementos Discretos é que a interação entre duas partículas \mathcal{P}_i e \mathcal{P}_j não interfere na interação entre outras partículas ou entre \mathcal{P}_i e uma terceira partícula \mathcal{P}_ℓ . Com isso, cada par de partículas pode ser estudado isoladamente, e o estado final do sistema, após as interações, é uma simples combinação desses resultados.

2.2 Equações de Movimento

Na dinâmica de partículas, os elementos estudados são considerados corpos rígidos, aos quais se aplicam as leis de Euler para o movimento ([SAMPAIO, 2017; WITTEBURG, 2008](#)).

Em um sistema inercial fixo, considera-se uma partícula de massa m com um centro de massa C cuja posição é descrita pela função vetorial \mathbf{r} . Além disso, considera-se que a

partícula possua uma velocidade angular $\boldsymbol{\omega}$ e um tensor momento de inércia $\hat{\mathbf{J}}$ definido com relação ao seu centro de massa.

Conforme demonstrado por Wittenburg (2008), a *quantidade de movimento linear* e a *quantidade de movimento angular* da partícula, denotadas respectivamente por \mathbf{p} e \mathbf{L} , são então dadas por

$$\mathbf{p} = m \cdot \dot{\mathbf{r}},$$

$$\mathbf{L} = \mathbf{r} \times \mathbf{p} + \hat{\mathbf{J}} \cdot \boldsymbol{\omega}.$$

2.2.1 Primeira Lei de Euler

A primeira lei de Euler trata do movimento de translação. Ela se expressa através da equação

$$\mathbf{F}_R = \dot{\mathbf{p}}, \quad (2.3)$$

sendo \mathbf{F}_R o vetor força resultante sobre a partícula. Para o caso em que a massa do corpo é constante, a primeira lei de Euler torna-se equivalente à segunda lei de Newton:

$$\mathbf{F}_R = m \cdot \ddot{\mathbf{r}}. \quad (2.4)$$

A equação (2.3) é aplicada nos fenômenos de fragmentação e aglutinação, casos em que partículas são divididas ou unidas, sendo que a massa de cada partícula isoladamente não é constante. Schwager e Pöschel (2005) apresentam métodos que descrevem o fenômeno de fragmentação. Tais situações, porém, são tratadas separadamente, de modo que a equação (2.4) é a que se considera para o estudo da evolução do sistema.

De acordo com o princípio da superposição, a força resultante sobre a partícula é igual à soma vetorial todas as forças aplicadas sobre ela, conforme a equação (2.1). Essas forças são comumente divididas em *forças internas* e *forças externas* ao sistema. As forças internas são aquelas oriundas das interações entre os elementos do sistema, enquanto as forças externas são atribuídas a elementos exteriores.

No sistema, as forças internas são sempre sujeitas ao princípio de ação e reação. Se a partícula \mathcal{P}_i aplica uma força \mathbf{F}_{ij} sobre a partícula \mathcal{P}_j , então, pela terceira lei de Newton, a força \mathbf{F}_{ji} que \mathcal{P}_j aplica sobre \mathcal{P}_i é dada por

$$\mathbf{F}_{ji} = -\mathbf{F}_{ij}.$$

Para a translação, os movimentos ocorrem separadamente em cada eixo coordenado de modo que os algoritmos de solução da equação (2.4) são facilmente generalizados de duas para três dimensões. Ainda, cabe observar que a primeira lei de Euler independe da posição do eixo de referência. Isso nem sempre ocorre no movimento de rotação.

2.2.2 Segunda Lei de Euler

A segunda lei de Euler, por sua vez, é o análogo rotacional da primeira:

$$\mathbf{M}_R = \dot{\mathbf{L}},$$

em que \mathbf{M}_R é o torque resultante sobre a partícula calculado com relação ao seu centro de massa. Para partículas rígidas com massa constante, é possível obter

$$\mathbf{M}_R = \dot{\mathbf{L}} = \hat{\mathbf{J}} \cdot \dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (\hat{\mathbf{J}} \cdot \boldsymbol{\omega}), \quad (2.5)$$

sendo $\dot{\boldsymbol{\omega}}$ a *aceleração angular* do elemento.

Novamente, a hipótese de que a massa da partícula é constante é geralmente considerada.

Entretanto, mesmo diante de situações em que m é constante, a resolução da equação (2.5) possui algumas complicações em simulações tridimensionais. Isso se deve ao fato de que são necessárias parametrizações para a orientação da partícula, e não ocorre que a velocidade angular seja a derivada desses parâmetros. Só é possível escrever $\boldsymbol{\omega} = \dot{\boldsymbol{\varphi}}$, isto é, a velocidade angular como taxa de variação de uma função de orientação, quando a direção da rotação é constante (WITTENBURG, 2008, p. 32).

O tensor $\hat{\mathbf{J}}$ identifica-se com a matriz de inércia $\underline{\mathbf{J}}$ dada por

$$\underline{\mathbf{J}} = \begin{pmatrix} J_{xx} & J_{xy} & J_{xz} \\ J_{yx} & J_{yy} & J_{yz} \\ J_{zx} & J_{zy} & J_{zz} \end{pmatrix}.$$

Para um ponto P qualquer, define-se o vetor radial $\underline{\boldsymbol{\rho}} = \overrightarrow{CP}$. Denotando por $\underline{\boldsymbol{\rho}}$ a matriz coluna correspondente a $\underline{\boldsymbol{\rho}}$, a matriz de inércia do corpo \mathcal{P}_i é dada por

$$\underline{\mathbf{J}} = \int_{\mathcal{P}_i} (\underline{\boldsymbol{\rho}}^T \underline{\boldsymbol{\rho}} \cdot \underline{\mathbf{I}} - \underline{\boldsymbol{\rho}} \underline{\boldsymbol{\rho}}^T) dm,$$

em que $\underline{\mathbf{I}}$ é a matriz identidade. Alternativamente, escrevendo-se $\underline{\boldsymbol{\rho}} = (\rho_x, \rho_y, \rho_z)$,

$$\underline{\mathbf{J}} = \begin{pmatrix} \int_{\mathcal{P}_i} (\rho_y^2 + \rho_z^2) dm & - \int_{\mathcal{P}_i} \rho_x \rho_y dm & - \int_{\mathcal{P}_i} \rho_x \rho_z dm \\ \int_{\mathcal{P}_i} (\rho_x^2 + \rho_z^2) dm & \int_{\mathcal{P}_i} (\rho_x^2 + \rho_y^2) dm & - \int_{\mathcal{P}_i} \rho_y \rho_z dm \\ \text{Simétrica} & \int_{\mathcal{P}_i} (\rho_x^2 + \rho_y^2) dm & \end{pmatrix}.$$

A cada corpo estão associados *eixos principais*, formados pelos autovetores da matriz de momento de inércia. Ao se utilizarem como sistema de referência os eixos principais, a matriz de inércia assume forma diagonal:

$$\underline{\mathbf{J}}^* = \begin{pmatrix} J_1 & 0 & 0 \\ 0 & J_2 & 0 \\ 0 & 0 & J_3 \end{pmatrix},$$

e reescrevendo, nessas coordenadas, o torque resultante como $\mathbf{M}_R^* = (M_1, M_2, M_3)$, a velocidade angular como $\boldsymbol{\omega}^* = (\omega_1, \omega_2, \omega_3)$ e a aceleração angular como $\dot{\boldsymbol{\omega}}^* = (\dot{\omega}_1, \dot{\omega}_2, \dot{\omega}_3)$, a equação (2.5) se transforma no sistema de equações

$$\begin{cases} J_1\dot{\omega}_1 - (J_2 - J_3)\omega_2\omega_3 = M_1 \\ J_2\dot{\omega}_2 - (J_3 - J_1)\omega_3\omega_1 = M_2 \\ J_3\dot{\omega}_3 - (J_1 - J_2)\omega_1\omega_2 = M_3 \end{cases}. \quad (2.6)$$

Esse sistema pode ser resolvido de forma análoga à primeira lei de Euler, conforme discutido na [subseção 2.2.3](#). Os valores de J_1 , J_2 e J_3 são dependentes da geometria da partícula e da distribuição de massa no seu interior. Em geral, essas características já são conhecidas no início das simulações.

Entretanto o sistema (2.6) exige que se conheçam os eixos principais da partícula, isto é, sua orientação. Esses eixos, porém, movem-se e rotacionam junto com o corpo, e então são variantes no tempo. Para se resolver o problema no caso geral, são propostas parametrizações, tais quais os ângulos de Euler, os ângulos de Bryan, e os parâmetros de Euler-Rodrigues. Detalhes sobre essas parametrizações podem ser encontrados em [Wittenburg \(2008, pp. 9–23\)](#) e em [Schwager e Pöschel \(2005, pp. 68–75\)](#).

Duas simplificações que podem se aplicar à equação (2.5) são: que a simulação ocorre em apenas duas dimensões; ou que as partículas são esféricas. No primeiro caso, a direção de rotação é constante. Essa situação é tratada na [subseção 2.2.2.1](#). Já no segundo caso, como explicado na [subseção 2.2.2.2](#), qualquer sistema de eixos centrado no centro de massa da partícula é um sistema de eixos principais, e então a orientação das partículas não é mais necessária para a determinação de (2.6). Por fim, na [subseção 2.2.2.3](#), apresenta-se o caso de rotação geral em três dimensões.

2.2.2.1 Simulação Bidimensional

Em se supondo que a simulação ocorre em apenas duas dimensões, por convenção no plano XY , a velocidade angular das partículas possui direção constante, a do eixo Z . Denotando por $\underline{e} = (\mathbf{i}, \mathbf{j}, \mathbf{k})$ a base canônica do espaço \mathbb{R}^3 , isso implica que existe uma função real ω tal que

$$\boldsymbol{\omega} = \omega \cdot \mathbf{k}.$$

Já que, neste caso, a velocidade angular possui direção constante, é possível encontrar uma função φ para a orientação da partícula, de forma que

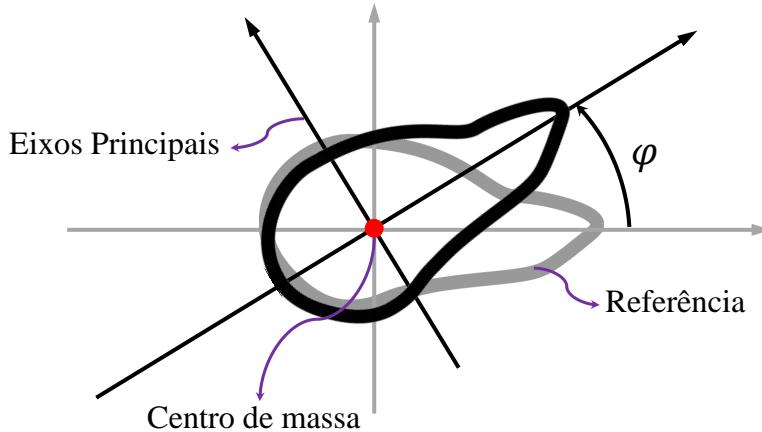
$$\omega = \dot{\varphi},$$

$$\boldsymbol{\omega} = \dot{\varphi} \cdot \mathbf{k},$$

$$\dot{\boldsymbol{\omega}} = \ddot{\varphi} \cdot \mathbf{k}.$$

Geometricamente, φ é o ângulo entre os eixos principais da partícula e sua orientação de referência, como na figura 2.1.

Figura 2.1 – Ângulo de rotação de uma partícula com relação à posição de referência em simulações bidimensionais.



Fonte: do Autor.

Além disso, da hipótese de que as rotações ocorrem apenas em torno do eixo Z segue que o torque resultante possui apenas componentes nessa direção, isto é,

$$\mathbf{M}_R = M \cdot \mathbf{k},$$

com $M \in \mathbb{R}$.

Com isso, a equação (2.5) fica simplesmente

$$M = J_{zz} \cdot \ddot{\varphi},$$

em que J_{zz} é o momento de inércia escalar da partícula com relação ao eixo Z . Essa última equação se torna, então, a equação de movimento rotacional para partículas em simulações em duas dimensões.

Uma situação um pouco mais geral é que a direção da velocidade angular seja apenas constante, não necessariamente a do eixo Z , mas sim a de um eixo qualquer Z' . Nesse caso, existe uma nova base $\underline{e}' = (\mathbf{i}', \mathbf{j}', \mathbf{k}')$ contendo os versores dos eixos $X'Y'Z'$ que satisfaz

$$\boldsymbol{\omega} = \omega \cdot \mathbf{k}',$$

e, analogamente ao caso anterior, uma função de orientação φ tal que

$$\begin{aligned} \boldsymbol{\omega} &= \dot{\varphi} \cdot \mathbf{k}', \\ \dot{\boldsymbol{\omega}} &= \ddot{\varphi} \cdot \mathbf{k}'. \end{aligned}$$

Novamente, a hipótese de direção de rotação constante implica que é possível escrever

$$\mathbf{M}_R = M \cdot \mathbf{k}',$$

e assim se obtém a equação

$$M = J'_{zz} \cdot \ddot{\varphi}.$$

Para a solução dessa equação, porém, ainda é necessária a determinação do valor de J'_{zz} . Isso pode ser feito como sugere Wittenburg (2008), calculando-se a matriz de rotação $\underline{\mathbf{A}}$ que transforma as coordenadas de vetores definidos na base $\underline{\mathbf{e}}$ em coordenadas definidas no sistema $X'Y'Z'$, na base $\underline{\mathbf{e}'}$, e obtendo

$$\underline{\mathbf{J}'} = \begin{pmatrix} J'_{xx} & J'_{xy} & J'_{xz} \\ J'_{yx} & J'_{yy} & J'_{yz} \\ J'_{zx} & J'_{zy} & J'_{zz} \end{pmatrix} = \underline{\mathbf{A}} \cdot \underline{\mathbf{J}} \cdot \underline{\mathbf{A}}^T.$$

Embora tenda-se a evitar multiplicações matriciais, esse cálculo pode ser feito apenas uma vez, no início da simulação, desde que se saiba a priori a direção da rotação.

2.2.2.2 Rotação de Partículas Esféricas

Segundo Wittenburg (2008), para partículas esféricas, *todas* as direções são direções principais. Sendo assim, o sistema de equações (2.6) é válido em qualquer sistema de eixos. Mais ainda, em função da simetria da partícula, ocorre que $J_1 = J_2 = J_3$. Ao se definir $J := J_1$, obtém-se

$$\begin{cases} J\dot{\omega}_1 = M_1 \\ J\dot{\omega}_2 = M_2 \\ J\dot{\omega}_3 = M_3 \end{cases},$$

ou ainda

$$\mathbf{M}_R = J \cdot \dot{\boldsymbol{\omega}}, \quad (2.7)$$

sendo que o momento de inércia J é facilmente calculado pela expressão

$$J = \frac{2}{5} \cdot mR^2, \quad (2.8)$$

em que R é o raio da esfera.

Com isso, a equação diferencial para o movimento de rotação está pronta para simulações tridimensionais de partículas esféricas.

2.2.2.3 Rotação Geral em Três Dimensões

Para simulações em três dimensões com partículas não esféricas, utiliza-se o sistema de equações (2.6). A solução do sistema, porém, não é trivial já que as equações são dadas em termos das direções principais da partícula.

As direções principais de uma partícula são dadas pela base ortonormal $\underline{e}^* = (\mathbf{i}^*, \mathbf{j}^*, \mathbf{k}^*)$ cujos versores são autovetores da matriz de momento de inércia. A importância dessa base se dá pelo fato de que a matriz de momento de inércia se torna diagonal quando escrita em termos de seus vetores.

Entretanto, em virtude do caráter dinâmico dos sistemas de partículas, as direções principais dos elementos são funções do tempo: $\underline{e}^* \equiv \underline{e}^*(t)$. Uma *parametrização* para a orientação é qualquer conjunto de coordenadas φ que defina, equivalentemente, as direções principais. Diversas parametrizações já foram propostas, como citado anteriormente, variando entre si em termos de estabilidade, eficiência, número de variáveis consideradas e de equações de restrição.

Com a parametrização φ , sempre é possível obter os eixos principais através de uma transformação inversível:

$$\underline{e}^*(t) \leftrightarrow \varphi(t),$$

e, com isso, determina-se uma matriz de rotação φ que transforma vetores da base de eixos locais na base das direções principais da partícula.

Ainda, com os eixos principais dados em função de φ , a velocidade angular e suas derivadas podem ser escritas a partir da parametrização e suas derivadas:

$$\begin{aligned}\omega &\equiv \omega(\varphi, \dot{\varphi}), \\ \dot{\omega} &\equiv \dot{\omega}(\varphi, \dot{\varphi}, \ddot{\varphi}), \\ &\vdots \\ \omega^{(k)} &\equiv \omega^{(k)}(\varphi, \dot{\varphi}, \dots, \varphi^{(k+1)}).\end{aligned}$$

Essas relações são inversíveis, e então é possível obter

$$\begin{aligned}\dot{\varphi} &\equiv \dot{\varphi}(\varphi, \omega) \\ \ddot{\varphi} &\equiv \ddot{\varphi}(\varphi, \omega, \dot{\omega}) \\ &\vdots \\ \varphi^{(k+1)} &\equiv \varphi^{(k+1)}(\varphi, \omega, \dots, \omega^{(k)}).\end{aligned}$$

Assim, com a parametrização φ , define-se toda a cinemática da rotação da partícula. Mais ainda, devido à equivalência entre o sistema de eixos principais e a parametrização, vetores definidos com relação ao sistema de eixos locais podem ser escritos em termos das direções principais. Isso significa que

$$\mathbf{M}_R^* \equiv \mathbf{M}_R^*(\mathbf{M}_R, \varphi).$$

Como resultado dessas considerações, o sistema de equações (2.6) transforma-se em uma equação diferencial da forma

$$\ddot{\varphi} = f^\varphi(\mathbf{M}_R, \varphi, \dot{\varphi}). \quad (2.9)$$

A função f^φ representa a dependência de $\dot{\varphi}$ com relação a M_R , φ e $\dot{\varphi}$, e seu formato depende da definição da parametrização e da expressão para o torque resultante em função das variáveis do problema.

2.2.3 Solução das Equações de Movimento

Para a determinação do comportamento do sistema, é necessária a obtenção das soluções para as equações (2.4) e (2.5). Isso exige o cálculo da força e do torque resultantes sobre cada partícula e a consequente resolução da equação.

O cálculo das forças e dos torques geralmente leva em conta *modelos de interação*. Cada modelo determina uma parcela do vetor força ou torque resultante a partir de parâmetros de material, da posição, da velocidade, da orientação e da velocidade angular das partículas. Na [seção 2.3](#), são apresentados alguns desses modelos. Os vetores resultantes são então a soma de todas as parcelas.

Com os vetores força e torque resultantes, as equações diferenciais do movimento ficam bem determinadas. Resta a escolha de procedimentos para a resolução dessas equações, algoritmos chamados de *métodos de integração* que, dadas duas partículas no instante t e as equações de movimento, sejam capazes de calcular as coordenadas e as propriedades dessas partículas em um instante futuro $t + \Delta t$.

As principais características consideradas para a escolha de um método de solução das equações diferenciais são a exatidão, a estabilidade e o custo computacional. Neste trabalho, é utilizado o algoritmo de Gear, descrito na [seção 2.4](#). A motivação para seu uso é que, como indicado na [seção 3.3](#), o principal custo computacional dos métodos de elementos discretos advém do cômputo das forças em cada instante de tempo. O método de Gear necessita de apenas uma avaliação das interações por passo de tempo, o que representa uma eficiência máxima nesse sentido. A estabilidade do algoritmo ainda dispensa iterações dentro de cada passo de tempo, reduzindo, mais uma vez, o custo computacional. Outros algoritmos de integração são o método baseado em pontos centrais *leapfrog*, os métodos de Verlet e os de Runge-Kutta ([SAMPAIO, 2017](#)).

De uma simulação, é possível extrair diversas informações. Alguns dos principais resultados que se buscam obter são a trajetória das partículas, o carregamento ou a pressão sobre determinada superfície, o coeficiente de restituição resultante das colisões, tempo para esvaziamento de um reservatório, entre outros.

O procedimento de se calcularem as forças e os torques e, posteriormente, resolverem-se as equações diferenciais é conhecido como *simulação baseada em forças*. Alternativamente a esse método, é possível executarem-se *simulações orientadas a eventos*. Nessas simulações, apenas o coeficiente de restituição é considerado na colisão entre as partículas. Essa técnica está fora do escopo desse trabalho, mas pode ser encontrada com mais detalhes em [Schwager](#)

e Pöschel (2005).

2.3 Modelos de Força de Colisão

Dentre as interações entre partículas do sistema, as mais comumente encontradas são as forças de colisão que surgem durante o seu contato. Os modelos de força têm o papel de atribuir, para cada par de partículas, uma componente de força \mathbf{F} e uma componente de torque \mathbf{M} .

Seguindo a notação apresentada na [seção 2.1](#), denota-se por \mathbf{F}_{ji} a componente de força aplicada pela partícula \mathcal{P}_j sobre a partícula \mathcal{P}_i , e por \mathbf{M}_{ji} a componente de torque correspondente definida no sistema de eixos centrado na partícula \mathcal{P}_i .

Os modelos de colisão mais simples são aqueles que ocorrem entre partículas esféricas ou entre uma partícula esférica e uma parede plana. Nesta seção, são apresentados alguns dos parâmetros geralmente empregados na determinação de forças e torques e alguns dos principais modelos considerados, segundo [Schwager e Pöschel \(2005\)](#).

2.3.1 Parâmetros de Cálculo

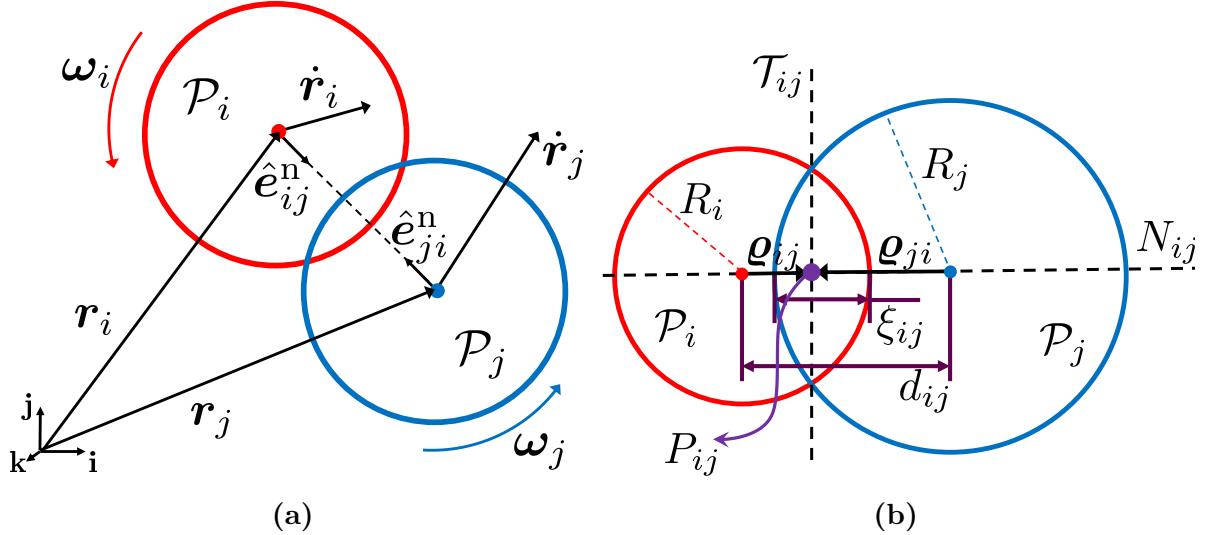
Os principais parâmetros considerados para o cálculo das forças de colisão são a geometria das partículas, as suas propriedades físicas, a velocidade relativa no ponto de contato, a *superposição* e os versores normal e tangencial. Também, o ponto de contato e o plano de contato são necessários na determinação da geometria do choque. Esses parâmetros descrevem as condições em que a colisão ocorre, e assim determinam as componentes de força. Além disso, o coeficiente de restituição e as energias cinética, potencial e mecânica são de grande importância para a análise dos resultados. Na [figura 2.2a](#) estão representadas graficamente as posições e as velocidades das partículas colidentes, assim como os versores normais. Na [figura 2.2b](#), estão a reta que liga os centros das partículas, o plano de contato, os vetores radiais de colisão, a distância entre centros e a superposição.

Superposição

No Método de Elementos Discretos, os elementos são considerados corpos rígidos. Sendo assim, quando ocorre contato entre dois elementos, não ocorre deformação, mas, sim, uma superposição ou penetração.

A medida da penetração pode variar dependendo dos tipos de elementos considerados. Para corpos com geometrias arbitrárias, pode ser necessária a medição do volume sobreposto. Entretanto, para um par de partículas esféricas ou um par formado por uma esfera e uma parede plana, é suficiente definir-se a superposição linearmente, como na [figura 2.2](#).

Figura 2.2 – Representação dos raios, das posições, e das velocidades das partículas, da superposição, dos versores normais, dos vetores radiais, do plano de contato e da reta normal.



Fonte: do Autor.

Sendo \mathcal{P}_i e \mathcal{P}_j duas partículas esféricas distintas, pode-se definir a distância entre seus centros como

$$d_{ij} = \|\mathbf{r}_j - \mathbf{r}_i\|.$$

Denotando-se por R_i e R_j os raios das partículas \mathcal{P}_i e \mathcal{P}_j , ocorre que há superposição se, e somente se, $R_i + R_j > d_{ij}$. Nesse caso, a superposição ξ_{ij} entre os elementos é dada por $\xi_{ij} = R_i + R_j - d_{ij}$.

Caso a distância entre os centros das partículas seja maior que a soma de seus raios, a superposição é nula. Com isso, pode-se definir a superposição entre as partículas \mathcal{P}_i e \mathcal{P}_j como

$$\xi_{ij} = \max \{R_i + R_j - d_{ij}, 0\}. \quad (2.10)$$

A simples diferenciação da equação (2.10) com relação ao tempo resulta na derivada da superposição:

$$\dot{\xi}_{ij} = \begin{cases} -\frac{\langle \mathbf{r}_j - \mathbf{r}_i, \dot{\mathbf{r}}_j - \dot{\mathbf{r}}_i \rangle}{\|\mathbf{r}_j - \mathbf{r}_i\|}, & \text{se } \xi_{ij} > 0 \\ 0, & \text{se } \xi_{ij} = 0 \end{cases}, \quad (2.11)$$

em que $\langle \cdot, \cdot \rangle$ representa o produto interno entre vetores.

Para o caso do contato entre esfera e parede plana, consideram-se \mathcal{P}_i uma partícula esférica e \mathcal{E}_j um elemento plano fixo no espaço. Sejam \mathbf{r}_j um ponto nesse plano e $\hat{\mathbf{n}}_j$ um versor normal a ele. A distância do centro de \mathcal{P}_i ao plano é dada por

$$d_{ij} = |\langle \mathbf{r}_i - \mathbf{r}_j, \hat{\mathbf{n}}_j \rangle|.$$

Assim, a sobreposição entre \mathcal{P}_i e \mathcal{E}_j fica

$$\xi_{ij} = \max \{R_i - d_{ij}, 0\},$$

e sua derivada,

$$\dot{\xi}_{ij} = \begin{cases} -\langle \dot{\mathbf{r}}_i, \hat{\mathbf{n}}_j \rangle, & \text{se } \xi_{ij} > 0 \text{ e } \langle \mathbf{r}_i - \mathbf{r}_j, \hat{\mathbf{n}}_j \rangle > 0 \\ \langle \dot{\mathbf{r}}_i, \hat{\mathbf{n}}_j \rangle, & \text{se } \xi_{ij} > 0 \text{ e } \langle \mathbf{r}_i - \mathbf{r}_j, \hat{\mathbf{n}}_j \rangle < 0 \\ 0, & \text{se } \xi_{ij} = 0 \end{cases}.$$

Vessor Normal

Sendo \mathcal{P}_i e \mathcal{P}_j duas partículas esféricas, o vessor normal que aponta de \mathcal{P}_i para \mathcal{P}_j é dado por

$$\hat{\mathbf{e}}_{ij}^n = \frac{\mathbf{r}_j - \mathbf{r}_i}{\|\mathbf{r}_j - \mathbf{r}_i\|},$$

e é fácil ver que $\hat{\mathbf{e}}_{ij}^n = -\hat{\mathbf{e}}_{ji}^n$.

Com isso, a derivada da superposição entre partículas esféricas pode ser calculada, equivalentemente à equação (2.11), como

$$\dot{\xi}_{ij} = \begin{cases} -\langle \dot{\mathbf{r}}_j - \dot{\mathbf{r}}_i, \hat{\mathbf{e}}_{ij}^n \rangle, & \text{se } \xi_{ij} > 0 \\ 0, & \text{se } \xi_{ij} = 0 \end{cases}. \quad (2.12)$$

Já para o caso de um elemento plano fixo \mathcal{E}_j , o vessor normal $\hat{\mathbf{e}}_{ji}^n$ é paralelo ao vessor normal ao plano, isto é, $\hat{\mathbf{e}}_{ji}^n = \pm \hat{\mathbf{n}}_j$. O sinal adotado é positivo se a partícula se encontrar na direção do vessor normal, e negativo no caso contrário. Isso pode ser escrito como

$$\hat{\mathbf{e}}_{ji}^n = \operatorname{sgn}(\langle \mathbf{r}_i - \mathbf{r}_j, \hat{\mathbf{n}}_j \rangle) \cdot \hat{\mathbf{n}}_j,$$

sendo $\operatorname{sgn}: \mathbb{R} \rightarrow \{-1, 0, 1\}$ a função sinal definida por

$$\operatorname{sgn} x = \begin{cases} 1, & \text{se } x > 0 \\ 0, & \text{se } x = 0 \\ -1, & \text{se } x < 0 \end{cases}.$$

Novamente, $\hat{\mathbf{e}}_{ij}^n = -\hat{\mathbf{e}}_{ji}^n$.

Em ambos os casos, é possível construir-se, a partir do centro da partícula i , a reta normal N_{ij} da colisão entre i e j como o espaço gerado por $\hat{\mathbf{e}}_{ij}^n$:

$$N_{ij} = \left\{ \mathbf{r}_i + \varrho \cdot \hat{\mathbf{e}}_{ij}^n \mid \varrho \in \mathbb{R} \right\}.$$

Ponto de Contato

Tanto na colisão entre partículas esféricas quanto na colisão entre partícula esférica e parede plana, existe um único plano, denominado de *plano de contato* e denotado por

\mathcal{T}_{ij} , que contém todos os pontos de interseção entre as superfícies dos colidentes, como mostra a figura 2.2.

O ponto de contato efetivo P_{ij} entre os elementos i e j é dado pela interseção entre o plano de contato e a reta normal N_{ij} . Vetorialmente, o contato é representado pela posição $\overrightarrow{OP_{ij}}$.

Esse ponto também pode ser representado pelo vetor radial $\boldsymbol{\varrho}_{ij} = \varrho_{ij} \cdot \hat{\mathbf{e}}_{ij}^n$ como

$$\overrightarrow{OP_{ij}} = \mathbf{r}_i + \boldsymbol{\varrho}_{ij}. \quad (2.13)$$

Para duas partículas esféricas, a geometria do contato é tal como mostrado na figura 2.2. Segue então do teorema de Pitágoras que

$$R_i^2 - \varrho_{ij}^2 = R_j^2 - \varrho_{ji}^2.$$

Além disso,

$$\varrho_{ij} + \varrho_{ji} = d_{ij},$$

e então, ao se combinarem essas duas equações, obtém-se que

$$\varrho_{ij} = \frac{R_i^2 - R_j^2 + d_{ij}^2}{2 \cdot d_{ij}}. \quad (2.14)$$

Com as equações (2.13) e (2.14), o ponto de contato fica bem determinado.

A situação se simplifica para a colisão entre uma partícula esférica e um elemento plano, já que ocorre que

$$\varrho_{ij} = d_{ij}.$$

Em ambos os casos, com a definição do vetor radial $\boldsymbol{\varrho}_{ij}$, a componente de força \mathbf{F}_{ji} está associada a uma componente de torque \mathbf{M}_{ji} dada por

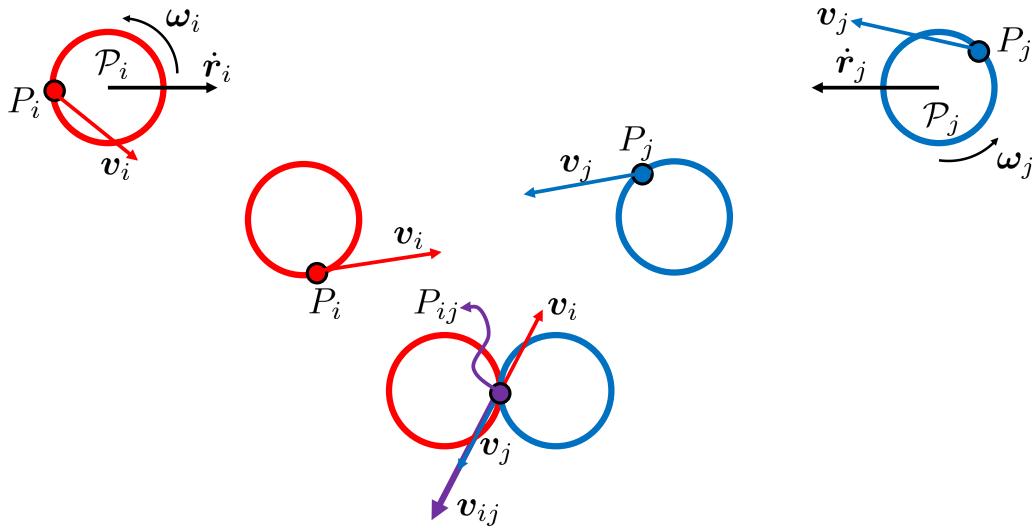
$$\mathbf{M}_{ji} = \boldsymbol{\varrho}_{ij} \times \mathbf{F}_{ji}$$

Velocidade Relativa no Ponto de Contato

A velocidade relativa no ponto de contato é o principal parâmetro cinemático na determinação da força tangencial que ocorre entre dois elementos. A velocidade do elemento j relativa ao elemento i no ponto de contato é denotada por \mathbf{v}_{ij} .

O cálculo dessa velocidade leva em consideração o fato de que, no ponto de contato P_{ij} , estão superpostos dois pontos: um ponto P_i pertencente à partícula i e um ponto P_j que pertence ao elemento j . Cada um desses pontos move-se juntamente com o elemento correspondente e, por ocasião do contato, os dois acabam por se encontrar. Isso é representado na figura 2.3.

Figura 2.3 – Velocidade relativa no ponto de contato.



Fonte: do Autor.

Como esses pontos movem-se no espaço, os vetores $\overrightarrow{OP_i}$ e $\overrightarrow{OP_j}$ não são constantes, mas, sim, funções do tempo com velocidades \mathbf{v}_i e \mathbf{v}_j . A velocidade \mathbf{v}_{ij} é então definida como a velocidade relativa entre esses dois pontos:

$$\mathbf{v}_{ij} = \frac{d}{dt}(\overrightarrow{OP_j} - \overrightarrow{OP_i}) = \mathbf{v}_j - \mathbf{v}_i.$$

De acordo com Wittenburg (2008, p. 25), e usando o fato de que os pontos P_i e P_j são fixos nos elementos a que pertencem (ou seja, os pontos têm velocidade relativa nula com relação aos seus respectivos corpos), é possível escrever

$$\begin{aligned}\mathbf{v}_i &= \dot{\mathbf{r}}_i + \boldsymbol{\omega}_i \times \boldsymbol{\varrho}_{ij}, \\ \mathbf{v}_j &= \dot{\mathbf{r}}_j + \boldsymbol{\omega}_j \times \boldsymbol{\varrho}_{ji}.\end{aligned}$$

Com isso, obtém-se

$$\mathbf{v}_{ij} = (\dot{\mathbf{r}}_j - \dot{\mathbf{r}}_i) + (\boldsymbol{\omega}_j \times \boldsymbol{\varrho}_{ji} - \boldsymbol{\omega}_i \times \boldsymbol{\varrho}_{ij}),$$

e nota-se dessa equação que

$$\mathbf{v}_{ji} = -\mathbf{v}_{ij}. \quad (2.15)$$

A velocidade normal de j relativa ao elemento i é a projeção de \mathbf{v}_{ij} na direção de $\hat{\mathbf{e}}_{ij}^n$:

$$\begin{aligned}\mathbf{v}_{ij}^n &= \langle \mathbf{v}_{ij}, \hat{\mathbf{e}}_{ij}^n \rangle \cdot \hat{\mathbf{e}}_{ij}^n \\ &= v_{ij}^n \cdot \hat{\mathbf{e}}_{ij}^n.\end{aligned} \quad (2.16)$$

Percebe-se, em consequência das equações (2.11), (2.12) e (2.16), que, tanto para a colisão entre partículas esféricas quanto para a colisão entre uma esfera e uma parede

plana, ocorre que

$$\dot{\xi}_{ij} = -v_{ij}^n. \quad (2.17)$$

Por sua vez, velocidade tangencial de j relativa ao elemento i é simplesmente a correspondente velocidade relativa sem a componente na direção normal:

$$\mathbf{v}_{ij}^t = \mathbf{v}_{ij} - \mathbf{v}_{ij}^n.$$

As simplificações que se fazem no caso em que o elemento j é uma parede plana são que

$$\dot{\mathbf{r}}_j = \mathbf{0},$$

$$\boldsymbol{\omega}_j = \mathbf{0},$$

do que segue que

$$\mathbf{v}_{ij} = -\dot{\mathbf{r}}_i - \boldsymbol{\omega}_i \times \boldsymbol{\varrho}_{ij}. \quad (2.18)$$

Versor Tangencial

Com a velocidade tangencial determinada, é possível definir-se o versor tangencial $\hat{\mathbf{e}}_{ij}^t$ como

$$\hat{\mathbf{e}}_{ij}^t = \frac{\mathbf{v}_{ij}^t}{\|\mathbf{v}_{ij}^t\|},$$

e então escrever

$$\mathbf{v}_{ij}^t = v_{ij}^t \cdot \hat{\mathbf{e}}_{ij}^t.$$

Para o caso em que $\mathbf{v}_{ij}^t = \mathbf{0}$, não é necessário calcular-se o versor tangencial pois, nessa situação, as forças tangenciais são sempre nulas, de acordo com os modelos de interação.

Coeficiente de Restituição

A determinação do coeficiente de restituição tem a vantagem de poder simplificar e reduzir o tempo de simulação em simulações orientadas a eventos, mas também pode ser usada para a validação de métodos numéricos baseados no cálculo das forças, que são o foco deste trabalho.

Para cada par de elementos, o coeficiente de restituição relaciona o estado posterior ao choque ao estado anterior. Escrevendo a velocidade do elemento j relativa ao i imediatamente antes da colisão como \mathbf{v}_{ij} , e a velocidade correspondente no final da colisão como $(\mathbf{v}_{ij})'$, define-se o coeficiente de restituição na direção normal ε_{ij}^n através da equação

$$(v_{ij}^n)' = -\varepsilon_{ij}^n \cdot v_{ij}^n.$$

De acordo com [Schwager e Pöschel \(2005\)](#), a seguinte desigualdade é sempre satisfeita:

$$0 \leq \varepsilon_{ij}^n \leq 1.$$

Formas analíticas para o coeficiente de restituição podem ser obtidas através da integração das equações diferenciais, dependendo dos modelos de interação considerados. Em função da complexidade do modelo, pode ser muito difícil, ou mesmo impossível, de se obter uma forma fechada para esse coeficiente. Outras maneiras de se determinar seu valor são o uso de simulações numéricas e a experimentação, em que se busca correlacioná-lo com as demais grandezas do problema.

De forma geral, o valor desse coeficiente depende de parâmetros de material e da velocidade relativa entre as partículas. Segundo [Schwager e Pöschel \(2005\)](#), coeficientes de restituição constantes (independentes da velocidade relativa) estão em desacordo com resultados experimentais e com o importante modelo de partículas viscoelásticas. Entretanto coeficientes constantes formam a base para diversas teorias e podem ser bastante úteis para a validação de métodos numéricos.

É possível ainda definir-se um coeficiente de restituição tangencial ε_{ij}^t conforme [Schwager e Pöschel \(2005\)](#), mas esse coeficiente não é usado neste trabalho.

Feitas essas considerações, os principais parâmetros para o cálculo das forças e torques de colisão estão bem definidos. Nas seções [2.3.2](#) e [2.3.3](#) são apresentados alguns dos modelos de força mais adotados e que se utilizam desses parâmetros.

Energia

A energia de uma partícula é uma quantidade escalar de extrema importância na análise de resultados de simulações. De acordo com [Wittenburg \(2008\)](#), a energia translacional E_T de uma partícula é dada por

$$E_T = \frac{1}{2} \cdot m \|\dot{\mathbf{r}}\|^2,$$

enquanto a energia rotacional E_R associada é

$$E_R = \frac{1}{2} \cdot \boldsymbol{\omega} \cdot \hat{\mathbf{J}} \cdot \boldsymbol{\omega}.$$

Com isso, a energia cinética E_C da partícula se calcula como

$$E_C = E_T + E_R. \tag{2.19}$$

Por sua vez, a energia potencial gravitacional E_P da partícula com relação ao ponto de referência P em um campo de gravidade constante \mathbf{g} é dada por

$$E_P = -m \langle \mathbf{g}, \mathbf{r} - \overrightarrow{OP} \rangle,$$

e, para o caso recorrente em que $\mathbf{g} = -g \cdot \mathbf{j}$ e o ponto de referência é a própria origem do sistema, essa expressão se simplifica para

$$E_P = m \cdot g \cdot r_y.$$

Com isso, a energia mecânica E_M da partícula é dada por

$$E_M = E_C + E_P. \quad (2.20)$$

2.3.2 Modelos de Força Normal

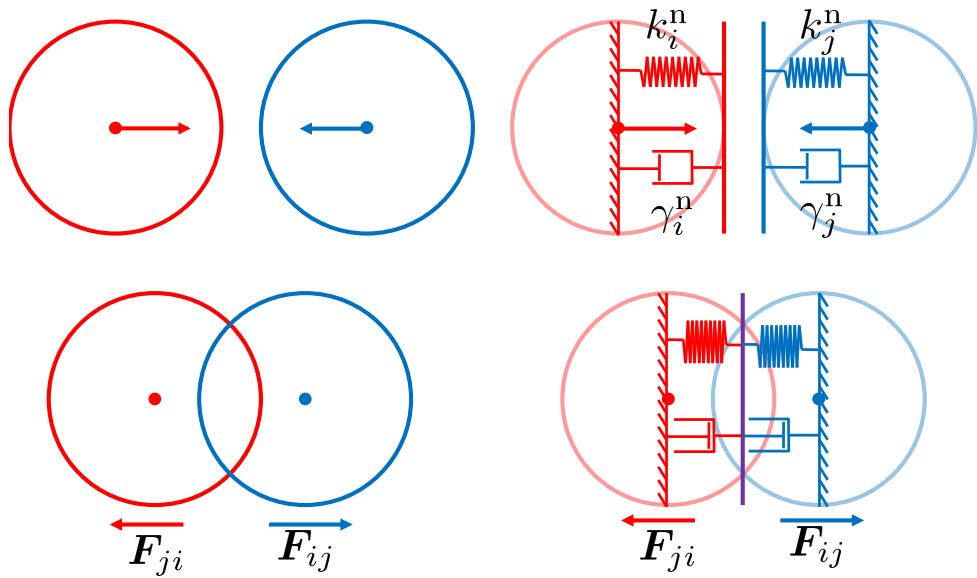
Nesta seção, são apresentados os dois principais modelos de força normal utilizados nas simulações de elementos discretos. Ambos os modelos buscam determinar

$$\mathbf{F}_{ji}^n = F_{ji}^n \cdot \hat{\mathbf{e}}_{ji}^n.$$

Modelo de Amortecedor Linear

O modelo de amortecedor linear é um dos modelos mais simples que se adotam para a colisão de partículas. Esse modelo pode ser representado esquematicamente como na [figura 2.4](#).

Figura 2.4 – Representação esquemática do modelo de amortecimento linear.



Fonte: do Autor.

A equação para a força normal dada por esse modelo é

$$F_{ji}^n = k_{ij}^n \cdot \xi_{ij} + \gamma_{ij}^n \cdot \dot{\xi}_{ij}, \quad (2.21)$$

em que

- k_{ij}^n é a constante elástica efetiva para o par de partículas i e j . Esse valor efetivo é calculado pela associação em série de molas elásticas, do que resulta que

$$k_{ij}^n = \left(\frac{1}{k_i^n} + \frac{1}{k_j^n} \right)^{-1},$$

sendo k_i^n e k_j^n as constantes elásticas das partículas i e j , respectivamente.

A constante elástica representa a rigidez das partículas, sendo que elementos com maior constante elástica resistem mais à penetração. Essa constante é uma propriedade que se associa a cada partícula, podendo ser modelada como função das propriedades de material, obtida experimentalmente ou simplesmente arbitrada ([SAMPAIO, 2017](#)).

- γ_{ij}^n é a constante de amortecimento efetiva para o par (i, j) , calculada através da associação em série de amortecedores de constantes γ_i^n e γ_j^n :

$$\gamma_{ij}^n = \left(\frac{1}{\gamma_i^n} + \frac{1}{\gamma_j^n} \right)^{-1}.$$

A constante de amortecimento corresponde à dissipação de energia na colisão. Maiores amortecimentos indicam maior perda de energia no choque. Essa constante é atribuída a cada partícula como se fosse uma propriedade de material. Essa atribuição, porém, muitas vezes não é possível, e se torna necessária a execução de experimentos para a determinação do valor dessa constante.

A partir da equação ([2.21](#)) e da primeira equação de movimento ([2.4](#)), é possível derivar-se uma expressão para o coeficiente de restituição na direção normal. Para um par de partículas esféricas, a massa efetiva é definida como

$$m_{ij} = (m_i^{-1} + m_j^{-1})^{-1},$$

e, para o caso em que a i -ésima partícula é esférica e a j -ésima representa uma parede plana,

$$m_{ij} = m_i.$$

Com isso, o coeficiente de restituição normal do par de partículas fica, de acordo com [Schwager e Pöschel \(2005\)](#),

$$\varepsilon_{ij}^n = \exp \left(-\frac{\pi \gamma_{ij}^n}{2m_{ij}} \sqrt{\frac{k_{ij}^n}{m_{ij}} - \left(\frac{\gamma_{ij}^n}{2m_{ij}} \right)^2} \right).$$

Percebe-se dessa expressão que o modelo de amortecedor linear implica um coeficiente de restituição normal independente da velocidade relativa entre as partículas, o que não representa bem o comportamento real de sistemas, como afirmado na [seção 2.3.1](#). Entretanto, justamente devido à simplicidade da expressão, esse modelo é utilizado na teoria de gases granulares, em simulações de Dinâmica Molecular e na validação de métodos numéricos.

Modelo de Esferas Viscoelásticas

Embora a equação (2.21) represente de maneira simplificada a força entre duas partículas colidentes, ela captura os dois principais mecanismos de interação, que são o termo elástico, representado por $F_{ji,\text{el}}^n = k_{ij}^n \cdot \xi_{ij}$, e o termo dissipativo, $F_{ji,\text{diss}}^n = \gamma_{ij}^n \cdot \dot{\xi}_{ij}$.

O modelo para esferas viscoelásticas tem a finalidade de expressar de forma mais fidedigna esses dois termos.

[Hertz \(1881\)](#) propôs primeiramente um modelo para a componente elástica da força, obtendo

$$F_{ji,\text{el}}^n = \frac{4}{3} Y_{ij} \sqrt{R_{ij}} \cdot \xi_{ij}^{3/2},$$

sendo que

- Y_{ij} é a elasticidade efetiva do par. Essa constante é definida por

$$Y_{ij} = \left(\frac{1 - \nu_i^2}{Y_i} + \frac{1 - \nu_j^2}{Y_j} \right)^{-1},$$

em que ν_i e ν_j são os coeficientes de Poisson das partículas i e j , e Y_i e Y_j são os respectivos módulos de elasticidade.

- R_{ij} é o raio efetivo do par de partículas. Esse raio é definido como

$$R_{ij} = (R_i^{-1} + R_j^{-1})^{-1}.$$

Esse modelo foi posteriormente estendido por [Brilliantov et al. \(1996\)](#) com a adição do termo dissipativo:

$$F_{ji,\text{diss}}^n = \frac{4}{3} Y_{ij} \sqrt{R_{ij}} \cdot A_{ij} \dot{\xi}_{ij} \sqrt{\xi_{ij}},$$

em que

- A_{ij} é a constante dissipativa efetiva do par. Essa constante é definida como a média aritmética das constantes dissipativas de cada partícula, isto é,

$$A_{ij} = \frac{A_i + A_j}{2}.$$

De acordo com [Schwager e Pöschel \(2005, p. 20\)](#), não é possível relacionar a constante dissipativa a propriedades de material. Essa constante, porém, pode ser calculada em função do coeficiente de atrito entre as partículas, e este, por sua vez, pode ser determinado experimentalmente.

Com isso, a expressão para a força normal, segundo o modelo de esferas viscoelásticas, fica

$$F_{ji}^n = \frac{4}{3} Y_{ij} \sqrt{R_{ij}} \cdot (\xi_{ij}^{3/2} + A_{ij} \dot{\xi}_{ij} \sqrt{\xi_{ij}}).$$

O coeficiente de restituição normal, nesse caso, é uma função da velocidade normal relativa no início do choque. Esse valor se expressa na equação

$$\begin{aligned}\varepsilon_{ij}^n(v_{ij}^n) &= 1 + C_1 A_{ij} \rho^{2/5} (v_{ij}^n)^{1/5} + C_2 A_{ij}^2 \rho^{4/5} (v_{ij}^n)^{2/5} + C_3 A_{ij}^3 \rho^{6/5} (v_{ij}^n)^{3/5} + \dots \\ &= 1 + \sum_{m=1}^{+\infty} C_m A_{ij}^m \rho^{2m/5} (v_{ij}^n)^{m/5},\end{aligned}$$

sendo que as constantes C_m são bem determinadas (SCHWAGER; PÖSCHEL, 2005, p. 143) e

$$\rho := \frac{4 \cdot Y_{ij}}{3} \cdot \frac{\sqrt{R_{ij}}}{m_{ij}}.$$

O modelo de esferas viscoelásticas se aproxima melhor dos resultados experimentais que o modelo de amortecedor linear, sendo por isso mais utilizado em simulações de sistemas reais (SCHWAGER; PÖSCHEL, 2005, p. 143).

Duração das Colisões

Um problema que se encontra com a definição dos modelos de amortecedor linear e de esferas viscoelásticas é a inversão de sinal da força.

No início do choque, tanto a superposição quanto a sua derivada são positivas. Isso implica, de acordo com as expressões para a força, que ocorre repulsão entre as partículas. A penetração entre as partículas aumenta gradativamente até que, em determinado instante, atinge um valor máximo. Devido à força normal repulsiva, inicia-se o fenômeno de descompressão, em que as partículas se afastam. Nessa etapa, a derivada da superposição é negativa. É possível, dependendo dos parâmetros da simulação, que essa derivada negativa resulte em uma força normal *atrativa*. Essa atração se inicia quando $\dot{\xi}_{ij}$ atinge um valor crítico $\dot{\xi}_{ij}^*$, em que a força calculada é nula.

No entanto a atração entre partículas geralmente não condiz com resultados experimentais. Uma das técnicas para eliminar esse fenômeno é limitarem-se inferiormente as forças calculadas. Esse artifício justifica-se pois, embora tenha sido feita a hipótese de que as partículas sejam corpos rígidos, partículas reais são deformáveis, e essa deformação pode ser suficiente para encerrar a colisão em um tempo anterior àquele considerado pelo modelo de partículas rígidas. Isso é representado na figura 2.5.

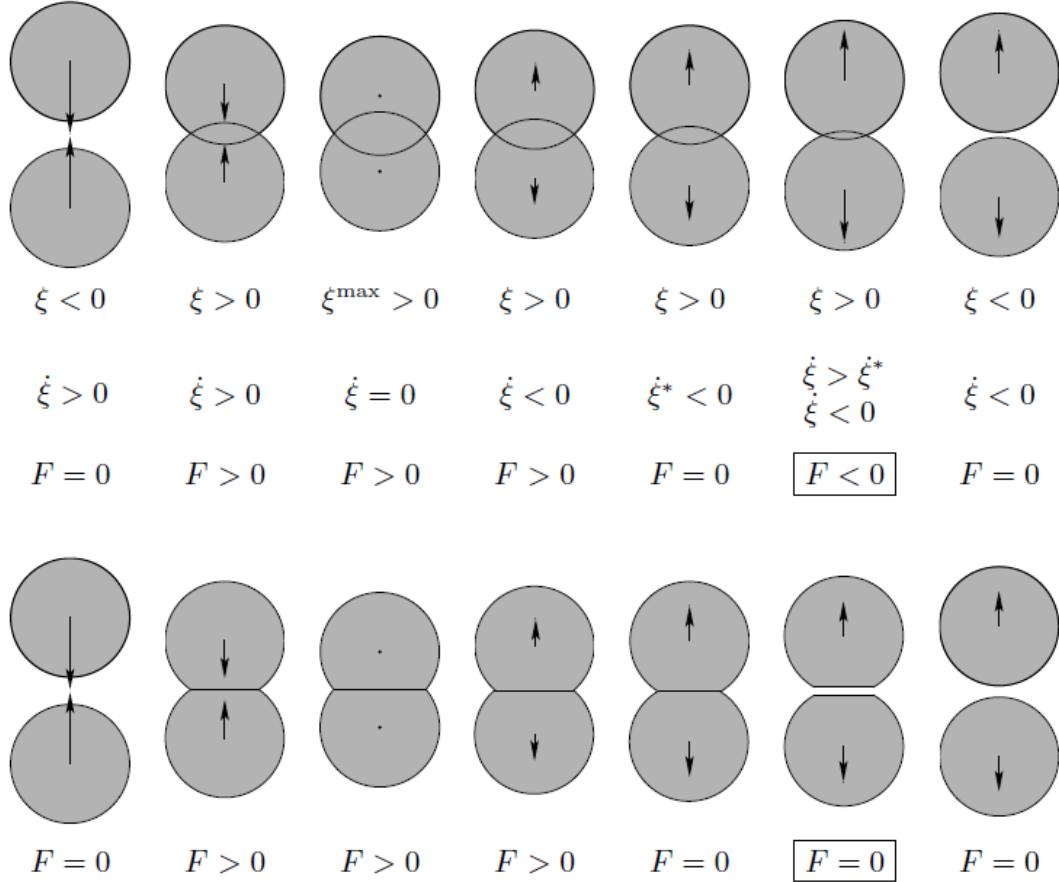
Para o amortecedor linear, isso significa que

$$F_{ji}^n = \max \left\{ k_{ij}^n \cdot \xi_{ij} + \gamma_{ij}^n \cdot \dot{\xi}_{ij}, 0 \right\}, \quad (2.22)$$

enquanto, para o modelo de esferas viscoelásticas, essa correção resulta em

$$F_{ji}^n = \max \left\{ \frac{4}{3} Y_{ij} \sqrt{R_{ij}} \cdot \left(\xi_{ij}^{3/2} + A_{ij} \dot{\xi}_{ij} \sqrt{\xi_{ij}} \right), 0 \right\}.$$

Figura 2.5 – Representação da colisão entre partículas. Na primeira linha, a avaliação das forças segundo os modelos não corrigidos. Na segunda linha, a colisão com forças limitadas à repulsão.



Fonte: Adaptado de [Schwager e Pöschel \(2005, p. 22\)](#).

Como consequência dessa mudança, o coeficiente de restituição normal tem seu valor alterado. [Schwager e Pöschel \(2007\)](#) demonstram que, para o modelo de amortecedor linear, o coeficiente de restituição normal fica

$$\varepsilon_{ij}^n = \begin{cases} \exp \left[-\frac{\beta_{ij}}{\omega_{ij}} \left(\pi - \arctan \frac{2 \beta_{ij} \omega_{ij}}{\omega_{ij}^2 - \beta_{ij}^2} \right) \right], & \text{se } \beta_{ij} < \frac{\omega_{ij}^*}{\sqrt{2}} \\ \exp \left[-\frac{\beta_{ij}}{\omega_{ij}} \arctan \frac{2 \beta_{ij} \omega_{ij}}{\omega_{ij}^2 - \beta_{ij}^2} \right], & \text{se } \beta_{ij} \in \left[\frac{\omega_{ij}^*}{\sqrt{2}}, \omega_{ij}^* \right] \\ \exp \left[-\frac{\beta_{ij}}{\Omega_{ij}} \ln \frac{\beta_{ij} + \Omega_{ij}}{\beta_{ij} - \Omega_{ij}} \right], & \text{se } \beta_{ij} > \omega_{ij}^* \end{cases}, \quad (2.23)$$

em que

$$\begin{aligned} (\omega_{ij}^*)^2 &= k_{ij}^n / m_{ij}, & \beta_{ij} &= \gamma_{ij}^n / (2 m_{ij}), \\ \omega_{ij} &= \sqrt{(\omega_{ij}^*)^2 - \beta_{ij}^2}, & \Omega_{ij} &= \sqrt{\beta_{ij}^2 - (\omega_{ij}^*)^2}. \end{aligned}$$

Uma expressão para o coeficiente de restituição corrigido para o modelo de esferas

viscoelásticas pode ser encontrada em Schwager e Pöschel (2008).

2.3.3 Modelos de Força Tangencial

Modelos para forças tangenciais, diferentemente dos modelos para forças normais, têm caráter intrinsecamente dinâmico, resistindo sempre ao movimento relativo entre as partículas. Esses modelos são responsáveis por fornecer um valor de força tangencial

$$\mathbf{F}_{ji}^t = F_{ji}^t \hat{\mathbf{e}}_{ji}^t.$$

O fato de que essa força resiste ao movimento é representado pela desigualdade

$$F_{ji}^t \leq 0.$$

Ainda, pela lei de Coulomb, a força tangencial não pode superar, em módulo, a força de atrito dinâmico

$$\mathbf{F}_{ji}^f = -\mu_{ij} F_{ji}^n \cdot \hat{\mathbf{e}}_{ji}^t,$$

em que μ_{ij} é o coeficiente de atrito dinâmico entre o par de partículas \mathcal{P}_i e \mathcal{P}_j . Disso segue a desigualdade

$$F_{ji}^t \geq -\mu_{ij} F_{ji}^n. \quad (2.24)$$

Ao se determinar a força tangencial, a componente de torque \mathbf{M}_{ji} que a partícula j aplica sobre a i pode ser calculada como

$$\mathbf{M}_{ji} = \boldsymbol{\varrho}_{ij} \times \mathbf{F}_{ji}^t.$$

Nesta seção, são apresentados os dois principais modelos de força tangencial, de acordo com Schwager e Pöschel (2005).

Modelo de Haff e Werner

O modelo de Haff e Werner já foi aplicado com sucesso em diversas simulações. Esse modelo é o análogo tangencial do modelo de amortecedor linear para a força normal (SCHWAGER; PÖSCHEL, 2005).

Para partículas esféricas, supondo que há contato, a equação (2.17) sugere a analogia

$$\dot{\xi}_{ij} \rightarrow v_{ij}^t.$$

Além disso, devido ao caráter dinâmico da força tangencial, desconsideram-se componentes elásticas.

Com isso e com a restrição da lei de Coulomb (2.24), o análogo tangencial da equação (2.21) fica

$$F_{ji}^t = -\min \left\{ \gamma_{ij}^t \cdot v_{ij}^t, \mu_{ij} \cdot F_{ji}^n \right\}, \quad (2.25)$$

em que

- γ_{ij}^t é a constante de amortecimento tangencial efetiva do par i e j . Essa constante pode ser obtida por uma associação em série de amortecedores de constantes γ_i^t e γ_j^t , cada um pertencente a uma das partículas, como

$$\gamma_{ij}^t = \left(\frac{1}{\gamma_i^t} + \frac{1}{\gamma_j^t} \right)^{-1}.$$

- μ_{ij} é o fator de atrito para o par (i,j) . Embora esse fator seja definido para pares de partículas, é possível atribuirem-se fatores μ_i e μ_j , um a cada partícula, de modo que

$$\mu_{ij} = \min \{\mu_i, \mu_j\}.$$

Com isso, o modelo de Haff e Werner assemelha-se a um modelo de amortecedor linear, exceto pelos fatos de haver a limitação da lei de Coulomb e de se desconsiderar a componente elástica.

A dificuldade apresentada por esse modelo é que não há propriedades de material mensuráveis a partir dos quais se possa extrair os valores de γ_i^t e γ_j^t . Esses parâmetros só são conhecidos posteriormente, com a comparação entre os resultados das simulações e experimentos.

Modelo de Cundall e Strack

Ao contrário do modelo de Haff e Werner, no modelo de Cundall e Strack supõe-se a existência de uma mola que resiste ao deslocamento relativo entre as partículas na direção tangencial. Esse modelo foi proposto por [Cundall e Strack \(1979, p. 52\)](#). Considera-se que a mola esteja na posição de equilíbrio no instante de início da colisão t_{ij} , e a sua deformação ou elongação no instante t é dada por

$$\zeta_{ij}(t) = \int_{t_{ij}}^t v_{ij}^t(\tau) d\tau.$$

Com isso, e considerando a restrição da lei de Coulomb, a força calculada pelo modelo de Cundall e Strack fica

$$F_{ji}^t = - \min \left\{ \kappa_{ij}^t \zeta_{ij}, \mu_{ij} \cdot F_{ji}^n \right\},$$

sendo que

- κ_{ij}^t é a constante elástica da mola equivalente no choque. Essa mola pode ser tratada como a associação em série de molas de constantes κ_i^t e κ_j^t , cada uma pertencente à respectiva partícula:

$$\kappa_{ij}^t = \left(\frac{1}{\kappa_i^t} + \frac{1}{\kappa_j^t} \right)^{-1}.$$

Porém essas constantes, assim como o amortecimento no modelo de Haff e Werner, só são obtidas na comparação posterior das simulações com resultados experimentais.

2.4 Algoritmo de Gear

Com as equações diferenciais bem determinadas através do uso de modelos de interação, inicia-se a busca por sua solução. O algoritmo de Gear é um método de resolução de equações diferenciais ordinárias originado do trabalho de [Gear \(1971\)](#).

Gear considerou o problema de extrapolar funções sujeitas a equações diferenciais. Dada uma função y tal que os pontos $y(t), \dot{y}(t), \dots, y^{(k)}(t)$ existem e são bem conhecidos, o objetivo é determinar os valores de $y(t + \Delta t), \dot{y}(t + \Delta t), \dots, y^{(k)}(t + \Delta t)$ sabendo que y deve satisfazer uma equação diferencial da forma

$$y^{(p)} = f^y(y, \dot{y}, \dots, y^{(p-1)}, t) \quad (2.26)$$

com $p \leq k$.

O algoritmo consiste em duas etapas: uma etapa de *previsão* e uma de *correção*, detalhadas nas seções [2.4.1](#) e [2.4.2](#).

2.4.1 Etapa de Predição

Nos métodos computacionais desenvolvidos para a solução dessas equações, a extração de funções possui um papel fundamental por permitir a estimativa de valores além do conjunto previamente conhecido.

A etapa de predição é responsável por obter uma estimativa para $y(t + \Delta t), \dot{y}(t + \Delta t), \dots, y^{(k)}(t + \Delta t)$ através de uma extração a partir dos valores de $y(t), \dot{y}(t), \dots, y^{(k)}(t)$.

Para simplificação da notação, dada uma função y , define-se o vetor das k primeiras derivadas de y como a função vetorial

$$\mathbf{y}^{[k]} = (y, \dot{y}, \ddot{y}, \dots, y^{(k)})$$

nos pontos em que todas as coordenadas estiverem definidas.

Conforme demonstrado por [Brezinski e Zaglia \(2013\)](#), métodos de extração lineares para uma função e suas derivadas podem ser escritos na forma

$$\begin{pmatrix} y_{\text{pr}} \\ \dot{y}_{\text{pr}} \\ \ddot{y}_{\text{pr}} \\ \vdots \\ y_{\text{pr}}^{(k-1)} \\ y_{\text{pr}}^{(k)} \end{pmatrix} = \begin{pmatrix} a_{0,0} & a_{0,1} & a_{0,2} & \cdots & a_{0,k-1} & a_{0,k} \\ a_{1,0} & a_{1,1} & a_{1,2} & \cdots & a_{1,k-1} & a_{1,k} \\ a_{2,0} & a_{2,1} & a_{2,2} & \cdots & a_{2,k-1} & a_{2,k} \\ \vdots & & & \ddots & & \vdots \\ a_{k-1,0} & a_{k-1,1} & a_{k-1,2} & \cdots & a_{k-1,k-1} & a_{k-1,k} \\ a_{k,0} & a_{k,1} & a_{k,2} & \cdots & a_{k,k-1} & a_{k,k} \end{pmatrix} \cdot \begin{pmatrix} y \\ \dot{y} \\ \ddot{y} \\ \vdots \\ y^{(k-1)} \\ y^{(k)} \end{pmatrix}.$$

ou, mais simplesmente,

$$\mathbf{y}_{\text{pr}}^{[k]} = \underline{\mathbf{A}}_k \cdot \mathbf{y}^{[k]}. \quad (2.27)$$

em que a matriz \underline{A}_k é determinada pelo método escolhido e $\mathbf{y}_{\text{pr}}^{[k]}$ é o vetor de derivadas de y *predito*.

Dentre os métodos de extrapolação mais utilizados estão o método de expansão de Taylor, o método de Richardson, o método de interpolação de Aitken e os métodos de Runge-Kutta, cada qual com diferentes características em termos de exatidão e estabilidade (GEAR, 1971).

O método de extrapolação por expansão de Taylor combina simplicidade, exatidão e estabilidade, e por isso é o método adotado neste trabalho. Esse método é fundamentado pelo [Teorema de Taylor](#).

Teorema (Teorema de Taylor). *Seja y uma função de uma variável com derivadas $\dot{y}, \dots, y^{(k+1)}$ todas definidas em um intervalo aberto que contenha t , e seja $R_{k,t,y}$ definida pela equação*

$$y(t + \Delta t) = y(t) + \dot{y}(t) \cdot \Delta t + \dots + \frac{y^{(k)}(t)}{k!} \cdot \Delta t^k + R_{k,t,y}(\Delta t).$$

Então

$$\lim_{\Delta t \rightarrow 0} \frac{R_{k,t,y}(\Delta t)}{\Delta t^k} = 0. \quad (2.28)$$

Uma versão mais completa desse teorema é apresentada e demonstrada por Spivak (1994).

A função $R_{k,t,y}$ é o resto de ordem k para a função y no entorno de t . A equação (2.28) indica que o resto é um termo da ordem de Δt^{k+1} , e motiva a aproximação

$$y(t + \Delta t) \cong y(t) + \dot{y}(t) \cdot \Delta t + \dots + \frac{y^{(k)}(t)}{k!} \cdot \Delta t^k. \quad (2.29)$$

A equação (2.29) é o truncamento da expansão de Taylor na k -ésima derivada.

Considerando uma função vetorial $\mathbf{y} : I \subseteq \mathbb{R} \rightarrow \mathbb{R}^m$, o [Teorema de Taylor](#) pode ser aplicado a cada uma de suas funções coordenadas¹, resultando em uma expansão similar à da equação (2.29), desde que as hipóteses do teorema sejam satisfeitas. Os casos de interesse são $m = 1$, para funções reais; $m = 2$, para vetores bidimensionais como a posição de uma partícula em uma simulação em duas dimensões; $m = 3$, para simulações em três dimensões; e $m = 4$ para algumas parametrizações para a orientação das partículas.

Assim, o [Teorema de Taylor](#) permite a estimativa do valor de uma função em um ponto $t + \Delta t$ a partir do valor da função e de suas derivadas em um ponto t , e essa estimativa é tanto melhor quanto menor for o valor de Δt .

¹ Escrevendo $\mathbf{y}(t) = (y_1(t), \dots, y_m(t))$, a i -ésima função coordenada de \mathbf{y} é a função y_i .

Não somente a função pode ser prevista, mas suas derivadas também. Para a j -ésima derivada de y :

$$y_{\text{pr}}^{(j)}(t + \Delta t) = y^{(j)}(t) + \cdots + \frac{\Delta t^{k-j}}{(k-j)!} \cdot y^{(k-j)}(t).$$

Com isso, é possível escrever

$$\begin{pmatrix} y_{\text{pr}} \\ \dot{y}_{\text{pr}} \\ \ddot{y}_{\text{pr}} \\ \vdots \\ y_{\text{pr}}^{(k-1)} \\ y_{\text{pr}}^{(k)} \end{pmatrix} = \begin{pmatrix} 1 & \Delta t & \frac{\Delta t^2}{2} & \cdots & \frac{\Delta t^{k-1}}{(k-1)!} & \frac{\Delta t^k}{k!} \\ 0 & 1 & \Delta t & \cdots & \frac{\Delta t^{k-2}}{(k-2)!} & \frac{\Delta t^{k-1}}{(k-1)!} \\ 0 & 0 & 1 & \cdots & \frac{\Delta t^{k-3}}{(k-3)!} & \frac{\Delta t^{k-2}}{(k-2)!} \\ \vdots & & & \ddots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 & \Delta t \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} y \\ \dot{y} \\ \ddot{y} \\ \vdots \\ y^{(k-1)} \\ y^{(k)} \end{pmatrix}. \quad (2.30)$$

A matriz da equação (2.30) é a matriz de extração do método de expansão de Taylor, sendo a mesma para a extração de ordem k de qualquer função.

Esse método ainda pode ser usado para aproximar as funções de posição e de orientação da partícula, assim como outros graus de liberdade que o problema porventura possua.

Todavia essa previsão geralmente não é exata. Uma das razões para isto é que o truncamento da expansão de Taylor, ou qualquer outro método de extração que se use, despreza a função resto, que não é necessariamente nula. Mesmo assim, essa diferença é aceitável quando se utilizam passos de tempo suficientemente pequenos.

A principal fonte de erros da equação (2.30) é que não se considera, em nenhum momento, equações diferenciais a que a função y pode estar restrita. Por exemplo, para a posição ou a orientação de uma partícula, pode haver a ação de forças e torques atuantes entre os instantes t e $t + \Delta t$. É necessário, então, *corrigir* a função prevista. Essa correção pode ser feita na etapa de correção, apresentada na seção 2.4.2.

2.4.2 Correção

Na etapa de correção, um termo é adicionado ao vetor previsto $\mathbf{y}_{\text{pr}}^{[k]}$ para se obter o vetor $\mathbf{y}_{\text{corr}}^{[k]}$, corrigido em função da equação diferencial (2.26).

O valor corrigido para $y^{(p)}$ pode ser obtido no instante $t + \Delta t$ diretamente através da equação:

$$y_{\text{corr}}^{(p)}(t + \Delta t) = f^y(y_{\text{pr}}, \dot{y}_{\text{pr}}, \dots, y_{\text{pr}}^{(p-1)}, t + \Delta t),$$

e assim é conhecido o valor do erro $\Delta y^{(p)} := y_{\text{corr}}^{(p)} - y_{\text{pr}}^{(p)}$ no ponto $t + \Delta t$.

Como demonstrado por Gear (1971), as demais derivadas de y podem ser corrigidas em função de $\Delta y^{(p)}$ de acordo com a equação

$$\begin{pmatrix} y_{\text{corr}} \\ \dot{y}_{\text{corr}} \\ \ddot{y}_{\text{corr}} \\ \vdots \\ y_{\text{corr}}^{(p)} \\ \vdots \\ y_{\text{corr}}^{(k-1)} \\ y_{\text{corr}}^{(k)} \end{pmatrix} = \begin{pmatrix} y_{\text{pr}} \\ \dot{y}_{\text{pr}} \\ \ddot{y}_{\text{pr}} \\ \vdots \\ y_{\text{pr}}^{(p)} \\ \vdots \\ y_{\text{pr}}^{(k-1)} \\ y_{\text{pr}}^{(k)} \end{pmatrix} + \begin{pmatrix} c_{p,0} \\ c_{p,1} \frac{1}{\Delta t} \\ c_{p,2} \frac{2}{\Delta t^2} \\ \vdots \\ c_{p,p} \frac{p!}{\Delta t^p} \\ \vdots \\ c_{p,k-1} \frac{(k-1)!}{\Delta t^{k-1}} \\ c_{p,k} \frac{k!}{\Delta t^k} \end{pmatrix} \cdot \frac{\Delta t^p}{p!} \Delta y^{(p)}, \quad (2.31)$$

sendo que as constantes corretoras $c_{p,0}, c_{p,1}, \dots, c_{p,k}$ dependem da ordem k da maior derivada considerada e da ordem p da equação diferencial. Alguns dos valores dessas constantes são apresentados na tabela 2.1. Em todos casos, sempre se tem que $c_{p,p} = 1$, pois $y_{\text{corr}}^{(p)} = y_{\text{pr}}^{(p)} + \Delta y^{(p)}$.

Por simplicidade, escreve-se

$$\mathbf{c}_{p,k} = \left(c_{p,0}, c_{p,1} \frac{1}{\Delta t}, c_{p,2} \frac{2}{\Delta t^2}, \dots, c_{p,k} \frac{k!}{\Delta t^k} \right),$$

e então a equação (2.31) pode ser reescrita como

$$\mathbf{y}_{\text{corr}}^{[k]} = \mathbf{y}_{\text{pr}}^{[k]} + \mathbf{c}_{p,k} \cdot \frac{\Delta t^p}{p!} \Delta y^{(p)}. \quad (2.32)$$

Ainda, de acordo com Gear (1971, p. 153), o erro global do algoritmo é da ordem de $\Delta t^{k+1+q-p}$, em que q é o maior inteiro tal que a equação diferencial (2.26) pode ser escrita como

$$y^{(p)} = f^y(y, \dot{y}, \dots, y^{(p-q)}, t). \quad (2.33)$$

E, com isso, a etapa de correção está bem definida. Em virtude das características apresentadas, o algoritmo de Gear é comumente usado como método de resolução das equações de movimento dos elementos dos sistemas de partículas.

Com as considerações feitas neste capítulo, todas as ferramentas matemáticas necessárias para a solução do problema já estão expostas. A próxima etapa, então, é desenvolver um método numérico que permita a implementação desse algoritmo de solução.

Tabela 2.1 – Constantes corretoras para o algoritmo de Gear em função da ordem k da maior derivada considerada e da ordem p da equação diferencial.

p	k	$c_{p,0}$	$c_{p,1}$	$c_{p,2}$	$c_{p,3}$	$c_{p,4}$	$c_{p,5}$	$c_{p,6}$	$c_{p,7}$
1	2	$\frac{5}{12}$	1	$\frac{1}{2}$
	3	$\frac{3}{8}$	1	$\frac{3}{4}$	$\frac{1}{6}$
	4	$\frac{251}{720}$	1	$\frac{11}{12}$	$\frac{1}{3}$	$\frac{1}{24}$
	5	$\frac{95}{288}$	1	$\frac{25}{24}$	$\frac{35}{72}$	$\frac{5}{48}$	$\frac{1}{120}$
	6	$\frac{19087}{60480}$	1	$\frac{137}{120}$	$\frac{5}{8}$	$\frac{17}{96}$	$\frac{1}{40}$	$\frac{1}{720}$..
	7	$\frac{5257}{17280}$	1	$\frac{49}{40}$	$\frac{203}{270}$	$\frac{49}{192}$	$\frac{7}{144}$	$\frac{7}{1440}$	$\frac{1}{5040}$
	3	$\frac{1}{6}$	$\frac{5}{6}$	1	$\frac{1}{3}$
2	4	$\frac{19}{120}$	$\frac{3}{4}$	1	$\frac{1}{2}$	$\frac{1}{12}$
	5	$\frac{3}{20}$	$\frac{251}{360}$	1	$\frac{11}{18}$	$\frac{1}{6}$	$\frac{1}{60}$
	6	$\frac{863}{6048}$	$\frac{665}{1008}$	1	$\frac{25}{36}$	$\frac{35}{144}$	$\frac{1}{24}$	$\frac{1}{360}$..
	7	$\frac{1925}{14112}$	$\frac{19087}{30240}$	1	$\frac{137}{180}$	$\frac{5}{16}$	$\frac{17}{240}$	$\frac{1}{120}$	$\frac{1}{2520}$
	4	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{5}{4}$	1	$\frac{1}{4}$
	5	$\frac{3}{80}$	$\frac{19}{40}$	$\frac{9}{8}$	1	$\frac{3}{8}$	$\frac{1}{20}$
	6	$\frac{221}{5040}$	$\frac{9}{20}$	$\frac{251}{240}$	1	$\frac{11}{24}$	$\frac{1}{10}$	$\frac{1}{120}$..
3	7	$\frac{2185}{46368}$	$\frac{863}{2016}$	$\frac{95}{96}$	1	$\frac{25}{48}$	$\frac{49}{336}$	$\frac{1}{48}$	$\frac{1}{840}$
	5	$\frac{1}{30}$	$\frac{1}{10}$	1	$\frac{5}{3}$	1	$\frac{1}{5}$
	6	$\frac{16}{630}$	$\frac{3}{20}$	$\frac{19}{20}$	$\frac{3}{2}$	1	$\frac{3}{10}$	$\frac{1}{30}$..
	7	$\frac{11}{630}$	$\frac{221}{1260}$	$\frac{9}{10}$	$\frac{251}{180}$	1	$\frac{11}{30}$	$\frac{1}{15}$	$\frac{1}{210}$

Fonte: [Gear \(1971\)](#).

3 O Método de Elementos Discretos

O Método de Elementos Discretos (DEM¹) refere-se a uma família de métodos numéricos aplicados na simulação de sistemas de partículas. Esses métodos compartilham diversas características entre si, como o monitoramento da vizinhança de cada partícula, o cálculo explícito das variáveis do problema a partir de seus valores nos instantes anteriores, dentre outras.

Este capítulo é dedicado à apresentação e à explicação dessas características. São apresentados o funcionamento geral de um algoritmo DEM, o procedimento de solução das equações do problema e as principais etapas do método, além de considerações adicionais acerca de métodos de monitoramento de vizinhança e acoplamento do DEM com outras famílias de métodos computacionais.

3.1 Características Gerais do Método

Segundo Bićanić (2007, p. 1), o DEM constitui-se de técnicas de modelagem computacional indicadas para a simulação do comportamento dinâmico de conjuntos de partículas de geometria arbitrária sujeitas a restrições de contato variantes no tempo.

Conforme a definição apresentada por Cundall e Hart (1989 apud BIĆANIĆ, 2007), métodos de elementos discretos são métodos computacionais que:

- a) consideram deslocamentos e rotações finitos de corpos discretos;
- b) reconhecem novos contatos automaticamente à medida que a simulação progride.

Cada partícula é considerada um corpo rígido com seis graus de liberdade: três translações e três rotações, e está sujeita às equações diferenciais de movimento descritas na seção 2.2. O processo de simulação consiste na solução dessas equações.

Em um sistema, porém, os elementos interagem entre si, e as forças e os torques atuantes sobre eles dependem dessas interações. Sendo assim, é necessário o *monitoramento das vizinhanças*, isto é, o método deve sempre determinar quais são os pares de elementos que interagem em cada passo de tempo. Dentre os algoritmos disponíveis, destacam-se o algoritmo de Verlet, dedicado a partículas esféricas, o algoritmo *link cell* e o algoritmo *lattice*.

Os elementos da simulação ainda podem pertencer a *tipos* distintos. Por exemplo, é possível distinguirem-se *partículas*, elementos cujas equações de movimento precisam ser resolvidas, de *elementos de contorno*, que têm seu estado conhecido já na etapa de

¹ Do inglês, *Discrete Element Method*.

inicialização. Além dos elementos de contorno, outras *condições de contorno* podem ser aplicadas ao sistema, como restrições de movimento, paredes reflexivas e condição de repetição.

De acordo com Bićanić (2007, p. 2), existem diversas metodologias para a simulação com elementos discretos, sendo que as variações ocorrem:

- a) nos algoritmos de detecção de contatos;
- b) nas formulações para as interações;
- c) nas condições de contorno;
- d) na consideração de modelos de fratura, fragmentação ou aglutinação;
- e) nos métodos de integração das equações;
- f) nos algoritmos de monitoramento de vizinhanças.

3.1.1 Discretização Temporal

Como é recorrente em métodos de simulação numérica, o DEM conta com a discretização do tempo para a solução das equações de movimento. Para o intervalo de simulação $[t^i, t^f]$, sendo t^i o instante inicial e t^f , o final, uma discretização temporal é um conjunto $\{t_0, t_1, \dots, t_N\}$ tal que

$$t^i = t_0 < t_1 < \dots < t_{N-1} < t_N = t^f.$$

O valor N é o número de passos de tempo da simulação. Um passo de tempo é o período compreendido entre dois instantes consecutivos da discretização. A duração do n -ésimo passo de tempo é dada então por

$$\Delta t_n := t_n - t_{n-1},$$

e escreve-se Δt no caso em que os passos de tempo têm todos a mesma duração.

Com isso, o objetivo do Método de Elementos Discretos torna-se obter a solução para as coordenadas das partículas no tempo discretizado.

3.1.2 Solução das Equações

A solução das equações governantes do sistema é feita conforme explicado na seção 2.2.3 aplicando algoritmos de integração como o algoritmo de Gear, descrito na seção 2.4.

Uma característica dos métodos de elementos discretos tradicionais é serem métodos explícitos, ou seja, o estado do sistema de partículas no instante t_n é calculado a partir do estado em t_{n-1} que se supõe ser conhecido. Isso é evidente no algoritmo de Gear.

O procedimento de solução das equações é apresentado em maiores detalhes na seção 3.2.2.

3.1.3 Elementos da Simulação

A classificação dos elementos pertencentes à simulação tem a finalidade de possibilitar simplificações, de determinar os métodos de busca de vizinhança, de busca de contato e de integração das equações, e de reduzir o custo computacional do algoritmo.

Não existe uma divisão comum a todos os métodos de elementos discretos, mas podem se observar classificações baseadas na geometria, nas propriedades, na forma como as coordenadas são obtidas e nas restrições aplicadas.

Toda simulação possui um *conjunto universo* \mathcal{U} que contém todas as entidades consideradas pelo algoritmo: corpos, fluidos, campos de força, entre outras. Nos métodos de elementos discretos, os corpos rígidos pertencentes ao conjunto universo são agrupados no *conjunto de elementos* \mathcal{E} . Por fim, é possível dividir-se o conjunto de elementos em um *conjunto de partículas* \mathcal{P} e um *conjunto de elementos de contorno* \mathcal{B} .

O conjunto de partículas é o conjunto de todos os elementos para os quais as equações de movimento devem ser resolvidas. Essa resolução é feita de acordo com os algoritmos de integração.

Os elementos de contorno, por outro lado, são elementos cujos estados já são conhecidos como funções explícitas do tempo desde a etapa de inicialização. Exemplo disso são os elementos fixos, cujas coordenadas são sempre iguais às coordenadas iniciais. Elementos de contorno, portanto, representam um alívio do custo computacional, pois não é necessária a resolução de suas equações de movimento, embora ainda sejam necessários no cálculo das interações com as partículas. Uma função que determina completamente um elemento em cada instante de tempo é chamada de *função de evolução*.

Campos de força, em geral, são considerados como entidades externas ao domínio da simulação, atuando indistintamente sobre todas as partículas do sistema e contribuindo com parcelas das forças externas $\mathbf{F}_i^{\text{ext}}$ e dos torques externos $\mathbf{M}_i^{\text{ext}}$.

Outras características que podem ser atribuídas aos elementos da simulação são:

- Propriedades:* as propriedades que se atribuem aos elementos são essenciais para o cálculo das interações que ocorrem entre eles. Seus valores podem ser função do material de que os corpos são compostos, de parâmetros obtidos experimentalmente, entre outros. Exemplos de propriedades físicas são a massa específica e o módulo de elasticidade. Além disso, outros parâmetros podem ser atribuídos às partículas, como o coeficiente de atrito e a constante de amortecimento, de acordo com a seção 2.3.

- b) *Geometria*: a geometria dos elementos determina fundamentalmente os modelos de interação a que os elementos são submetidos, os métodos de monitoramento de vizinhança e a busca por contatos. As geometrias mais simples, nas quais este trabalho se baseia, são a esférica e a plana. Também são aplicáveis superelipsoides e poliedros. Um exemplo da aplicação de superelipsoides pode ser encontrado em [Sampaio \(2017\)](#). Outras geometrias possíveis que podem ser citadas são as poliedrais arredondadas, as cilíndricas, as cilíndricas arredondadas e formas geométricas arbitrariamente mais complexas podem ser obtidas através da composição de partículas de forma mais simples.
- c) *Restrição de movimento*: o tipo de restrição de movimento que um elemento sofre também pode ser considerado. É usual definirem-se elementos *fixos*, cujas coordenadas se mantêm constantes na simulação, e elementos com movimento *prescrito*, que são aqueles cujas coordenadas são definidas por funções temporais arbitrárias fornecidas na etapa de inicialização. Outra restrição bastante aplicada é aquela que ocorre entre os subelementos de uma partícula composta, limitando o seu movimento relativo. Mais detalhes sobre essa restrição podem ser encontrados em [Schwager e Pöschel \(2005\)](#).

3.1.4 Algoritmo Geral do DEM

De forma geral, os métodos de elementos discretos apresentam uma estrutura básica composta, essencialmente, da etapa de inicialização, do laço de simulação e da terminação do programa.

O laço da simulação é a parte mais custosa do algoritmo pois é nele que ocorrem as subetapas de resolução das equações e exportação de dados.

A forma geral do DEM pode ser representada como o fluxograma exposto na [figura 3.1](#), cujas etapas são apresentadas em mais detalhes na [seção 3.2](#).

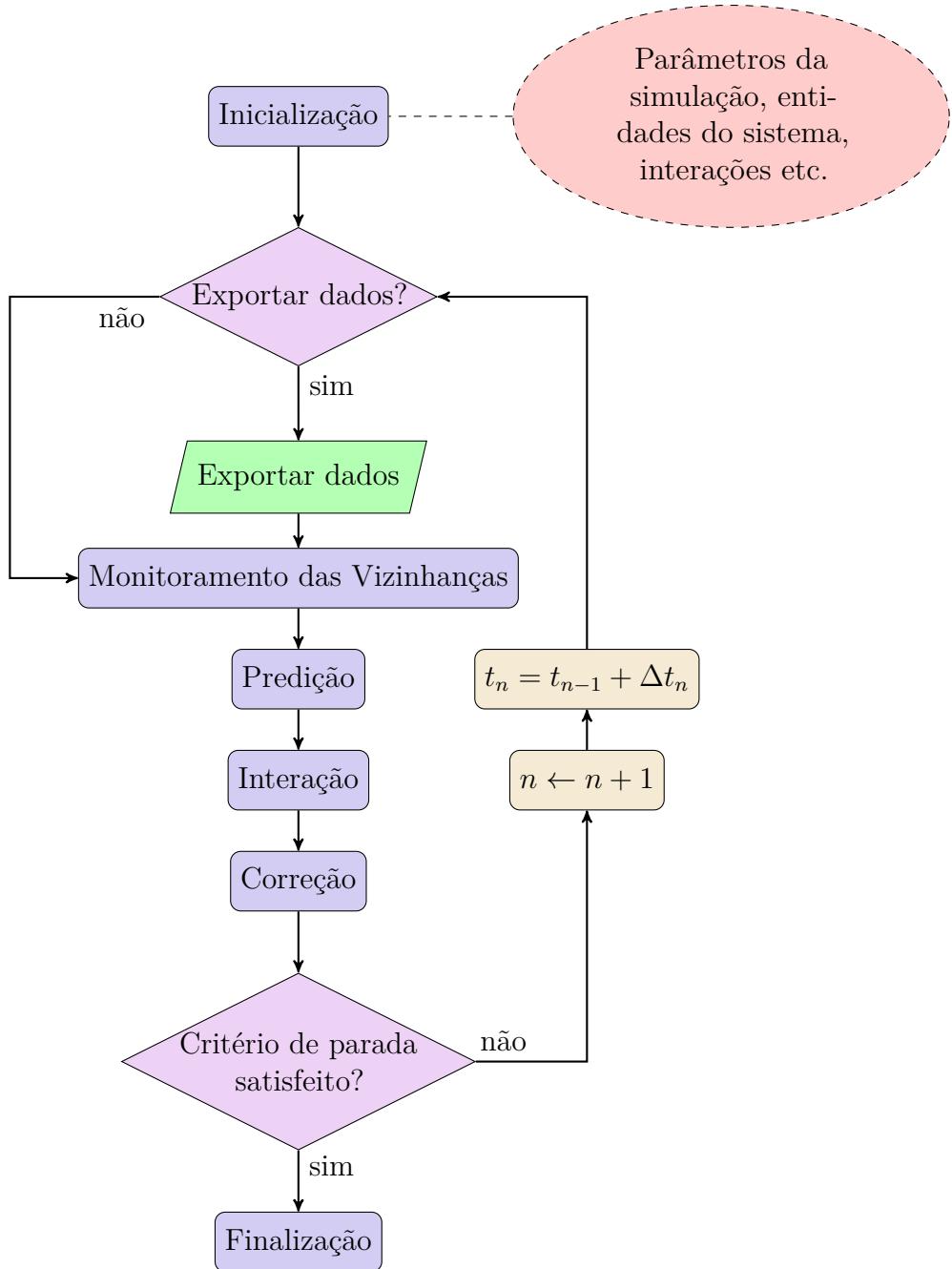
3.2 Etapas do Algoritmo

3.2.1 Inicialização do Sistema de Partículas

A primeira etapa em uma simulação DEM é a inicialização do sistema de partículas. Essa etapa consiste em se definir o ambiente da simulação e se construirão todos os elementos necessários.

De acordo com [Schwager e Pöschel \(2005\)](#), a inicialização é comumente feita por meio de um arquivo de entrada, muito embora programas de simulação geralmente apresentem uma interface gráfica com essa finalidade.

Figura 3.1 – Fluxograma do funcionamento geral do DEM.



Fonte: do Autor.

O resultado dessa etapa é o estado inicial do sistema de partículas, com todas as posições, orientações e outras variáveis inicializadas. Os parâmetros de entrada são:

- Parâmetros da simulação:* os principais parâmetros, como instantes inicial e final, passo de tempo, critério de parada e caminhos para exportação de arquivos, são configurados;
- Partículas:* todas as partículas do sistema devem ser construídas. A cada partícula devem ser atribuídos um vetor posição, uma orientação e propriedades

físicas iniciais. Além disso, as derivadas dos vetores posição e orientação devem ser informados. A geometria das partículas também deve ser definida;

- c) *Elementos de contorno*: de maneira semelhante às partículas, os elementos de contorno devem ser construídos com geometria, posição, orientação e propriedades físicas. Ainda, é necessário determinar-se a função de evolução desses elementos;
- d) *Outras entidades*: todas as outras entidades do sistema, como campos de força e fluidos, também devem ser inicializadas;
- e) *Interações*: além dos elementos do sistema, devem ser definidos os modelos de interação utilizados. Por exemplo, para as forças normais de colisão entre partículas esféricas, deve-se escolher entre o modelo de amortecimento linear, o de esferas viscoelásticas ou outros que estejam implementados no simulador;
- f) *Algoritmo de integração*: a determinação do algoritmo de solução das equações do sistema é fundamental para a simulação. Dentre os principais métodos aplicados se encontram aqueles citados na [seção 2.2.3](#): o algoritmo de Gear, o método *leapfrog*, os métodos de Verlet e os de Runge-Kutta, desde que estejam disponíveis no simulador;
- g) *Método de monitoramento de vizinhanças*: a escolha do método de monitoramento de vizinhanças também é de extrema importância na simulação. Esse método tem por finalidade determinar quais elementos interagem em cada passo de tempo, sendo determinante para o custo computacional do algoritmo. O monitoramento de vizinhanças deve ser escolhido tendo em vista os tipos e as formas geométricas dos elementos considerados e as características da simulação;
- h) *Outros parâmetros*: por fim, dependendo da implementação do simulador e do método de elementos discretos usado, podem ser necessárias outras entradas mais específicas, além das já citadas.

3.2.2 Solução das Equações

De acordo com [Schwager e Pöschel \(2005\)](#), a solução das equações pode ser dividida em quatro etapas, sendo elas:

1. *Determinação das Vizinhanças*: Nessa etapa, a vizinhança de cada partícula do sistema é construída (no início da simulação) ou atualizada (em instantes posteriores) com o intuito de limitar a quantidade de interações avaliadas no passo de tempo.
2. *Predição*: Nesse passo, o estado do sistema no instante t_n é predito em função do estado em t_{n-1} por meio de métodos de extração, como explicado na [seção 2.4.1](#).

3. *Interação*: A partir do estado predito do sistema, são computadas as interações entre cada partícula e os elementos de sua vizinhança.
4. *Correção*: Na etapa de correção, o estado do sistema no instante t_n é corrigido em função das interações calculadas na etapa 3 e das equações governantes do problema, como apresentado na [seção 2.4.2](#). Embora esse processo não seja exato, o DEM considera o estado corrigido como a solução do problema para o instante t_n .

Os passos 2, 3 e 4 são específicos do algoritmo de Gear, podendo mudar se forem adotados outros métodos.

Para a determinação da vizinhança no passo 1, são usados algoritmos de monitoramento de vizinhança, como explicado na [seção 3.3](#). O objetivo desta seção é, assim, apresentar os passos 2, 3 e 4, mostrando a aplicação dos conceitos apresentados no [capítulo 2](#) aos sistemas de partículas.

Para tanto, pode ser feita uma distinção entre o caso de partículas com geometria arbitrária, em que é necessário monitorar-se a orientação das partículas, e o de partículas esféricas, situação em que é possível aplicarem-se simplificações. O caso geral é apresentado no [apêndice A](#). Para partículas esféricas, as equações de movimento assumem uma forma mais simples.

Sendo t_n o instante de interesse e t_{n-1} , o instante anterior em que o sistema é conhecido, as posições e as orientações das partículas podem ser previstas como na [equação \(2.27\)](#). Escolhida uma ordem de extração k , e definido o conjunto \mathcal{S} das partículas esféricas do sistema, calculam-se

$$\begin{aligned}\mathbf{r}_{i,\text{pr}}^{[k]}(t_n) &= \underline{\mathbf{A}}_k(\Delta t_n) \cdot \mathbf{r}_i^{[k]}(t_{n-1}) \\ \boldsymbol{\omega}_{i,\text{pr}}^{[k-1]}(t_n) &= \underline{\mathbf{A}}_{k-1}(\Delta t_n) \cdot \boldsymbol{\omega}_i^{[k-1]}(t_{n-1})\end{aligned}, \quad \mathcal{P}_i \in \mathcal{S},$$

em que $\underline{\mathbf{A}}_k(\Delta t_n)$ e $\underline{\mathbf{A}}_{k-1}(\Delta t_n)$ são as matrizes de extração por expansão de Taylor de ordem k e $k - 1$ calculadas com o passo de tempo Δt_n como indicado na [seção 2.4.1](#).

A vizinhança de uma partícula, como explicado na [seção 3.3](#), é o conjunto de todas as entidades da simulação com as quais ela interage. A vizinhança da i -ésima partícula do sistema é denotada por \mathcal{N}_i . Cada elemento \mathcal{E}_j da vizinhança de \mathcal{P}_i contribui para a força \mathbf{F}_i e o torque \mathbf{M}_i resultantes sobre a mesma com parcelas \mathbf{F}_{ji} e \mathbf{M}_{ji} que são, por hipótese, dependentes apenas do par $(\mathcal{P}_i, \mathcal{E}_j)$, isto é,

$$\begin{aligned}\mathbf{F}_{ji} &\equiv \mathbf{F}_{ji}(\mathcal{P}_i, \mathcal{E}_j) \\ \mathbf{M}_{ji} &\equiv \mathbf{M}_{ji}(\mathcal{P}_i, \mathcal{E}_j)\end{aligned}, \quad \mathcal{P}_i \in \mathcal{S}, \quad \mathcal{E}_j \in \mathcal{N}_i.$$

Mais ainda, a partícula i pode sofrer com forças e torques provocados por entidades externas ao sistema, como forças de campo, cujas resultantes são dadas por

$$\begin{aligned}\mathbf{F}_i^{\text{ext}} &\equiv \mathbf{F}_i^{\text{ext}}(\mathcal{P}_i) \\ \mathbf{M}_i^{\text{ext}} &\equiv \mathbf{M}_i^{\text{ext}}(\mathcal{P}_i)\end{aligned}, \quad \mathcal{P}_i \in \mathcal{S}.$$

Com isso, obtém-se os vetores resultantes de acordo com o princípio da superposição dado pelas equações (2.1) e (2.2):

$$\begin{aligned}\mathbf{F}_i &= \sum_{\mathcal{E}_j \in \mathcal{N}_i} \mathbf{F}_{ji} + \mathbf{F}_i^{\text{ext}} \\ \mathbf{M}_i &= \sum_{\mathcal{E}_j \in \mathcal{N}_i} \mathbf{M}_{ji} + \mathbf{M}_i^{\text{ext}}\end{aligned}, \quad \mathcal{P}_i \in \mathcal{P}.$$

A cada um dos graus de liberdade da partícula é associada uma equação que determina seu comportamento, e o algoritmo de Gear baseia-se nisso para executar a etapa de correção. O valor corrigido para a aceleração advém da equação (2.4):

$$\ddot{\mathbf{r}}_{i,\text{corr}}(t_n) = \frac{1}{m_i} \cdot \mathbf{F}_i(t_n), \quad \mathcal{P}_i \in \mathcal{S}.$$

Já para a aceleração angular, usa-se a equação de movimento rotacional simplificada para o caso de esferas (2.7), obtendo-se, para cada $\mathcal{P}_i \in \mathcal{S}$,

$$\dot{\boldsymbol{\omega}}_{i,\text{corr}}(t_n) = \frac{1}{J_i} \cdot \mathbf{M}_i(t_n).$$

Com isso, os erros são dados por

$$\begin{aligned}\Delta \ddot{\mathbf{r}}_i(t_n) &= \ddot{\mathbf{r}}_{i,\text{corr}}(t_n) - \ddot{\mathbf{r}}_{i,\text{pr}}(t_n) \\ \Delta \dot{\boldsymbol{\omega}}_i(t_n) &= \dot{\boldsymbol{\omega}}_{i,\text{corr}}(t_n) - \dot{\boldsymbol{\omega}}_{i,\text{pr}}(t_n)\end{aligned}, \quad \mathcal{P}_i \in \mathcal{S}.$$

E, finalmente, são corrigidos os vetores de derivadas conforme a equação (2.32). Para a translação, a equação diferencial é de ordem $p = 2$, e isso, juntamente com a ordem de extração k , determina as constantes de correção, de modo que

$$\mathbf{r}_{i,\text{corr}}^{[k]}(t_n) = \mathbf{r}_{i,\text{pr}}^{[k]}(t_n) + \mathbf{c}_{2,k} \cdot \frac{\Delta t_n^2}{2} \Delta \ddot{\mathbf{r}}_i(t_n), \quad \mathcal{P}_i \in \mathcal{S}.$$

A velocidade angular, por sua vez, está sujeita a uma equação diferencial de primeira ordem. O vetor de coeficientes $\mathbf{c}_{1,k-1}$, portanto, deve ser calculado para $p = 1$. Nessas condições, o vetor de derivadas da velocidade angular fica

$$\boldsymbol{\omega}_{i,\text{corr}}^{[k-1]}(t_n) = \boldsymbol{\omega}_{i,\text{pr}}^{[k-1]}(t_n) + \mathbf{c}_{1,k-1} \cdot \Delta t_n \cdot \Delta \dot{\boldsymbol{\omega}}_i(t_n), \quad \mathcal{P}_i \in \mathcal{S}.$$

Esse procedimento pode ser representado em forma de pseudocódigo como na listagem 3.1.

Listagem 3.1 – Pseudocódigo para a solução das equações de movimento de partículas esféricas por meio do algoritmo de Gear.

```

1 // Algoritmo de Gear
2 para cada partícula  $\mathcal{P}_i \in \mathcal{S}$ :
3   // Determinação da Vizinhança
4   determinar a vizinhança  $\mathcal{N}_i$ 
5   // Predição
6   calcular o vetor posição predito  $\mathbf{r}_{i,\text{pr}}^{[k]}(t_n) = \underline{\mathbf{A}}_k(\Delta t_n) \cdot \mathbf{r}_i^{[k]}(t_{n-1})$ 
7   calcular o vetor velocidade angular predito  $\omega_{i,\text{pr}}^{[k-1]}(t_n) = \underline{\mathbf{A}}_{k-1}(\Delta t_n) \cdot \omega_i^{[k-1]}(t_{n-1})$ 
8   atribuir  $\mathbf{F}_i \leftarrow \mathbf{0}$ 
9   atribuir  $\mathbf{M}_i \leftarrow \mathbf{0}$ 
10 fim
11
12 para cada partícula  $\mathcal{P}_i \in \mathcal{S}$ :
13   // Interação
14   para cada elemento  $\mathcal{E}_j \in \mathcal{N}_i$ :
15     calcular a força  $\mathbf{F}_{ji}$  que  $\mathcal{E}_j$  aplica sobre  $\mathcal{P}_i$ 
16     calcular o torque  $\mathbf{M}_{ji}$  que  $\mathcal{E}_j$  aplica sobre  $\mathcal{P}_i$ 
17     adicionar  $\mathbf{F}_i \leftarrow \mathbf{F}_i + \mathbf{F}_{ji}$ 
18     adicionar  $\mathbf{M}_i \leftarrow \mathbf{M}_i + \mathbf{M}_{ji}$ 
19   fim
20   para cada entidade  $\mathcal{E}$  externa ao domínio:
21     calcular a componente de força  $\mathbf{F}$  que  $\mathcal{E}$  aplica sobre  $\mathcal{P}_i$ 
22     calcular a componente de torque  $\mathbf{M}$  que  $\mathcal{E}$  aplica sobre  $\mathcal{P}_i$ 
23     adicionar  $\mathbf{F}_i \leftarrow \mathbf{F}_i + \mathbf{F}$ 
24     adicionar  $\mathbf{M}_i \leftarrow \mathbf{M}_i + \mathbf{M}$ 
25   fim
26 fim
27
28 para cada partícula  $\mathcal{P}_i \in \mathcal{S}$ :
29   // Correção
30   calcular o valor corrigido  $\ddot{\mathbf{r}}_{i,\text{corr}}(t_n) = (1/m_i) \cdot \mathbf{F}_i$ 
31   calcular o valor corrigido  $\ddot{\omega}_{i,\text{corr}}(t_n) = (1/J_i) \cdot \mathbf{M}_i$ 
32   calcular o erro  $\Delta\ddot{\mathbf{r}}_i(t_n) = \ddot{\mathbf{r}}_{i,\text{corr}}(t_n) - \ddot{\mathbf{r}}_{i,\text{pr}}(t_n)$ 
33   calcular o erro  $\Delta\ddot{\omega}_i(t_n) = \ddot{\omega}_{i,\text{corr}}(t_n) - \ddot{\omega}_{i,\text{pr}}(t_n)$ 
34   corrigir o vetor posição  $\mathbf{r}_{i,\text{corr}}^{[k]}(t_n) = \mathbf{r}_{i,\text{pr}}^{[k]}(t_n) + c_{2,k} \cdot \frac{\Delta t_n^2}{2} \Delta\ddot{\mathbf{r}}_i(t_n)$ 
35   corrigir o vetor velocidade angular  $\omega_{i,\text{corr}}^{[k-1]}(t_n) = \omega_{i,\text{pr}}^{[k-1]}(t_n) + c_{1,k-1} \cdot \Delta t_n \cdot \Delta\ddot{\omega}_i(t_n)$ 
36   atribuir  $\mathbf{r}_i^{[k]}(t_n) \leftarrow \mathbf{r}_{i,\text{corr}}^{[k]}(t_n)$  // Dessa forma, a correção é considerada exata
37   atribuir  $\omega_i^{[k-1]}(t_n) \leftarrow \omega_{i,\text{corr}}^{[k-1]}(t_n)$  //
38 fim
```

3.2.3 Exportação de Dados

A exportação de dados é a subetapa da simulação em que, de fato, são obtidos os resultados da simulação. A saída do programa é o conjunto de informações de interesse produzidas e exportadas pelo simulador. Essa exportação é, geralmente, direcionada para um ou mais arquivos de saída.

O pós-processamento dos resultados consiste em se processarem os dados exportados da simulação com a finalidade de se obterem as informações de interesse. Para tanto, utilizam-se gráficos, animações, tabelas e diversas outras técnicas de visualização de dados.

Existem dois fatores limitantes quanto à aquisição dos dados da simulação: o primeiro é a restrição da memória do computador. Geralmente, a quantidade de passos de tempo simulados e o número de elementos do sistema gera um volume de dados maior que a capacidade da memória. A segunda restrição é que o processo de escrita em disco é frequentemente uma operação lenta, reduzindo o desempenho do computador.²

Uma solução comumente adotada é a definição de dois parâmetros: o *número de passos de tempo para armazenamento* N_s e o *número de passos armazenados para a escrita* N_w . O número N_s conta quantos passos de tempo são simulados até que um seja armazenado. Com isso, a cada N_s simulados, $N_s - 1$ são descartados e apenas um é guardado na memória. O segundo parâmetro, N_w , define quantos passos de tempo são armazenados na memória antes de serem escritos em disco.

Essa solução pode ser implementada como na [listagem 3.2](#). Definem-se os contadores n_s e n_w , inicializados com o valor zero. A cada passo de tempo, o contador n_s é incrementado. Quando esse contador atinge o valor máximo de N_s , ele é reiniciado, o contador n_w é incrementado e é feito um registro do instante de tempo na memória. Quando o contador n_w se iguala a N_w , o mesmo é reiniciado, e as informações armazenadas são escritas em disco.

3.2.4 Condição de Parada

A condição de parada representa a condição que interrompe o laço de simulação e leva à finalização do programa. Essa condição se apresenta de diversas formas dependendo da aplicação, podendo levar em conta o estado do sistema e parâmetros da simulação ([SCHWAGER; PÖSCHEL, 2005](#)). O critério de parada pode ser equivalentemente definido por um critério de continuação que, enquanto for verdadeiro, mantém a execução do laço.

Uma das condições mais simples e mais aplicadas é a condição de parada por tempo

² Por simplicidade, é empregado aqui o termo *memória* para referir-se unicamente à *memória principal* do computador, comumente representada pela memória RAM e pelos registradores. Por outro lado, o termo *disco* é usado para referir-se à *memória secundária*. Esse tipo de memória é geralmente representada pelas memórias não voláteis, como discos rígidos (HDs), unidades de estado sólido (SSDs) e discos de leitura ótica (CDs e DVDs).

Listagem 3.2 – Pseudocódigo para a exportação de dados.

```

1  inicializar  $n_s \leftarrow 0$ 
2  inicializar  $n_w \leftarrow 0$ 
3  // Laço de simulação
4  enquanto o critério de parada não for atingido:
5      incrementar  $n_s \leftarrow n_s + 1$ 
6      se  $n_s = N_s$ :
7          atribuir  $n_s \leftarrow 0$ 
8          incrementar  $n_w \leftarrow n_w + 1$ 
9          guardar estado atual na memória
10         se  $n_w = N_w$ :
11             atribuir  $n_w \leftarrow 0$ 
12             escrever estados guardados nos arquivos de saída
13         fim
14     fim
15     ... // Solução das equações do sistema
16 fim

```

Listagem 3.3 – Pseudocódigo para a condição de parada do laço da simulação.

```

1  ler  $t^i$  do arquivo de inicialização
2  inicializar  $n \leftarrow 0$ 
3  inicializar  $t_0 \leftarrow t^i$ 
4  // Laço de simulação
5  enquanto  $t_n \leq t^f$ :
6      atribuir  $n \leftarrow n + 1$ 
7      ... // Exportação de dados
8      ... // Solução das equações
9      ... // Cálculo de  $\Delta t_n$ 
10     atribuir  $t_n \leftarrow t_{n-1} + \Delta t_n$ 
11 fim

```

final da simulação. Esse critério baseia-se na definição de um valor t^f para o instante final da simulação, e a simulação progride até que esse limite seja ultrapassado. O laço da simulação pode então ser representado como na [listagem 3.3](#). O instante inicial t^i é proveniente da etapa de inicialização, e o n -ésimo passo de tempo Δt_n é dependente do método escolhido, do arquivo de inicialização e dos algoritmos de integração e de determinação de vizinhanças.

3.2.5 Finalização

Por fim, quando o critério de parada é atingido (ou o critério de continuação falha), o programa inicia a etapa de finalização. Nessa etapa, os últimos dados são escritos nos arquivos de saída, os elementos da simulação são destruídos e o programa termina. Tem início, então, a fase de pós-processamento.

É comum também que a etapa de finalização determine arquivos de inicialização para serem usados por outras simulações. Os dados escritos nesses arquivos são suficientes para se gerar um estado inicial idêntico ao estado final da simulação finalizada. Dessa

forma, é possível fazer uma simulação dar prosseguimento a outra.

3.3 Monitoramento da Vizinhança

Segundo [Schwager e Pöschel \(2005\)](#), p. 26), o maior custo computacional em uma simulação de elementos discretos está associado à avaliação das forças atuantes sobre as partículas. Esse custo advém da quantidade de interações que devem ser consideradas dado o geralmente grande número de pares de colisão possíveis. Em um sistema de $N_{\mathcal{E}}$ elementos, dos quais $N_{\mathcal{P}}$ são partículas, e com a simplificação de que cada par interage independentemente dos demais, cada partícula pode interagir com até $N_{\mathcal{E}} - 1$ outros elementos. Isso resulta em um total de

$$N_{\mathcal{P}} \cdot (N_{\mathcal{E}} - 1)$$

interações computadas a cada passo de tempo. Adicionando a hipótese simplificativa de que não há elementos além das partículas, o número total de interações avaliadas por passo de tempo fica da ordem de $N_{\mathcal{P}}^2$.

Essa quantidade de interações limita o número de partículas e de passos de tempo aplicáveis, e motiva a busca por métodos mais eficientes para o monitoramento da vizinhança das partículas.

A vizinhança de uma partícula \mathcal{P}_i é o conjunto \mathcal{N}_i de todos os elementos da simulação com os quais a partícula pode interagir.

Uma simples consideração, a Terceira Lei de Newton, pode reduzir pela metade a quantidade de interações avaliadas entre partículas. Ainda assim, o uso de algoritmos de monitoramento de vizinhanças mais elaborados pode ser imprescindível, dependendo da simulação executada.

Segundo [Bićanić \(2007\)](#), p. 6), esses algoritmos podem ser divididos entre os *baseados em corpos* e os *baseados no espaço*. No primeiro caso, são consideradas as formas geométricas das partículas, e dois elementos só podem colidir se estiverem suficientemente próximos uns dos outros. Como exemplo desse tipo de método, tem-se o algoritmo de Verlet e os volumes limitantes. Métodos baseados no espaço, por outro lado, fundamentam-se na divisão do domínio da simulação em regiões, como o algoritmo *link cell* e o algoritmo *lattice*. Dois elementos, então, só podem colidir se estiverem na mesma região ou em regiões vizinhas. Esses algoritmos podem ser encontrados em [Schwager e Pöschel \(2005\)](#).

Dadas duas partículas \mathcal{P}_i e \mathcal{P}_j , a terceira lei de Newton estabelece que à força \mathbf{F}_{ij} que a partícula i aplica sobre a j corresponde uma reação \mathbf{F}_{ji} que satisfaz

$$\mathbf{F}_{ji} = -\mathbf{F}_{ij}.$$

Listagem 3.4 – Pseudocódigo para a aplicação da Terceira Lei de Newton.

```

1 ... // Predição e zeragem dos vetores força e torque resultantes
2 para cada partícula  $\mathcal{P}_i \in \mathcal{P}$ :
3   // Interação
4     para cada elemento  $\mathcal{E}_j \in \mathcal{N}_i$ :
5       calcular a força  $\mathbf{F}_{ji}$  que  $\mathcal{E}_j$  aplica sobre  $\mathcal{P}_i$ 
6       calcular o torque  $\mathbf{M}_{ji}$  que  $\mathcal{E}_j$  aplica sobre  $\mathcal{P}_i$ 
7       adicionar  $\mathbf{F}_i \leftarrow \mathbf{F}_i + \mathbf{F}_{ji}$ 
8       adicionar  $\mathbf{M}_i \leftarrow \mathbf{M}_i + \mathbf{M}_{ji}$ 
9       adicionar  $\mathbf{F}_j \leftarrow \mathbf{F}_j - \mathbf{F}_{ji}$ 
10      adicionar  $\mathbf{M}_j \leftarrow \mathbf{M}_j - \mathbf{M}_{ji}$ 
11    fim
12  ... // Interações com entidades externas
13  ... // Etapa de correção
14 fim

```

Esse princípio possui duas consequências. A primeira é que, se \mathcal{P}_j está na vizinhança de \mathcal{P}_i , então \mathcal{P}_i está, necessariamente, na vizinhança de \mathcal{P}_j . A segunda, que a força e o torque de interação entre i e j precisam ser computados apenas uma vez.

Uma maneira de se aplicar essa consideração é definir-se que \mathcal{P}_j somente pode pertencer a \mathcal{N}_i se $i < j$. Se, de fato, ocorrer que $i < j$, então \mathcal{P}_j pertence a \mathcal{N}_i , mas \mathcal{P}_i não pertence a \mathcal{N}_j , e por isso a parcela de força que i aplica sobre j é perdida no laço de integração de j . Sendo assim, a força $\mathbf{F}_{ij} = -\mathbf{F}_{ji}$ é adicionada a \mathbf{F}_j quando esse valor for calculado no laço de integração da partícula i . Isso corresponde a uma alteração na listagem 3.1 nas linhas 13 a 19 como indicado na listagem 3.4.

3.4 Limitação do Passo de Tempo

Diversos autores sugerem a limitação do passo de tempo da simulação para se garantirem a estabilidade numérica e a representatividade das soluções. Segundo Gomes (2014), uma possível estratégia é definir-se um valor de *passo de tempo crítico* Δt_c com a forma

$$\Delta t_c = C \cdot \sqrt{m/k}, \quad C \in \mathbb{R}, \quad (3.1)$$

sendo m a massa efetiva e k a constante elástica efetiva dos pares de colisão. O valor da constante C é, geralmente, determinado com base na experiência do usuário.

A limitação do passo de tempo, porém, depende fortemente da configuração da simulação. Por exemplo, quanto maiores forem as velocidades das partículas, menores os passos de tempo que devem ser usados a fim de evitar que as partículas se atravessem sem que a sua colisão seja computada. Entretanto a expressão em (3.1) não indica nenhuma dependência com relação à velocidade.

Na ausência de melhores formulações, passos de tempo arbitrados acabam sendo utilizados nas simulações, e sua obtenção baseia-se fortemente na intuição do usuário, em

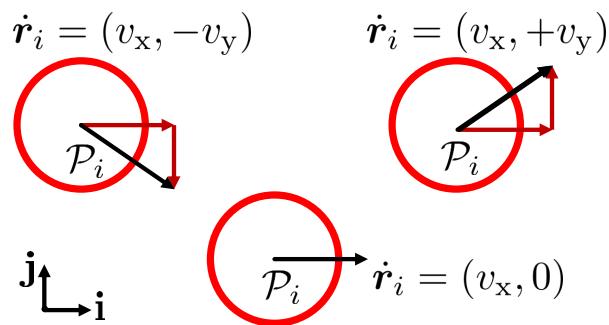
cálculos prévios e na experimentação numérica.

3.5 Condições de Contorno e Restrições

Comumente, além das interações entre partículas, os métodos de elementos discretos contam com condições adicionais para a execução das simulações. Essas considerações são denominadas de *condições de contorno*. Dentre elas, as mais aplicadas são:

- Movimento prescrito*: circunstância em que o movimento de um elemento é totalmente determinado por uma função fornecida na etapa de inicialização. É essa a condição que rege os elementos de contorno.
- Condição de repetição*: nesse caso, repete-se periodicamente a região da simulação em uma ou mais dimensões (SCHWAGER; PÖSCHEL, 2005, p. 15). Partículas que saem da região da simulação por um lado são reinseridas no lado oposto. Dessa forma, o domínio da simulação torna-se efetivamente infinito. A vizinhança de uma partícula deve, então, levar em consideração não apenas os elementos fisicamente próximos a ela, mas também aqueles localizados no extremo oposto da região de simulação, com os quais a partícula pode vir a colidir se houver o cruzamento das fronteiras. Mais detalhes sobre essa condição são apresentados por Allen e Tildesley (1987, p. 24), Bićanić (2007) e Schwager e Pöschel (2005, p. 15).
- Condição de reflexão*: essa condição resume-se à definição de paredes reflexivas. Quando uma partícula colide com um elemento desse tipo, sua velocidade é refletida com relação à parede, enquanto a componente de velocidade paralela à parede mantém-se inalterada, como representado na figura 3.2. Mais considerações acerca dessa condição podem ser encontradas em Schwager e Pöschel (2005, p. 169).

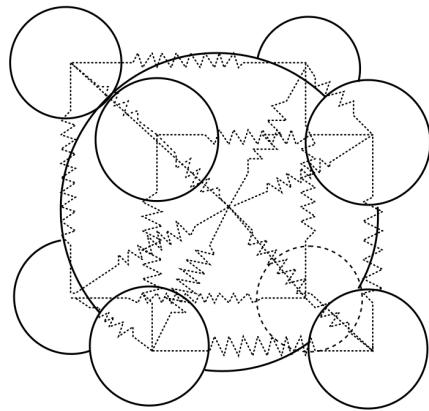
Figura 3.2 – Condição de parede reflexiva.



Fonte: do Autor.

- d) *Restrição de movimento:* a restrição de movimento consiste em se acoparem as equações de movimento de duas ou mais partículas. Essa condição encontra-se presente em partículas compostas para mediar a movimentação relativa entre os subelementos. Uma maneira simples de se implementar essa restrição é com a adição de uma força elástica representada pela ação de molas, como indicado na figura 3.3. Também é possível adicionarem-se restrições nos graus de liberdade, como demonstrado em Machado et al. (2012, pp. 113,114), ao se utilizarem juntas cinemáticas.

Figura 3.3 – Restrição ao movimento de subelementos de uma partícula composta pela ação de molas.



Fonte: Schwager e Pöschel (2005, p. 84).

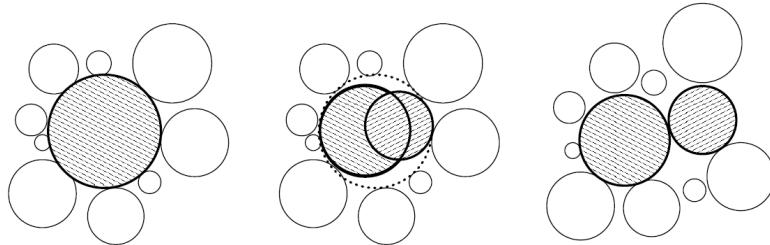
3.6 Outros Aspectos do Método

Além das principais características expostas neste capítulo, existem outros aspectos mais específicos que podem ser considerados em um método de elementos discretos. Em especial, tem-se o problema de fragmentação de partículas e a paralelização dos simuladores.

A fragmentação de partículas pode ser abordada de diversas formas. Segundo Schwager e Pöschel (2005), uma maneira simples mas eficiente de se implementar esse fenômeno é considerar uma função probabilidade de fratura, dependente da geometria, das propriedades físicas e do carregamento sobre as partículas. Após computadas as forças sobre cada partícula, calcula-se a probabilidade de sua fragmentação e, a partir dessa distribuição de probabilidades, determina-se se ocorre a sua quebra.

O processo de fragmentação de uma partícula, representado na figura 3.4, consiste em substituir a partícula fragmentada por dois novos elementos. Essas duas partículas são construídas com uma elevada superposição entre si, o que resultaria numa força extrema caso se considerassem os modelos de interação usuais. Para evitar esse problema, aplicam-se modelos de força específicos, de menor intensidade, para o caso especial da fragmentação.

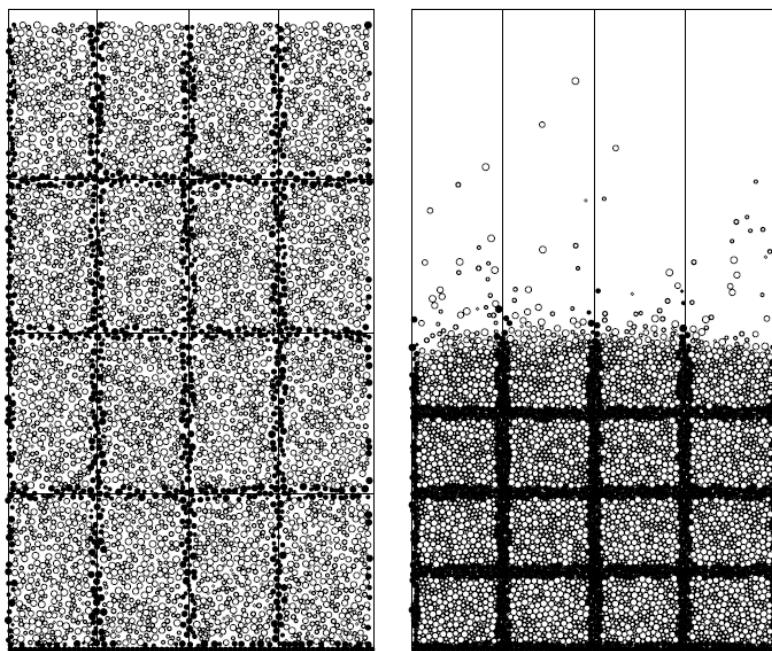
Figura 3.4 – Processo de fragmentação de uma partícula. O elemento fragmentado é substituído por duas novas partículas com uma elevada superposição.



Fonte: Schwager e Pöschel (2005, p.112).

Outro aspecto de grande relevância para o DEM é a paralelização de seus algoritmos. A computação em paralelo é aplicada em processos que podem ser divididos em tarefas menores, ou subprocessos, independentes entre si, e que podem ser executados paralelamente. Cada subprocesso é associado a uma unidade de processamento do computador para ser executado. Essa técnica permite uma drástica redução do tempo de computação em razão da execução de várias tarefas simultaneamente. Em contrapartida, caso exista dependência entre os subprocessos, é necessária a comunicação entre os processadores, o que pode reduzir bastante o desempenho do método. Sendo assim, a paralelização só é usualmente aplicada em sistemas com elevado número de partículas (SCHWAGER; PÖSCHEL, 2005, p. 128).

Figura 3.5 – Divisão do domínio da simulação em 16 regiões. Os segmentos de reta indicam a fronteira dos domínios locais dos processadores. Partículas localizadas próximo às fronteiras são representadas em cor preta.



Fonte: Schwager e Pöschel (2005, p.128).

Em métodos de elementos discretos, cada processador recebe, na etapa de inicialização, a incumbência de simular uma determinada região do domínio da simulação, como representado na figura 3.5.

Cada processador executa a etapa de predição das partículas pertencentes ao seu domínio local. Depois, ocorre a comunicação entre processadores responsáveis por regiões vizinhas com a finalidade de permitir a interação entre partículas próximas às fronteiras dos domínios locais. Após a troca de informações, cada processador executa, separadamente, as etapas de interação, correção, exportação de dados e, novamente, caso o critério de parada não tenha sido atingido, a etapa de predição.

3.7 Acoplamento com Outros Métodos

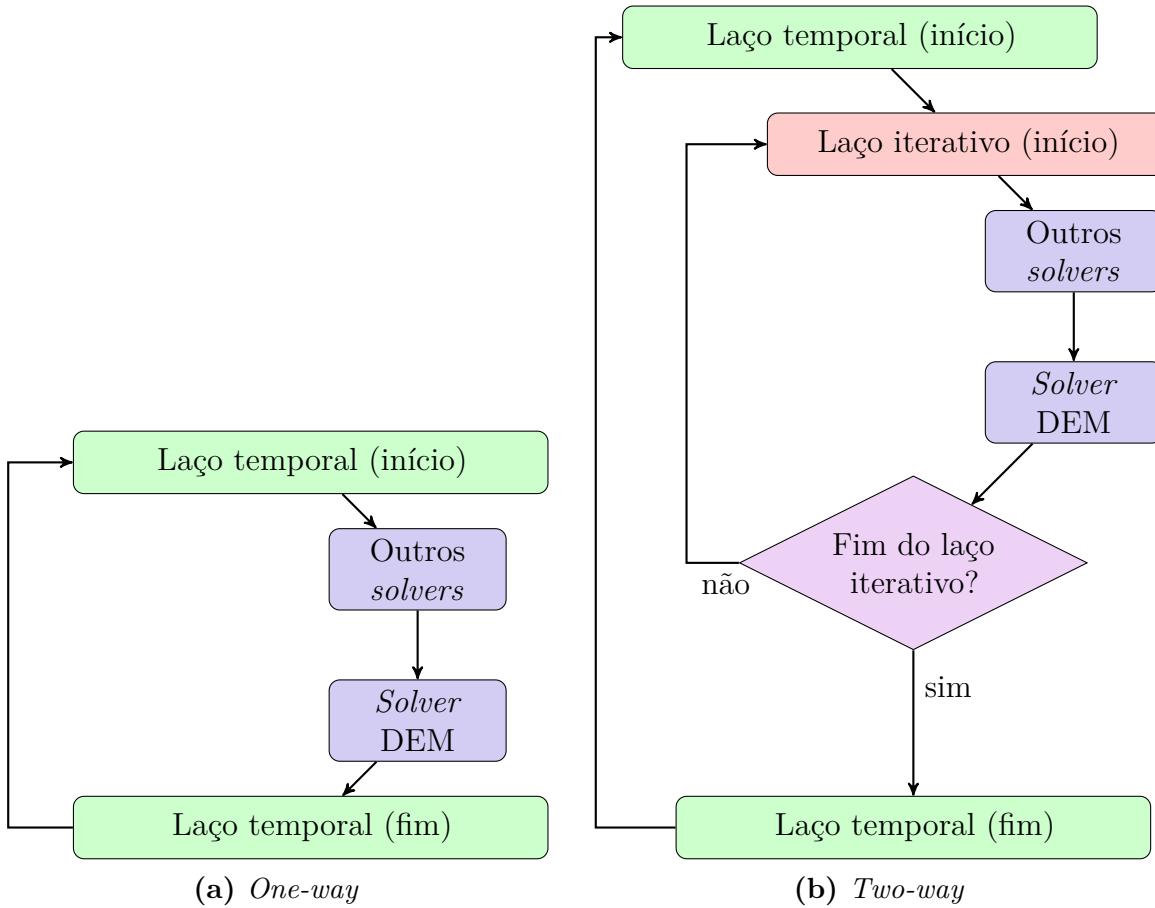
Um dos ramos de pesquisa mais importantes para o DEM é o seu acoplamento com outros métodos, como o Método de Elementos Finitos (FEM) e a mecânica de fluidos computacional (CFD). Isso permite aplicar os pontos fortes de cada um dos métodos em simulações de sistemas complexos. A extensão do Método de Elementos Discretos, quando se consideram outras características físicas como equações de estado termodinâmico, interação eletromagnética, trocas de calor e distribuição de tensões, é conhecida como Método de Elementos Discretos Estendido (XDEM).

Cada componente do XDEM contribui com um *solver*, isto é, com uma rotina de solução para resolver a parte do problema que lhe compete. As principais abordagens para o acoplamento entre esses *solvers* são a *one-way* (unidirecional) e a *two-way* (bidirecional). Na primeira, um dos *solvers* é executado independentemente, e seu resultado é fornecido como parâmetro para os demais. Na abordagem *two-way*, os *solvers* são executados alternadamente dentro um laço adicional executado em cada passo de tempo. Essas abordagens são ilustradas na figura 3.6.

Como exemplos do acoplamento entre o DEM e o FEM, podem-se citar Oñate e Rojek (2004), Nakashima e Oida (2004), Dang e Meguid (2011) e Gomes (2014). A interação entre as partículas e os elementos finitos pode ser feita de maneira semelhante à interação entre partículas e elementos planos, como descrito no capítulo 2, ou entre partículas e elementos poliédricos, como explicado por Schwager e Pöschel (2005, pp. 86–104). Outra técnica, apresentada por Gomes (2014), consiste em se considerarem partículas de contorno na fronteira dos corpos contínuos. Essas abordagens estão ilustradas na figura 3.7.

O acoplamento entre DEM e CFD, por sua vez, têm como exemplo os trabalhos de Tsuji, Kawaguchi e Tanaka (1993), Tsuji, Yabumoto e Tanaka (2008), Maio, Renzo e Trevisan (2009), Fries et al. (2011) e Deen et al. (2007). A influência das partículas sobre o escoamento do fluido é considerada como um termo fonte nas equações de Navier-Stokes.

Figura 3.6 – Abordagens *one-way* e *two-way* para o acoplamento no Método de Elementos Discretos Estendido.

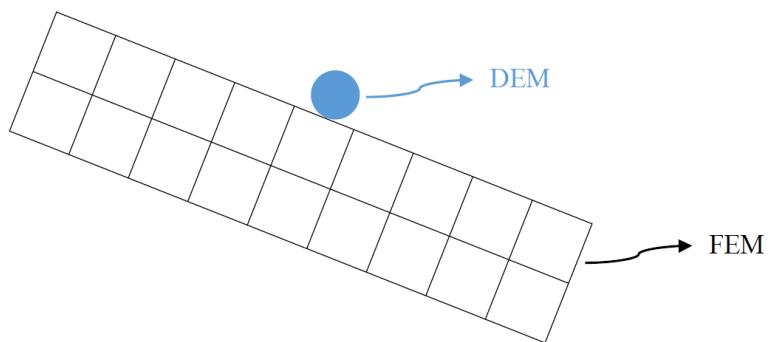


Fonte: do Autor.

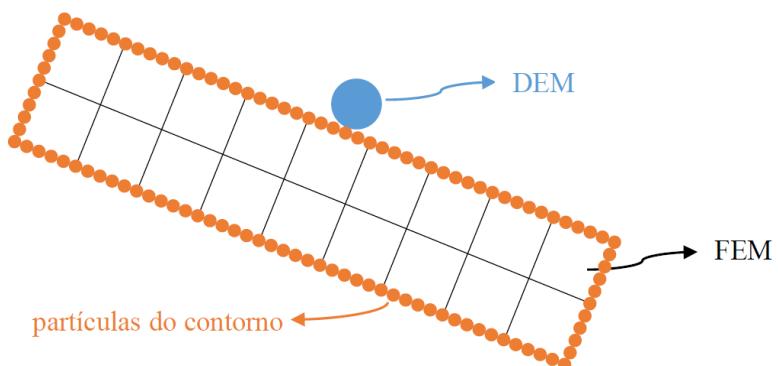
Por outro lado, o fluido afeta o movimento das partículas através do arrasto. Tanto o termo fonte quanto o arrasto são função do número de Reynolds do escoamento, da velocidade relativa entre as partículas e o fluido, da quantidade de partículas e das propriedades do fluido.

Além disso, pode ser considerada a troca de calor entre as partículas. A potência de calor trocado também é dada por modelos de interação. Como exemplo, pode-se citar o trabalho de Simsek et al. (2009), em que aplica-se um coeficiente de troca de calor dependente da condutividade térmica, do módulo de elasticidade e da geometria das partículas interagentes, bem como da força normal de colisão.

Figura 3.7 – Abordagens para o contato entre partículas, representantes do DEM, e corpos contínuos, em que se utiliza o FEM.



(a) Interação entre partícula e elemento plano.



(b) Uso de partículas de contorno.

Fonte: Gomes (2014).

4 Implementação Computacional e Resultados

Os conceitos apresentados nos capítulos anteriores servem como fundamento para a implementação de algoritmos de solução através do Método de Elementos Discretos. Este capítulo é destinado a apresentar a implementação computacional desenvolvida neste trabalho e a executar as tarefas de apresentação de resultados de simulação e de validação numérica da implementação.

4.1 Implementação Computacional

A implementação computacional de um algoritmo consiste em se transformarem modelos matemáticos e numéricos em programas executáveis por computadores, e, portanto, passíveis de análises concretas. Esta seção é dedicada à exposição dos principais pontos concernentes à implementação e, com isso, possibilita o entendimento das capacidades e das limitações do simulador desenvolvido.

O projeto está disponível on-line em [Comelli \(2017\)](#). Como suporte ao desenvolvimento do código, foram usadas as bibliotecas Boost ([KARLSSON, 2005](#)) e JSON for Modern C++ ([LOHMAN, 2017](#)).

Em primeiro lugar, foi implementada, em linguagem C++, uma biblioteca de simulação contendo o suporte à construção de simuladores. Essa biblioteca se fundamenta na técnica de *metaprogramação template* da linguagem para gerar *funções* e *classes* generalizadas, podendo especializar-se dependendo das necessidades do simulador. A biblioteca foi projetada de tal forma que a inserção de novos tipos de elementos e de interações seja simples e direta. O principal componente da biblioteca é a *classe template Simulator*, que gerencia a simulação e representa os simuladores. Essa classe recebe como parâmetros os tipos de partículas, de elementos de contorno, campos de força e elementos externos e as interações da simulação. Com isso, a classe especializada é capaz de ler arquivos de entrada e simular o comportamento do sistema.

Dentre as capacidades da classe template *Simulator*, destacam-se:

- a) a possibilidade de se usarem interações de qualquer natureza. Essa característica é fundamental por permitir que não apenas forças de colisão entre partículas sejam consideradas, mas também interações como a força de arrasto aplicada no problema da queda com arrasto, descrito na [seção 4.6](#);
- b) a capacidade de se simularem entidades de geometria, tipo e propriedades

arbitrários;

- c) embora os arquivos de entrada e saída estejam restritos ao formato JSON, o método utilizado para transformar esses dados em elementos de simulação e, no sentido inverso, exportar elementos para arquivos de saída, é customizável;
- d) a estrutura de arquivos e diretórios é definida pelos próprios arquivos de entrada sendo, portanto, independentes da implementação do simulador.

Entretanto existem algumas limitações que não foram ultrapassadas neste trabalho por não fazerem parte de seu escopo. Dentre as principais restrições, tem-se que:

- a) o único método de integração possível, na implementação corrente, é o algoritmo de Gear;
- b) o critério de parada é, necessariamente, o critério de tempo limite de simulação;
- c) o método de monitoramento de vizinhanças é restrito àquele apresentado na [seção 3.3](#), em que cada combinação de elementos possível é avaliada levando em conta apenas a simplificação resultante da Terceira Lei de Newton;
- d) os arquivos de entrada e saída estão limitados ao formato JSON.

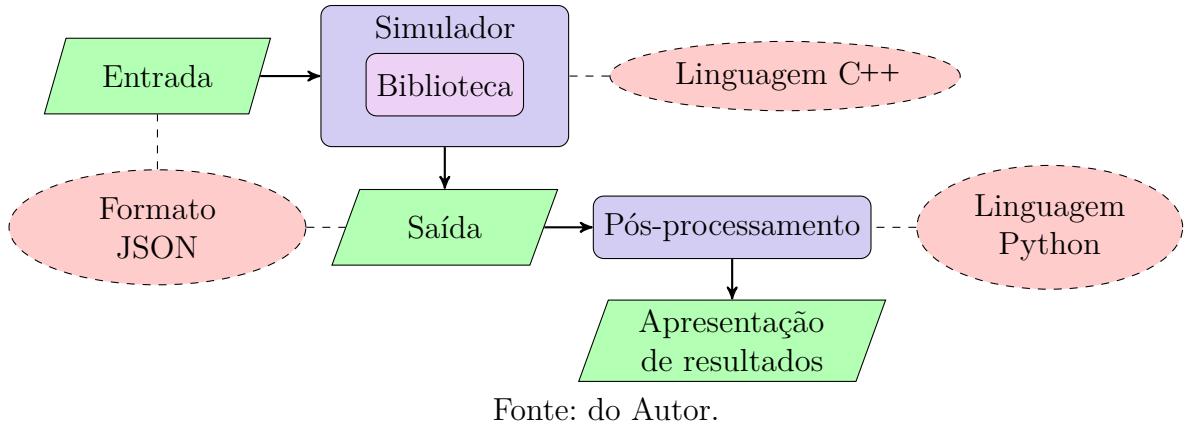
Além da [Simulator](#), outras classes, classes template, funções e funções template foram predefinidas visando facilitar a criação de simuladores.

Como exemplo, foram definidas as propriedades físicas necessárias aos modelos de interação, como o módulo de elasticidade, as constantes de amortecimento normal e tangencial, o coeficiente de atrito e a massa específica. A criação de novas propriedades físicas é auxiliada pelas classes template [Property](#) e [PositiveProperty](#) por meio da herança de classes.

Os tipos de elementos presentes na biblioteca são partículas esféricas, paredes planas, o campo gravitacional e fluido ambiente. Às partículas esféricas, às paredes planas e ao fluido ambiente podem ser associadas propriedades quaisquer dependendo dos modelos de interação estudados. Novos tipos de entidade podem ser definidos usando como base as classes template predefinidas [Particle](#) e [Boundary](#).

Posteriormente, as classes e funções da biblioteca foram especializadas para obter-se um simulador. Tal simulador recebe dados de entrada através de arquivos no formato JSON. Um exemplo de arquivo nesse formato é o apresentado no [apêndice B](#), arquivo de entrada do problema do lançamento oblíquo, descrito na [seção 4.3](#). Após essa etapa, o simulador promove a simulação correspondente com base na entrada e exporta os resultados, também em formato JSON. A partir dos resultados exportados, scripts de pós-processamento, escritos em linguagem Python, são executados para a geração de gráficos, figuras, tabelas e animações. Esses componentes e a sua inter-relação estão ilustrados na [figura 4.1](#).

Figura 4.1 – Componentes do programa implementado.



Fonte: do Autor.

Dentre as interações consideradas estão os modelos de força de colisão apresentados na [seção 2.3](#), que podem ocorrer entre partículas esféricas e, no caso do modelo de amortecimento linear, entre partículas esféricas e paredes planas. Mais ainda, o simulador considera a interação entre partículas e o campo gravitacional e a resistência imposta pelos fluidos à movimentação dos corpos imersos nele.

O algoritmo de integração selecionado foi o algoritmo de Gear como apresentado na [listagem 3.1](#). A exportação de dados e a condição de parada foram escolhidas como nas listagens [3.2](#) e [3.3](#). Ainda, para as aplicações deste trabalho, não são consideradas grandes quantidades de partículas, de modo que o algoritmo para monitoramento de vizinhanças é simplesmente aquele indicado na [listagem 3.4](#).

4.2 Resultados e Validação

Com o simulador finalizado, é necessária a avaliação de seus resultados para que a implementação seja validada numericamente. Para tanto, simulações são executadas e seus resultados são comparados com soluções analíticas ou qualitativamente avaliados.

O procedimento adotado para validação numérica da implementação considera:

- a análise qualitativa da solução numérica. Por exemplo, em simulações sem forças externas, as quantidades de movimento linear e angular devem se manter constantes no sistema, e a verificação desse fato no resultado da simulação indica, nesse sentido, uma representatividade correta do fenômeno físico estudado;
- a comparação qualitativa com a solução analítica, se esta estiver disponível. Este item é semelhante ao anterior, mas considera ainda a solução analítica para o estudo do comportamento dos resultados numéricos;
- o estudo do erro da solução numérica com relação à analítica, se esta for conhecida. O erro representa o grau de discordância entre as soluções, e valores

baixos indicam a corretude do algoritmo.

O *erro* de um resultado é uma medida fundamental para se atestar a validade do método. Quanto maior for o erro, menos representativa se torna a simulação. Sendo y uma solução obtida numericamente e y_S a solução exata do problema, define-se o erro de y como

$$\text{Erro}(y) = |y - y_S|.$$

O erro máximo, por sua vez, é o maior erro pontual entre y e y_S , isto é, é o maior elemento do conjunto imagem da função erro:

$$\text{Erro}_{\max}(y) = \max(\text{Im}(\text{Erro}(y))).$$

Para funções vetoriais \mathbf{y} , o erro e o erro máximo são definidos equivalentemente como

$$\text{Erro}(\mathbf{y}) = \|\mathbf{y} - \mathbf{y}_S\|, \quad (4.1)$$

$$\text{Erro}_{\max}(\mathbf{y}) = \max(\text{Im}(\text{Erro}(\mathbf{y}))). \quad (4.2)$$

Todas as simulações foram executadas em um computador com épsilon de máquina aproximadamente igual a 2×10^{-16} , memória RAM de 16 GB e processador Intel Core i7-4770 3,40 GHz.

Foram escolhidos quatro problemas a serem apresentados e estudados: o problema do lançamento oblíquo, em que uma partícula é lançada no espaço sujeita unicamente à ação da gravidade; o problema da esfera quicando, em que uma partícula liberada no espaço em repouso é atraída, pela gravidade, em direção ao chão, colidindo com ele; a colisão entre esferas, em que duas partículas esféricas se chocam, podendo inclusive ter velocidade angular não nula; e o problema da queda com arrasto, em que uma partícula é solta no espaço e cai em decorrência da ação da gravidade, sofrendo, ao longo de sua trajetória, a resistência do ar.

4.3 Lançamento Oblíquo

O lançamento oblíquo é um dos problemas clássicos de física básica. Considera-se uma partícula de massa m sujeita unicamente à ação de um campo gravitacional constante cuja aceleração da gravidade é \mathbf{g} . Nessas condições, as equações de movimento para a partícula se tornam, simplesmente,

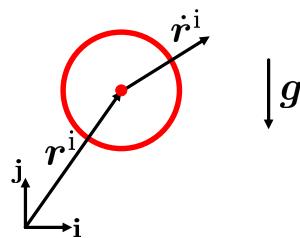
$$\ddot{\mathbf{r}} = \mathbf{g}, \quad (4.3)$$

$$\dot{\boldsymbol{\omega}} = \mathbf{0}. \quad (4.4)$$

O instante inicial da simulação é definido como $t^i = 0$, e o instante final é um valor arbitrário t^f . A posição, a velocidade e a velocidade angular da partícula no instante inicial assumem, respectivamente, os valores \mathbf{r}^i , $\dot{\mathbf{r}}^i$ e $\boldsymbol{\omega}^i$. Com isso, as equações (4.3) e (4.4) podem ser resolvidas por integração simples.

Por simplicidade, considera-se que o movimento ocorra apenas no plano XY , que a gravidade atue no sentido negativo do eixo Y , isto é, $\mathbf{g} = -g \cdot \mathbf{j}$ com $g > 0$, e que a partícula não possua rotação, como representado na figura 4.2. Nessa situação, a posição em Y da partícula é chamada de altura.

Figura 4.2 – Ilustração do problema do lançamento oblíquo.



Fonte: do Autor.

Com isso, escrevendo $\mathbf{r} = (r_x, r_y, r_z)$, $\dot{\mathbf{r}} = (\dot{r}_x, \dot{r}_y, \dot{r}_z)$ e $\ddot{\mathbf{r}} = (\ddot{r}_x, \ddot{r}_y, \ddot{r}_z)$, é obtida a solução analítica apresentada na tabela 4.1.

Tabela 4.1 – Solução do problema do lançamento oblíquo.

	Direção X	Direção Y
Posição	$r_x(t) = r_x^i + \dot{r}_x^i \cdot t$	$r_y(t) = r_y^i + \dot{r}_y^i \cdot t - \frac{g \cdot t^2}{2}$
Velocidade	$\dot{r}_x(t) = \dot{r}_x^i$	$\dot{r}_y(t) = \dot{r}_y^i - g \cdot t$
Aceleração	$\ddot{r}_x(t) = 0$	$\ddot{r}_y(t) = -g$

Fonte: do Autor.

Embora seja conceitualmente simples e de fácil resolução, o problema do lançamento oblíquo permite:

- a validação numérica da implementação do algoritmo de Gear em simulações com a atuação exclusiva de forças constantes. Em cada instante, as coordenadas da partícula são previstas e, em seguida, corrigidas a partir do valor da força da gravidade. A proximidade da resposta numericamente calculada com a solução analítica evidencia o bom funcionamento do algoritmo;
- a obtenção da evolução do erro com o tempo de simulação. Não só existe um erro associado ao método como esse erro se acumula com o tempo.

A fim de se compararem os resultados numéricos com a solução analítica, foram arbitrados aos parâmetros do problema os valores dispostos na [tabela 4.2](#). O arquivo em formato JSON utilizado para essa simulação é apresentado no [apêndice B](#).

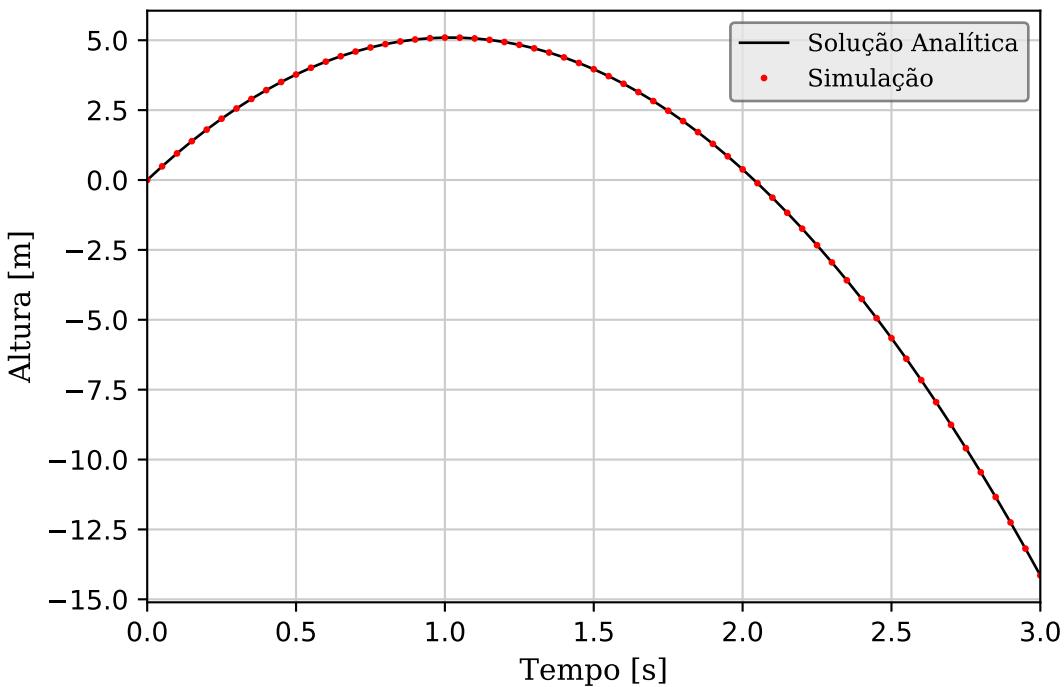
Tabela 4.2 – Parâmetros para o problema de lançamento oblíquo.

Gravidade	$g = 9,81 \text{ m/s}^2$
Massa da partícula	$m = 1 \text{ kg}$
Raio da partícula	$R = 3 \text{ cm}$
Posição inicial	$(r_x^i, r_y^i, r_z^i) = (0; 0; 0) \text{ m}$
Velocidade inicial	$(\dot{r}_x^i, \dot{r}_y^i, \dot{r}_z^i) = (20; 10; 0) \text{ m/s}$
Aceleração inicial	$(\ddot{r}_x^i, \ddot{r}_y^i, \ddot{r}_z^i) = (0; -9,81; 0) \text{ m/s}^2$
Instante final	$t^f = 3 \text{ s}$
Passo de tempo	$\Delta t = 10^{-3} \text{ s}$
Ordem de extração	$k = 7$

Fonte: do Autor.

Conquanto se trate de um problema bidimensional sem rotação, o simulador emprega a formulação tridimensional e considera a evolução da velocidade angular da partícula.

Figura 4.3 – Solução para a altura da partícula no problema de lançamento oblíquo¹.

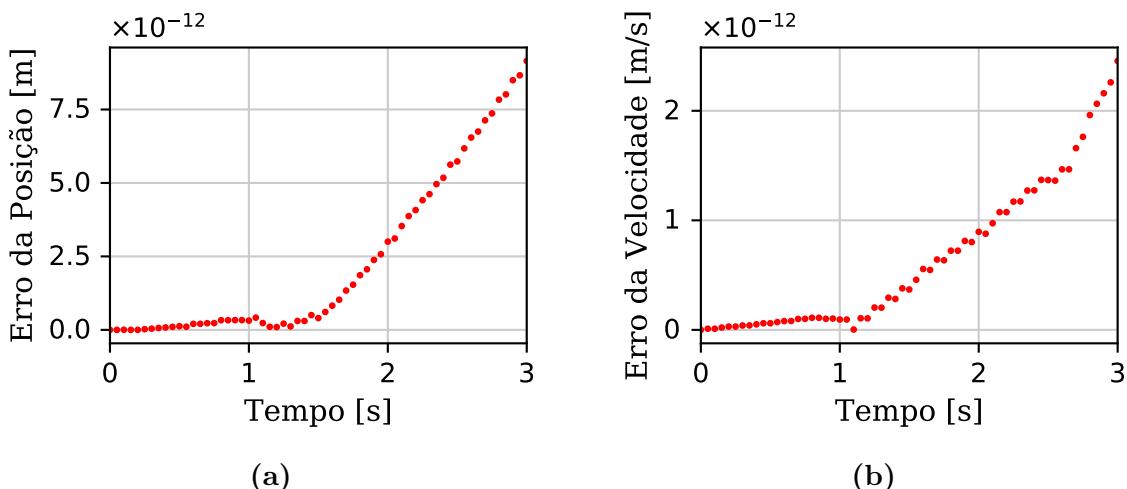


Fonte: do Autor.

¹ Para melhor visualização dos resultados, é apresentado apenas um ponto a cada dez passos de tempo.

Com esses parâmetros, foi calculada a trajetória da partícula dada pela solução analítica e aquela resultante da simulação. Na figura 4.3 estão expostas as soluções para a altura da partícula. Observa-se nessa figura que os resultados obtidos pelo simulador se aproximam bastante da solução analítica das equações. As demais variáveis do problema, como a posição horizontal, as velocidades horizontal e vertical e a aceleração vertical, também seguem o comportamento esperado, mantendo-se bastante próximos da solução analítica.

Figura 4.4 – Erro da simulação com relação à solução numérica para (a) a posição e (b) a velocidade da partícula do problema do lançamento oblíquo.



Fonte: do Autor.

O erro do vetor posição e o erro do vetor velocidade, como funções do tempo, são apresentados na figura 4.4. Esses erros são calculados de acordo com a equação (4.1). Observa-se nessa figura que o erro da posição se mantém inferior a 0,01 nm durante toda a simulação, evidenciando a exatidão do método. Além disso, é possível notar que, no início da simulação, o erro se mantém próximo do epsilon de máquina, acumulando-se com o avanço da simulação. Tal comportamento mantém-se mesmo com o uso de passos de tempo distintos.

4.4 Esfera Quicando

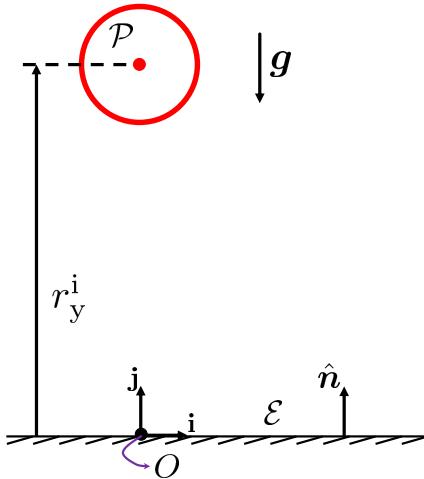
Como extensão do problema de queda livre, o problema da esfera quicando inclui, além da gravidade, a colisão entre uma esfera e uma parede plana. Supõe-se uma partícula \mathcal{P} esférica de raio R e massa m que, no instante inicial $t^i = 0$, é liberada no espaço, em repouso, na posição $\mathbf{r}^i = (0, r_y^i, 0)$.

A partícula está sujeita a um campo gravitacional constante cuja aceleração da gravidade é $\mathbf{g} = -g \cdot \mathbf{j}$ com $g > 0$. Nessa situação, a componente r_y denota a altura da

partícula.

Além disso, considera-se um elemento \mathcal{E} identificado com o plano XZ representando o chão. Sendo um plano fixo, esse elemento é completamente determinado ao se fornecerem um ponto e um versor normal a ele. Para o plano XZ , podem ser adotados a origem O do sistema como ponto pertencente ao plano e o versor normal $\hat{n} = (0, 1, 0)$.

Figura 4.5 – O problema da esfera quicando.



Fonte: do Autor.

Essa configuração pode ser representada como na [figura 4.5](#).

Além de servir como mais um caso para a validação do código, o problema da esfera quicando possibilita a comparação entre o coeficiente de restituição calculado a partir dos resultados da simulação e o coeficiente calculado analiticamente a partir do modelo de força escolhido.

Como consequência das equações [\(2.15\)](#) e [\(2.18\)](#), a configuração do problema implica que a velocidade da esfera relativa ao plano é a sua própria velocidade, já que o plano não se movimenta.

Com isso, sendo ε o coeficiente de restituição associado ao modelo de força de colisão escolhido, adota-se a expressão

$$\varepsilon = -\frac{(\dot{r}_y)' + g \cdot \Delta t'}{\dot{r}_y}, \quad (4.5)$$

sendo $(\dot{r}_y)'$ a velocidade da partícula \mathcal{P} após a colisão com o chão, e $\Delta t'$, o tempo em que os corpos permanecem em contato. O termo $g \cdot \Delta t'$ é uma compensação feita pelo fato de que, no problema em questão, a gravidade contribui para a variação da velocidade das partículas, o que não é considerado no cálculo analítico do coeficiente de restituição. Essa correção foi responsável por reduzir o erro aproximadamente pela metade nas simulações.

Para a solução desse problema, foram escolhidos dois casos: um conservativo ([seção 4.4.1](#)), em que ocorre a conservação da energia mecânica ao se considerarem nulas as constantes de amortecimento, e um dissipativo ([seção 4.4.2](#)), em que os elementos da simulação possuem constantes de amortecimento diferentes de zero.

4.4.1 Caso Conservativo

Para o estudo do problema da esfera quicando no caso conservativo, foram considerados os parâmetros de simulação expostos na [tabela 4.3](#). O modelo de interação considerado é o modelo do amortecimento linear corrigido, cuja expressão para a força é dada pela [equação \(2.22\)](#), enquanto o coeficiente de restituição é calculado pela [equação \(2.23\)](#).

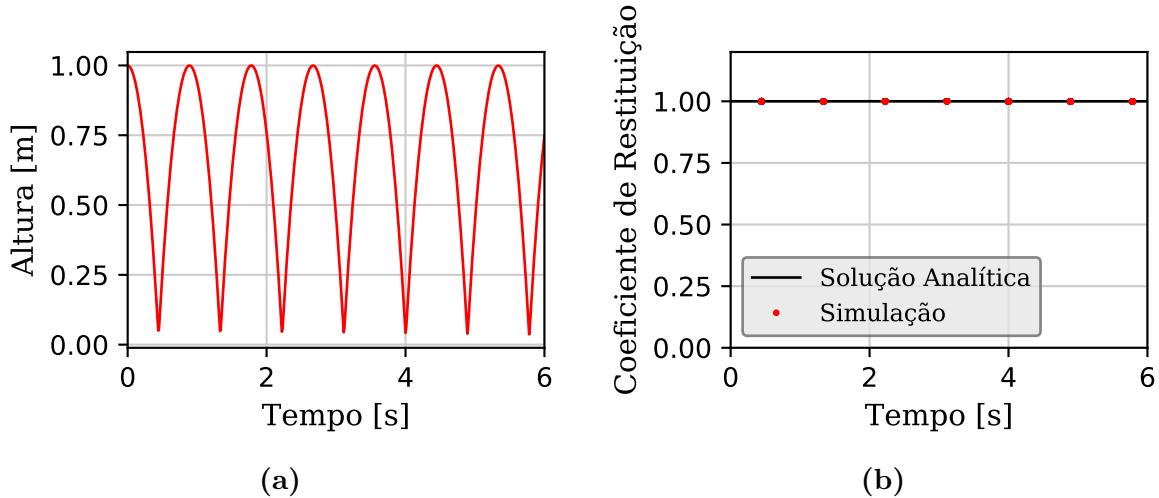
Tabela 4.3 – Parâmetros para o caso conservativo do problema da esfera quicando.

Modelo de força normal	Amortecedor linear corrigido
Gravidade	$g = 9,81 \text{ m/s}^2$
Massa da partícula	$m_p = 1 \text{ kg}$
Raio da partícula	$R_p = 3 \text{ cm}$
Constante elástica da partícula	$k_p^n = 10^9 \text{ N/m}$
Constante de amortecimento da partícula	$\gamma_p^n = 0 \text{ N s/m}$
Altura inicial da partícula	$r_y^i = 1 \text{ m}$
Velocidade inicial da partícula	$(\dot{r}_x^i, \dot{r}_y^i, \dot{r}_z^i) = (0, 0, 0) \text{ m/s}$
Aceleração inicial da partícula	$(\ddot{r}_x^i, \ddot{r}_y^i, \ddot{r}_z^i) = (0, 0, 0) \text{ m/s}^2$
Ponto da parede	$\mathbf{r}_\varepsilon = (0, 0, 0) \text{ m}$
Vensor normal à parede	$\hat{\mathbf{n}}_\varepsilon = (0, 1, 0) \text{ m}$
Constante elástica da parede	$k_\varepsilon^n = 10^9 \text{ N/m}$
Constante de amortecimento da parede	$\gamma_\varepsilon^n = 0 \text{ N s/m}$
Instante final	$t^f = 6 \text{ s}$
Passo de tempo	$\Delta t = 10^{-6} \text{ s}$
Ordem de extração	$k = 7$

Fonte: do Autor.

Nessa situação, os resultados obtidos pela simulação para a altura e para o coeficiente de restituição são obtidos como na [figura 4.6](#). Observa-se nessa figura que o coeficiente de restituição manteve-se bastante próximo de 1, o que corresponde ao resultado analítico, com erro máximo de 0,02 %. Além disso, energia se conserva, sendo constante com exceção dos instantes de colisão, em que a energia mecânica se transforma em energia puramente elástica. De fato, observou-se que a diferença entre a energia mecânica inicial E_M^i e a final E_M^f foi da ordem de 10^{-3} J , o que representa 0,036 % da energia inicial. Esses resultados estão expostos na [tabela 4.4](#). Convém destacar que a altura mostrada na [figura 4.6a](#) não se anula pois a mesma é medida com relação ao centro da partícula, e não à sua superfície.

Figura 4.6 – Solução numérica para a altura da partícula (a) e comparação entre as soluções numérica e analítica para o coeficiente de restituição (b) no caso conservativo do problema da esfera quicando.



Fonte: do Autor.

Tabela 4.4 – Energia mecânica e coeficiente de restituição resultantes do caso conservativo do problema da esfera quicando.

Energia mecânica inicial	$E_M^i = 9,81 \text{ J}$
Energia mecânica final	$E_M^f = 9,806\,447\,759\,55 \text{ J}$
Variação da energia mecânica	$\Delta E_M = E_M^i - E_M^f = 3,552 \text{ mJ}$
Percentual de energia dissipada	$\Delta E_M / E_M^i = 0,036 \%$
Coeficiente de restituição analítico	$\varepsilon = 1$
Erro máximo do coeficiente de restituição	$\text{Erro}(\varepsilon)_{\max} = 2 \cdot 10^{-4}$
Erro máximo percentual do coeficiente de restituição	$\text{Erro}(\varepsilon) / \varepsilon_{\max} = 0,02 \%$

Fonte: do Autor.

4.4.2 Caso Dissipativo

O caso conservativo, como exposto na [seção 4.4.1](#), apresenta conservação da energia mecânica em virtude do fato de que as constantes de amortecimento foram tomadas como nulas. Nesta seção, considera-se o caso dissipativo.

Para essa simulação, foram utilizados parâmetros idênticos aos indicados na [tabela 4.3](#) à exceção das constantes de amortecimento. Essas constantes são responsáveis pela dissipação da energia, e, para o caso dissipativo, devem assumir valores não nulos, como indicado na [tabela 4.5](#).

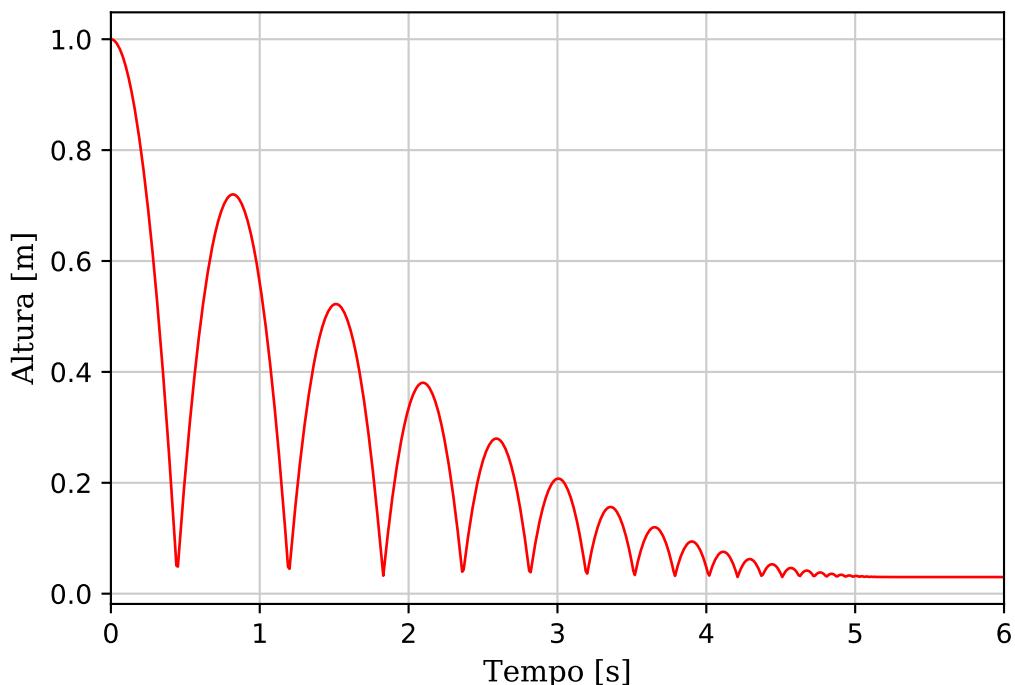
A partir desses parâmetros, uma simulação foi executada, resultando na altura da partícula em função do tempo como exposta na [figura 4.7](#).

Tabela 4.5 – Parâmetros para o caso dissipativo do problema da esfera quicando.

Constante de amortecimento da partícula	$\gamma_p^n = 5000 \text{ N s/m}$
Constante de amortecimento da parede	$\gamma_e^n = 5000 \text{ N s/m}$
Demais parâmetros	Tabela 4.3

Fonte: do Autor.

Figura 4.7 – Solução para a altura da partícula no caso dissipativo do problema da esfera quicando.

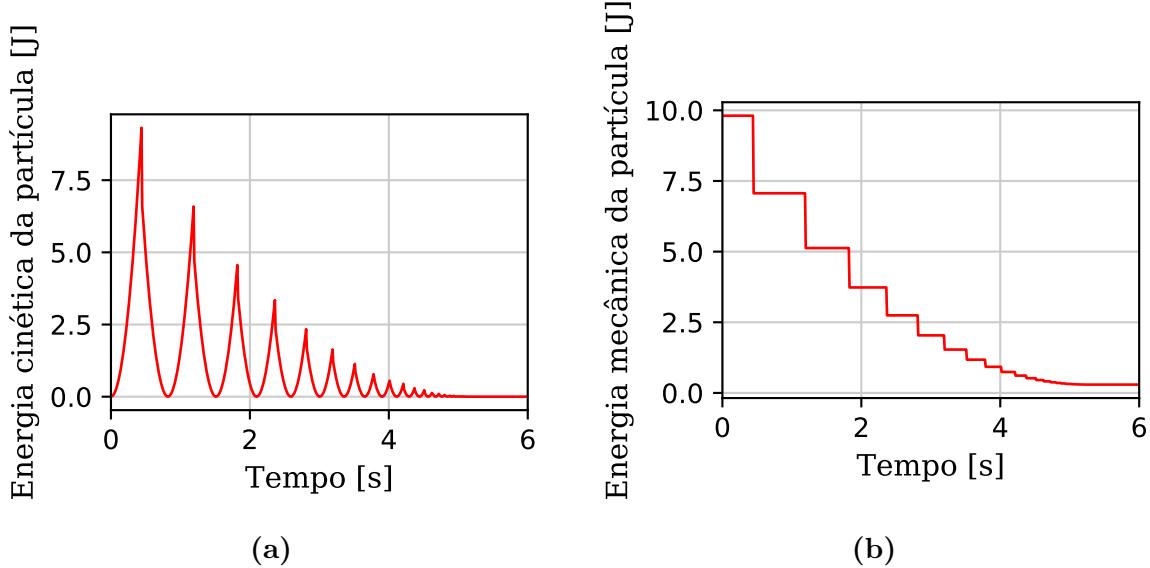


Fonte: do Autor.

Uma análise importante a se fazer nesse problema é a da energia. A energia cinética e a energia mecânica resultantes da simulação, calculadas de acordo com as [equações \(2.19\)](#) e [\(2.20\)](#), são apresentadas na [figura 4.8](#). Observa-se nos gráficos que a partícula experiente uma queda em sua energia mecânica após as colisões com o chão, sendo que a energia cinética anula-se aproximadamente após 5 s de simulação.

Além da energia, o coeficiente de restituição resultante da colisão também pode ser estudado. Na [figura 4.9](#), estão representados o coeficiente calculado numericamente através da [equação \(4.5\)](#) e o coeficiente analítico dado pela expressão [\(2.23\)](#). Observa-se uma boa concordância entre os resultados até aproximadamente 5 s de simulação. A divergência que ocorre após esse instante explica-se pelo fato de que a partícula entra em repouso sobre o chão.

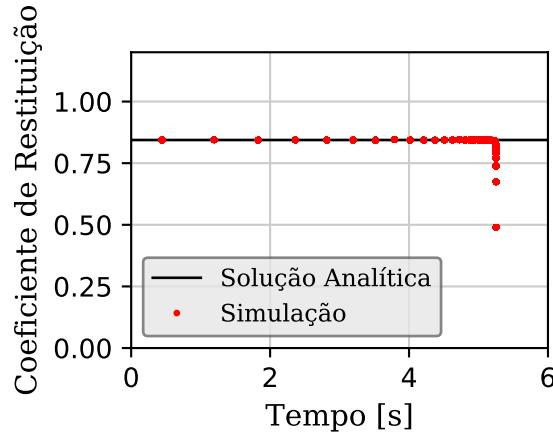
Figura 4.8 – Energia cinética (a) e energia mecânica (b) da partícula no caso dissipativo do problema da esfera quicando.



Fonte: do Autor.

Nota: Existe um detalhe observado na simulação que não é reproduzido nesses gráficos: durante a colisão, a energia cinética da partícula se anula, sendo parcialmente restituída depois. O motivo por que isso não é ilustrado é que, devido à quantidade de informações exportadas na simulação, a limitação da memória do computador impede a reprodução dos resultados em todos os instantes de tempo.

Figura 4.9 – Coeficiente de restituição resultante da colisão dissipativa do problema da esfera quicando.



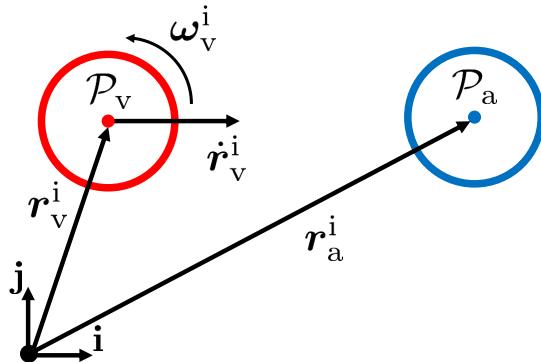
Fonte: do Autor.

4.5 Colisão entre Esferas

A questão central dos métodos de elementos discretos é a interação entre partículas, situação em que existem duas ou mais equações de movimento a serem resolvidas

simultaneamente. Nesta seção, é apresentado o problema da colisão entre duas esferas.

Figura 4.10 – O problema da colisão entre esferas.



Fonte: do Autor.

Consideram-se duas partículas esféricas: a partícula vermelha \mathcal{P}_v e a azul \mathcal{P}_a . A partícula azul encontra-se inicialmente em repouso na posição \mathbf{r}_a^i . A partícula vermelha encontra-se inicialmente na posição \mathbf{r}_v^i e avança, com uma velocidade linear $\dot{\mathbf{r}}_v^i$ e uma velocidade angular $\omega_v^i = \omega_v^i \cdot \mathbf{k}$, na direção de \mathcal{P}_a . As partículas colidem, e ocorre transferência de movimento da partícula vermelha para a azul. Essa situação é representada na figura 4.10.

Nesse problema, não é considerada a ação da gravidade, o que permite a análise simples das quantidades de movimento linear e angular. Como no problema da esfera quicando, são consideradas duas situações: o caso conservativo e o caso dissipativo. Devido à natureza das forças tangenciais apresentadas na seção 2.3, simulações com rotações são intrinsecamente dissipativas, e por isso esse caso só é considerado na seção 4.5.2.

4.5.1 Caso Conservativo

Para a simulação com conservação da energia mecânica, foram arbitrados os parâmetros expostos na tabela 4.6. Novamente, foi escolhido o modelo de força normal de amortecimento linear pela facilidade de se examinar o coeficiente de restituição resultante da colisão.

Com esses parâmetros, foi executada a simulação conservativa. A posição e a velocidade das partículas são apresentadas na figura 4.11. Também estão representadas, na mesma figura, a quantidade de movimento linear horizontal, a força de contato horizontal entre as partículas, a energia cinética e o coeficiente de restituição resultante do choque.

Como pode ser observado, os resultados aproximam-se bastante do esperado. Por se tratar de uma colisão sem termos dissipativos, a partícula vermelha transfere todo o seu movimento à azul, entrando em repouso após a colisão. A figura 4.11f e a tabela 4.7

indicam que o coeficiente de restituição numericamente calculado praticamente se iguala ao analítico com valor unitário, o que implica uma colisão perfeitamente elástica. Mais ainda, infere-se das figuras 4.11c e 4.11e que ocorre a conservação da quantidade de movimento e a conservação da energia. Além disso, chama a atenção o fato de que parte da energia cinética é transformada em energia elástica durante o choque, e é restituída finda a colisão. Também é interessante notar que a força de colisão entre as partículas, indicada na figura 4.11d, ultrapassa os 150 kN em módulo.

Tabela 4.6 – Parâmetros para o caso conservativo do problema da colisão entre esferas.

Modelo de força normal	Amortecedor linear corrigido
Massa da partícula vermelha	$m_v = 1 \text{ kg}$
Raio da partícula vermelha	$R_v = 3 \text{ cm}$
Constante elástica da partícula vermelha	$k_v^n = 10^9 \text{ N/m}$
Constante de amortecimento normal da partícula vermelha	$\gamma_v^n = 0 \text{ N s/m}$
Posição inicial da partícula vermelha	$\mathbf{r}_v^i = (0, 0, 0) \text{ m}$
Velocidade inicial da partícula vermelha	$\dot{\mathbf{r}}_v^i = (10, 0, 0) \text{ m/s}$
Aceleração inicial da partícula vermelha	$\ddot{\mathbf{r}}_v^i = (0, 0, 0) \text{ m/s}^2$
Massa da partícula azul	$m_a = 1 \text{ kg}$
Raio da partícula azul	$R_a = 3 \text{ cm}$
Constante elástica da partícula azul	$k_a^n = 10^9 \text{ N/m}$
Constante de amortecimento normal da partícula azul	$\gamma_a^n = 0 \text{ N s/m}$
Posição inicial da partícula azul	$\mathbf{r}_a^i = (62, 0, 0) \text{ mm}$
Velocidade inicial da partícula azul	$\dot{\mathbf{r}}_a^i = (0, 0, 0) \text{ m/s}$
Aceleração inicial da partícula azul	$\ddot{\mathbf{r}}_a^i = (0, 0, 0) \text{ m/s}^2$
Instante final	$t^f = 1 \text{ ms}$
Passo de tempo	$\Delta t = 10^{-8} \text{ s}$
Ordem de extração	$k = 7$

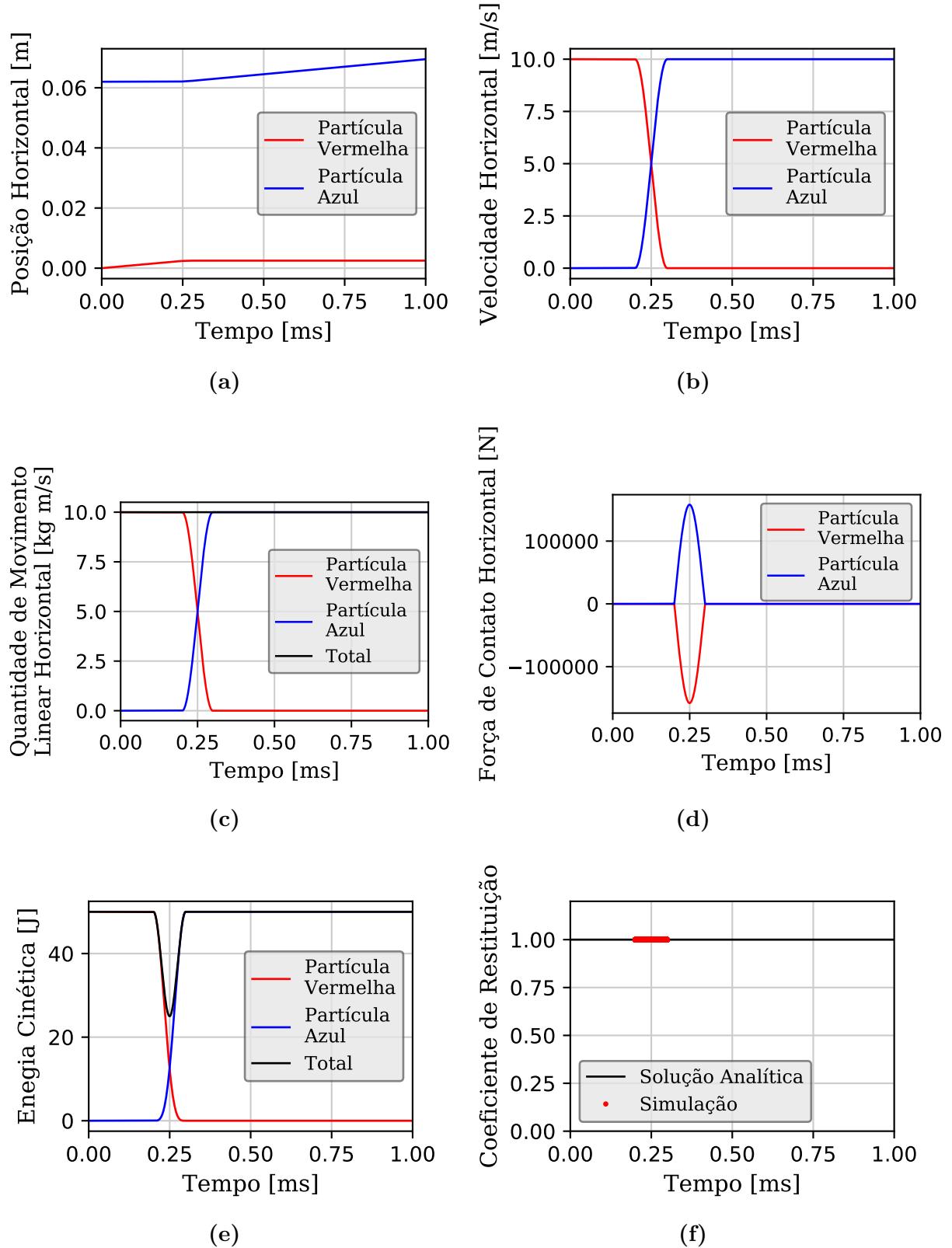
Fonte: do Autor.

Tabela 4.7 – Coeficiente de restituição resultante do caso conservativo do problema das esferas colidentes.

Coeficiente de restituição analítico	$\varepsilon = 1$
Erro máximo do coeficiente de restituição	$\text{Erro}_{\max}(\varepsilon) = 9 \cdot 10^{-9}$

Fonte: do Autor.

Figura 4.11 – Solução para a posição horizontal (a), para a velocidade horizontal (b), para quantidade de movimento linear (c), para a força de contato (d), para a energia cinética (e) e para o coeficiente de restituição (f) da simulação conservativa do problema das esferas colidentes.



Fonte: do Autor.

4.5.2 Caso Dissipativo

No caso dissipativo, são atribuídas às partículas constantes de amortecimento não nulas. A simulação dissipativa pode ser separada em dois casos: sem rotação e com rotação.

4.5.2.1 Sem Rotação

Para a simulação dissipativa sem rotação, consideram-se os parâmetros da [tabela 4.8](#). Os resultados desse caso podem ser compilados como na [figura 4.12](#).

Tabela 4.8 – Parâmetros para o caso dissipativo sem rotação do problema da colisão entre esferas.

Constante de amortecimento da partícula vermelha	$\gamma_v^n = 10000 \text{ N s/m}$
Constante de amortecimento da partícula azul	$\gamma_a^n = 10000 \text{ N s/m}$
Demais parâmetros	Tabela 4.6

Fonte: do Autor.

Por fim, a simulação foi repetida com os parâmetros da [tabela 4.8](#) à exceção do passo tempo, que foi variado. Foram executadas simulações para

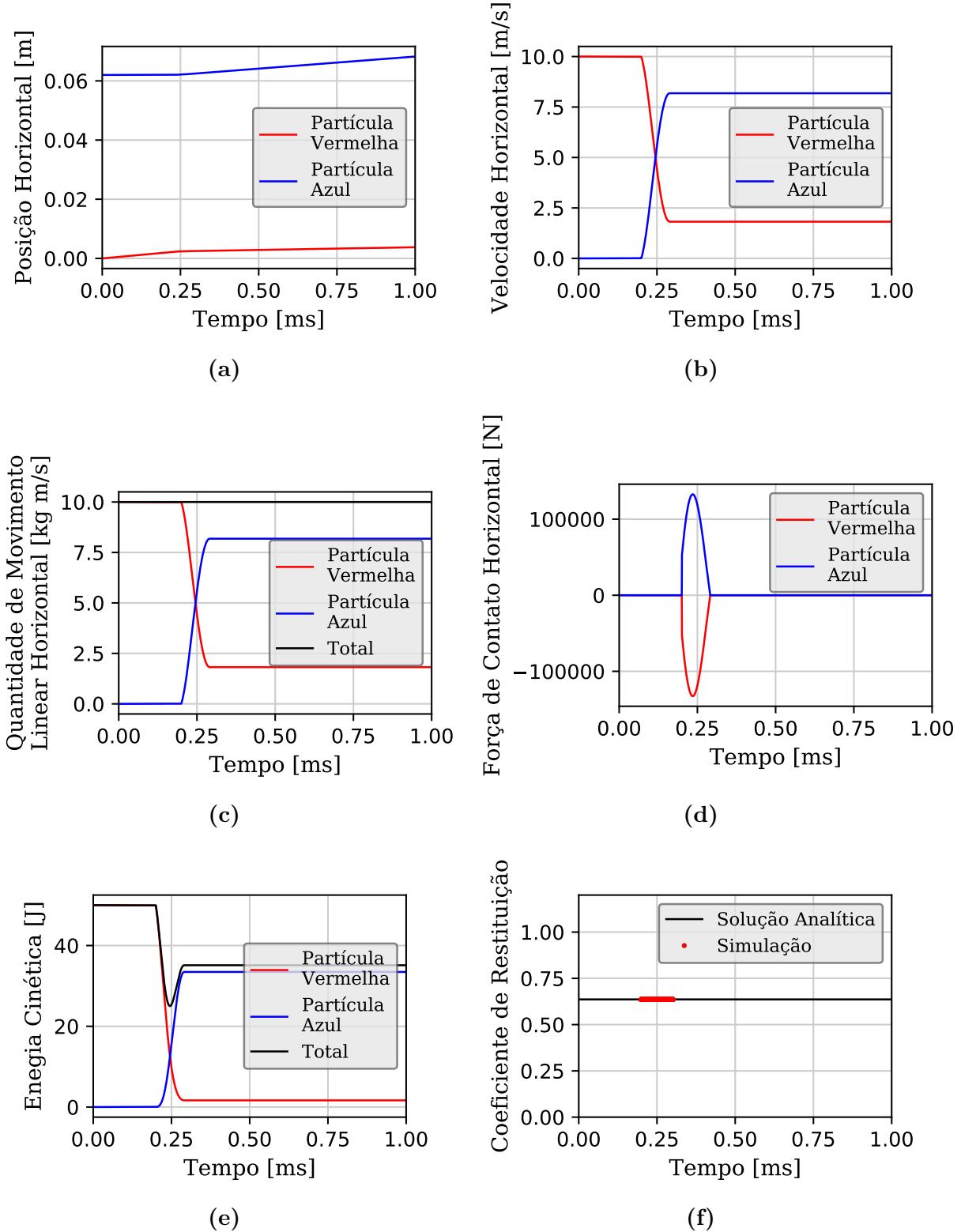
$$\Delta t = 10^{-8} \text{ s}, \quad \sqrt{2} \cdot 10^{-8} \text{ s}, \quad 2 \cdot 10^{-8} \text{ s}, \quad 2\sqrt{2} \cdot 10^{-8} \text{ s}, \quad 4 \cdot 10^{-8} \text{ s}, \quad \dots, \quad 1024 \cdot 10^{-8} \text{ s},$$

e o erro máximo do coeficiente de restituição foi computado. O resultado desse processo está exposto, em escala logarítmica, na [figura 4.13](#).

Dentre as características que se observam nesses resultados, nota-se que a partícula vermelha, ao contrário do caso sem amortecimento, não entra em repouso após a colisão, mas, sim, continua a sua trajetória com uma velocidade reduzida. Também, a energia cinética total do sistema não se conserva, sofrendo uma queda de aproximadamente 16,5 J do início até o fim da simulação. Embora ocorra a perda de energia, a quantidade de movimento linear mantém-se constante, como deve acontecer em qualquer sistema sem forças externas, mesmo que existam forças internas dissipativas. O coeficiente de restituição obtido foi de 0,63, apresentando um erro de apenas $3,18 \cdot 10^{-5}$ com relação ao coeficiente calculado analiticamente. Além dessas considerações, percebe-se, comparando-se as figuras [4.11d](#) e [4.12d](#), que ocorre uma alteração na curva da força de contato horizontal. No caso conservativo, existe uma simetria do gráfico com relação ao instante de máxima superposição, o que não se observa na força do caso dissipativo. Esse deslocamento no perfil de força, com intensidade maior na aproximação que no afastamento, é o responsável pela dissipaçāo de energia.

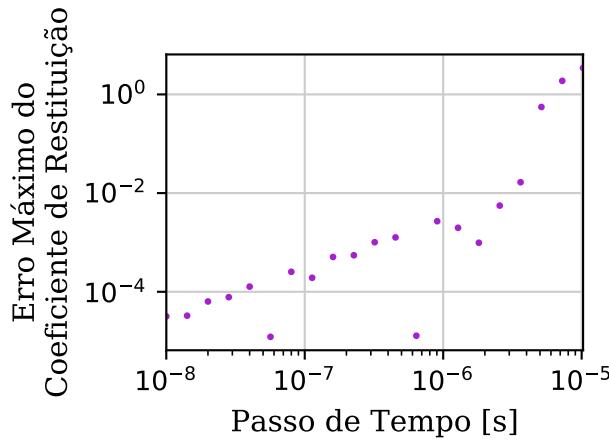
Por fim, observa-se que o erro máximo do coeficiente de restituição cresce com o aumento do passo de tempo, chegando a um erro de mais de 100% para passos de aproximadamente 10^{-5} s , o que corresponde à ordem de grandeza do tempo de colisão.

Figura 4.12 – Solução da posição (a), da velocidade (b), da quantidade de movimento linear (c), da força de contato horizontal (d), da energia cinética (e) e do coeficiente de restituição (f) do par de partículas colidentes na simulação dissipativa sem rotação.



Fonte: do Autor.

Figura 4.13 – Erro máximo do coeficiente de restituição em função do passo de tempo no caso dissipativo do problema das esferas colidentes sem rotação.



Fonte: do Autor.

4.5.2.2 Com Rotação

Para a colisão de partículas com rotação, utilizou-se o modelo de interação de Haff e Werner, que é o análogo rotacional do modelo do amortecedor linear, e cuja expressão para a força tangencial é dada pela [equação \(2.25\)](#). Para a força normal, o modelo adotado ainda é o do amortecedor linear corrigido. Como entrada para a simulação, foram aplicados os parâmetros da [tabela 4.9](#). O momento de inércia das partículas é consequência de sua massa e seu raio, de acordo com a [equação \(2.8\)](#).

Com esses parâmetros, foram obtidos os resultados expostos nas [figuras 4.14 a 4.16](#). Observa-se, na [figura 4.14](#), que, na colisão, a rotação da partícula vermelha tem o efeito de iniciar um movimento na direção vertical. Ainda, em virtude da não elasticidade do choque, a partícula vermelha mantém-se em movimento mesmo após a colisão com a azul. Na [figura 4.15](#), estão expostas as curvas da quantidade de movimento linear e da quantidade de movimento angular. Novamente, a simulação apresentou resultados fisicamente verossímeis, visto que ocorre a conservação dessas quantidades. Na mesma figura, estão representados os gráficos da força e do torque sobre as partículas. Nota-se que a força horizontal atinge a casa de 100 kN, enquanto a vertical alcança 30 kN. As partículas expericienciam torques resultantes iguais em razão de serem, elas próprias, idênticas.

Por sua vez, na [figura 4.16](#), são expostas as curvas de energia e o coeficiente de restituição obtidos. Observa-se que, por conta da dissipação, todas as formas de energia sofrem redução. Devido ao modelo intrinsecamente dissipativo para a força tangencial, não existe energia rotacional armazenada elasticamente. A consequência disso é que não há recuperação da energia rotacional após a colisão, como se vê na [figura 4.16a](#).

Por fim, na [figura 4.16d](#), apresenta-se o coeficiente de restituição. Mais uma vez, é

observada a sua correspondência com o valor analítico. A [tabela 4.10](#) sumariza os resultados para a energia e para o coeficiente de restituição.

Tabela 4.9 – Parâmetros para o caso dissipativo do problema da colisão entre esferas considerando rotações.

Modelo de força normal	Amortecedor linear corrigido
Modelo de força tangencial	Modelo de Haff e Werner
Momento de inércia da partícula vermelha	$J_v = 3,6 \text{ kg cm}^2$
Velocidade angular da partícula vermelha	$\omega_v^i = 100 \text{ rad/s}$
Constante de amortecimento normal da partícula vermelha	$\gamma_v^n = 10\,000 \text{ N s/m}$
Constante de amortecimento tangencial da partícula vermelha	$\gamma_v^t = 10\,000 \text{ N s/m}$
Coeficiente de atrito associado à partícula vermelha	$\mu_v = 0,95$
Momento de inércia da partícula azul	$J_a = 3,6 \text{ kg cm}^2$
Constante de amortecimento normal da partícula azul	$\gamma_a^n = 10\,000 \text{ N s/m}$
Constante de amortecimento tangencial da partícula azul	$\gamma_a^t = 10\,000 \text{ N s/m}$
Coeficiente de atrito associado à partícula azul	$\mu_a = 0,95$
Demais parâmetros	Tabela 4.6

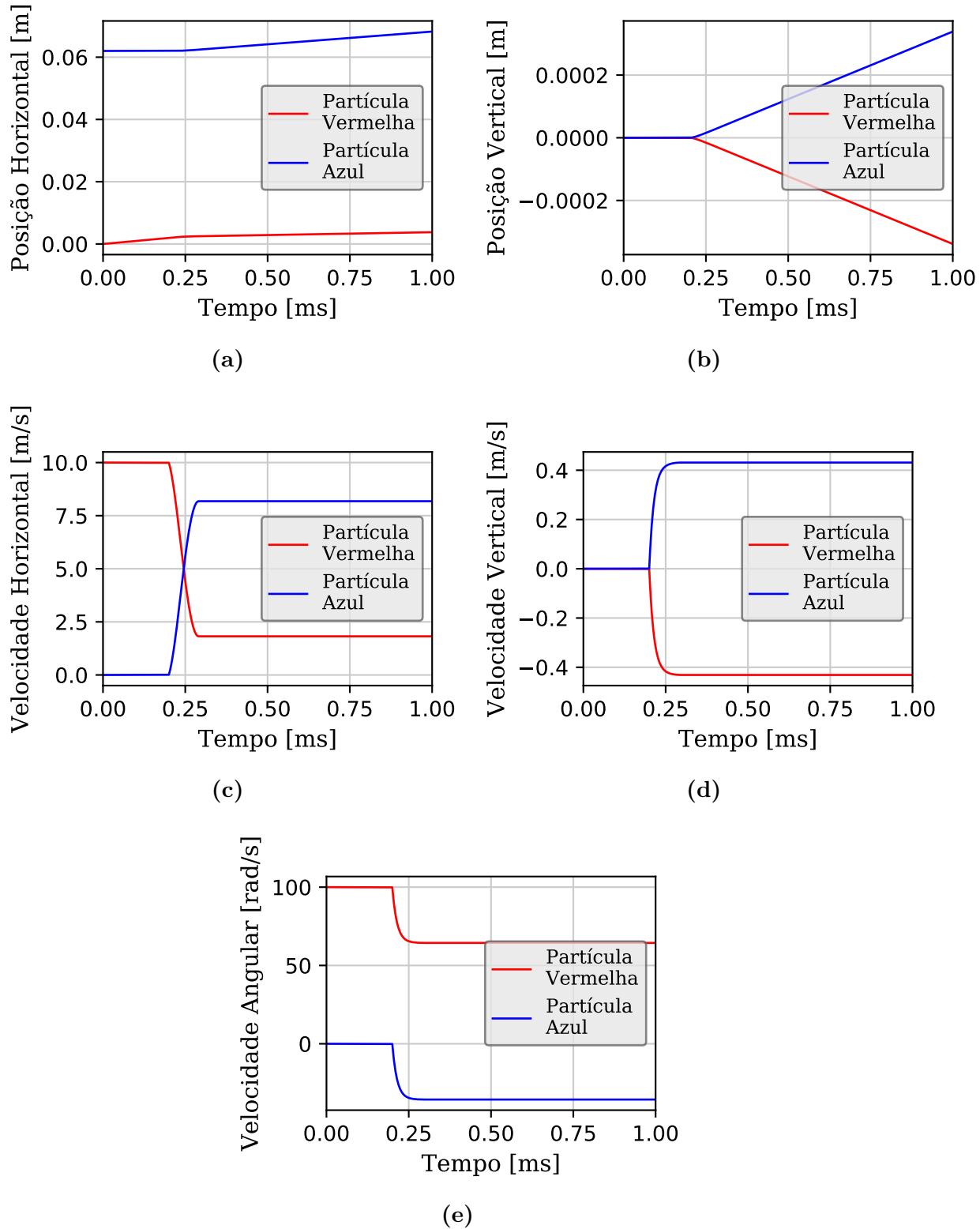
Fonte: do Autor.

Tabela 4.10 – Energia mecânica e coeficiente de restituição resultantes do caso dissipativo do problema da colisão de esferas com rotação.

Energia mecânica inicial	$E_M^i = 51,80 \text{ J}$
Energia mecânica final	$E_M^f = 33,78 \text{ J}$
Variação da energia mecânica	$\Delta E_M = E_M^i - E_M^f = 18 \text{ J}$
Percentual de energia dissipada	$\Delta E_M / E_M^i = 34,78 \%$
Coeficiente de restituição analítico	$\varepsilon = 0,64$
Erro máximo do coeficiente de restituição	$\text{Erro}_{\max}(\varepsilon) = 9,8 \cdot 10^{-6}$
Erro máximo relativo do coeficiente de restituição	$\text{Erro}_{\max}(\varepsilon) / \varepsilon = 1,5 \cdot 10^{-6}$

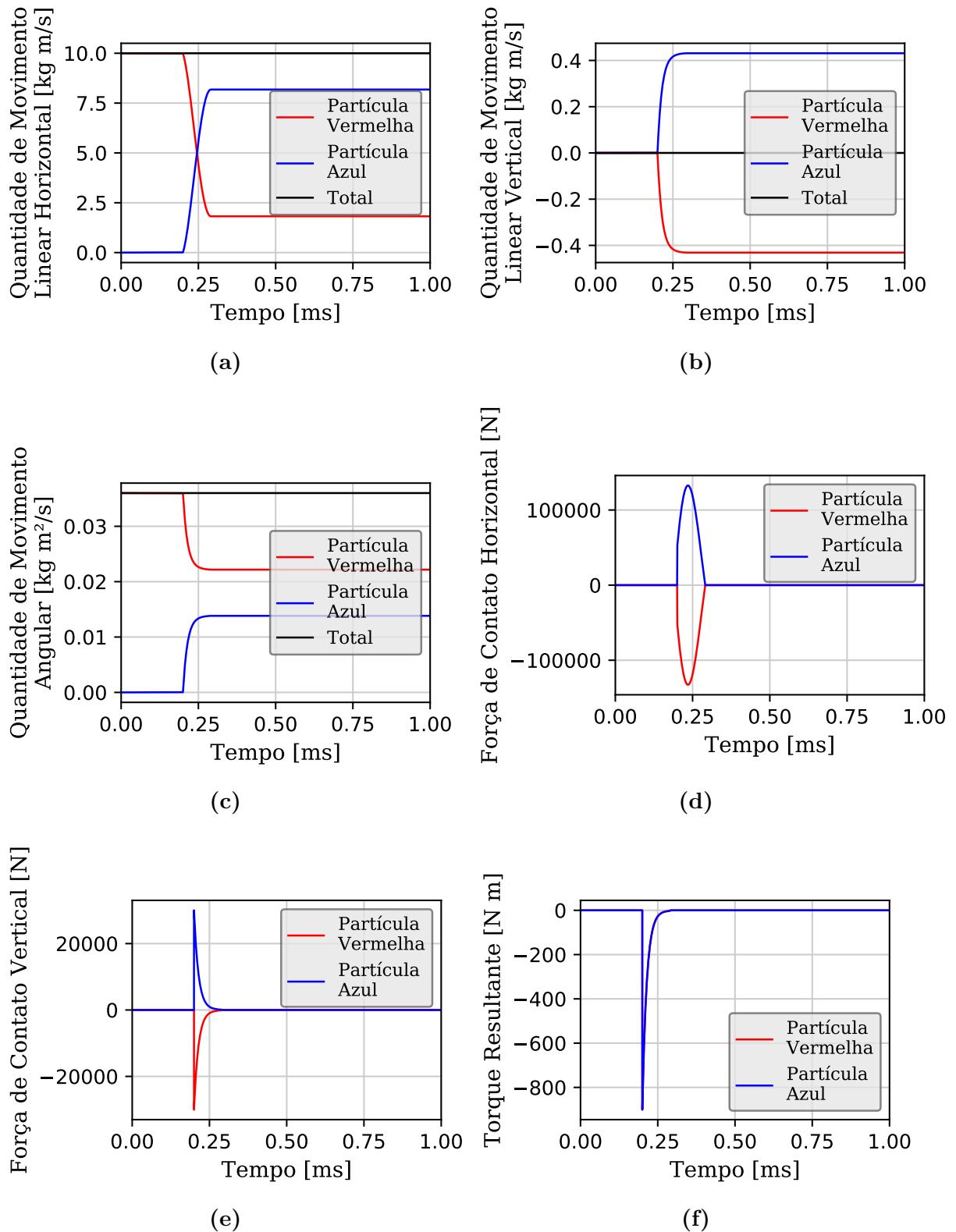
Fonte: do Autor.

Figura 4.14 – Solução das posições horizontal (a) e vertical (b), das velocidades horizontal (c) e vertical (d), e da velocidade angular (e) das partículas no caso dissipativo do problema da colisão de esferas com rotação.



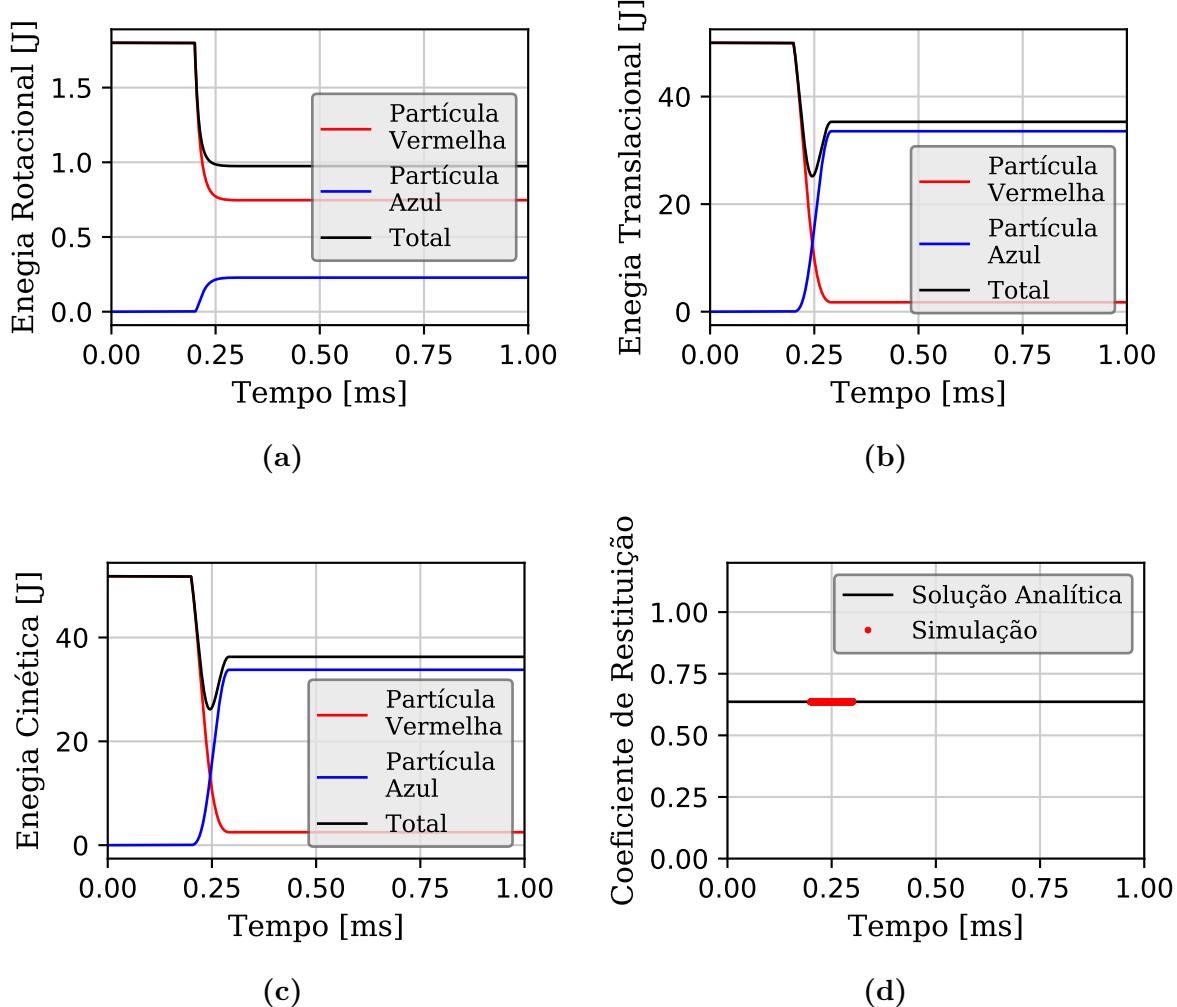
Fonte: do Autor.

Figura 4.15 – Solução numérica das quantidades de movimento linear (a e b) e angular (c), para a força de colisão (d e e) e para o torque resultante (f) no caso dissipativo do problema das esferas colidentes com rotação.



Fonte: do Autor.

Figura 4.16 – Energias rotacional (a), translacional (b) e cinética (c) e coeficiente de restituição (d) no caso dissipativo do problema das esferas colidentes com rotação.



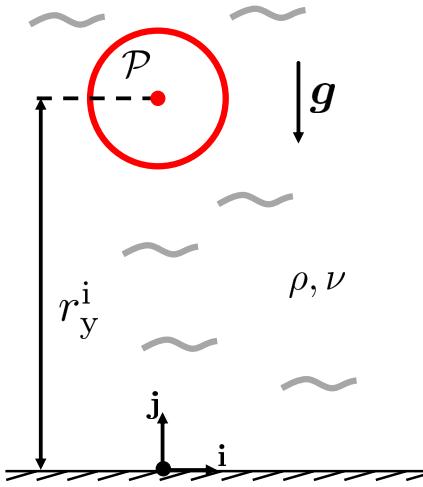
Fonte: do Autor.

4.6 Queda com Arrasto

Como afirmado na [seção 1.5](#), um dos objetivos deste trabalho é possibilitar o uso de interações de naturezas distintas e o acoplamento entre a dinâmica de partículas e a mecânica de fluidos. Assim, esta seção é dedicada a demonstrar a capacidade do simulador de lidar com esse tipo de problema.

Considera-se uma partícula esférica \mathcal{P} de raio R e massa m liberada, em repouso, na posição $\mathbf{r} = (0, r_y, 0)$. A partícula sofre a ação da gravidade $\mathbf{g} = (0, -g, 0)$ com $g > 0$. A esfera é envolvida por ar atmosférico, e, por isso, sofre a ação da força de arrasto durante a sua queda. Essa configuração é representada pela [figura 4.17](#).

Figura 4.17 – O problema da queda com arrasto.



Fonte: do Autor.

Segundo Fox, Pritchard e McDonald (2009, p. 392), a força de arrasto \mathbf{F}_D que a partícula sofre é dada por

$$\mathbf{F}_D = C_D \frac{1}{2} \rho A \|\dot{\mathbf{r}}\| \dot{\mathbf{r}},$$

em que C_D é o coeficiente de arrasto do par partícula-fluido, ρ é a massa específica do ar e A é a área projetada da partícula no plano normal ao seu movimento. Para a esfera, a área projetada é dada por $A = \pi R^2$.

O coeficiente de arrasto não é constante, mas varia em função do número de Reynolds (Re) definido, para partículas esféricas, como

$$Re = \frac{\dot{r}_{\text{rel}} \cdot D}{\nu_f},$$

sendo \dot{r}_{rel} a norma da velocidade relativa entre a partícula e o fluido, D , o diâmetro da partícula, e ν_f , a viscosidade cinemática do ar. Para o problema em questão, supõe-se que o ar está em repouso, do que segue que a velocidade relativa é idêntica à velocidade \dot{r} da partícula.

Apesar dessa dependência, o coeficiente de arrasto apresenta pouca variação para números de Reynolds entre 1000 e 300 000 se forem consideradas partículas esféricas lisas (FOX; PRITCHARD; McDONALD, 2009, p. 397). Para uma esfera de 2 cm de raio, e considerando o ar com viscosidade cinemática de $\nu_f = 1,5 \cdot 10^{-5} \text{ m}^2/\text{s}$, esses limites correspondem às velocidades de $\dot{r}_{\text{rel}} = 0,375 \text{ m/s}$ e $\dot{r}_{\text{rel}} = 112,5 \text{ m/s}$. Por simplicidade, considera-se, aqui, que o coeficiente de arrasto é constante mesmo para velocidades inferiores a 0,375 m/s. Essa aproximação é razoável tendo em vista que, para o valor da gravidade de $9,81 \text{ m/s}^2$, a partícula leva aproximadamente 76 ms para atingir essa velocidade mínima, sendo que o resto do movimento ocorre dentro da faixa de C_D constante.

Com isso, e a partir dos resultados expostos em Fox, Pritchard e McDonald (2009, p. 397), foi fixado o valor $C_D = 0,5$.

A velocidade terminal \dot{r}_t da partícula é aquela para a qual não ocorre mais a aceleração da partícula. Nessa situação, a força de arrasto se iguala, em módulo, à força peso:

$$m g = C_D \frac{1}{2} \rho A \dot{r}_t^2 \iff \dot{r}_t = \sqrt{\frac{2 m g}{C_D \rho A}}, \quad (4.6)$$

e assim se obtém a expressão analítica para a velocidade terminal.

A fim de se simular esse problema, foram considerados os parâmetros expostos na [tabela 4.11](#).

Tabela 4.11 – Parâmetros para o problema da queda com arrasto.

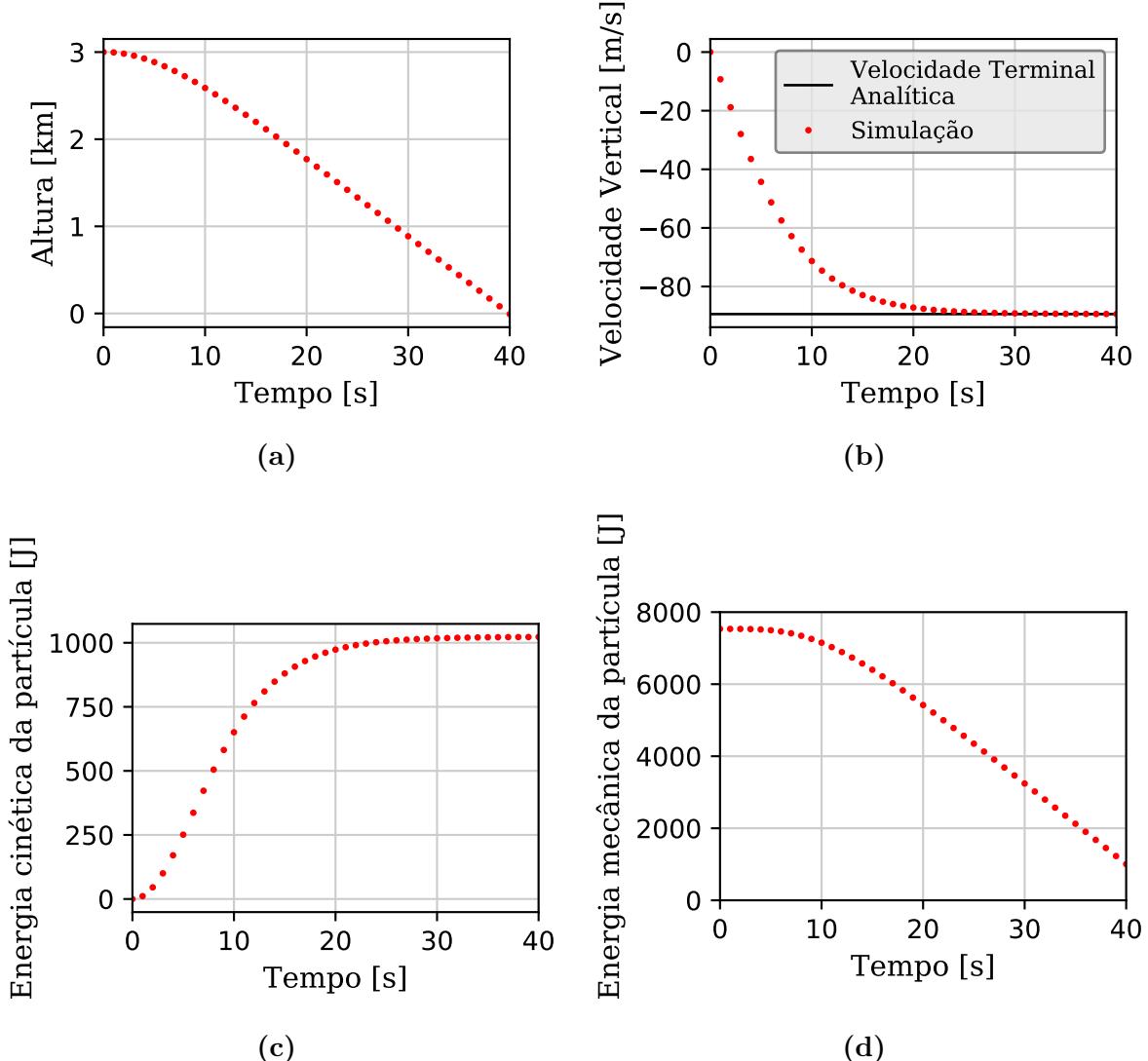
Gravidade	$g = 9,81 \text{ m/s}^2$
Massa da partícula	$m = 256 \text{ g}$
Raio da partícula	$R = 2 \text{ cm}$
Altura inicial da partícula	$r_y^i = 3 \text{ km}$
Velocidade inicial da partícula	$(\dot{r}_x^i, \dot{r}_y^i, \dot{r}_z^i) = (0, 0, 0) \text{ m/s}$
Aceleração inicial da partícula	$(\ddot{r}_x^i, \ddot{r}_y^i, \ddot{r}_z^i) = (0, 0, 0) \text{ m/s}^2$
Densidade do ar	$\rho = 1 \text{ kg/m}^3$
Coeficiente de arrasto	$C_D = 0,5$
Instante final	$t^f = 40 \text{ s}$
Passo de tempo	$\Delta t = 0,1 \text{ s}$
Ordem de extração	$k = 7$

Fonte: do Autor.

Com esses valores, foi executada uma simulação, e a altura da partícula em função do tempo foi obtida tal como na [figura 4.18a](#). A velocidade vertical, por sua vez, está ilustrada na [figura 4.18b](#), sendo que a velocidade terminal é calculada analiticamente pela [equação \(4.6\)](#). A energia cinética e a energia mecânica, calculadas de acordo com as equações [\(2.19\)](#) e [\(2.20\)](#), estão expostas nas figuras [4.18c](#) e [4.18d](#).

Observa-se que a velocidade da partícula converge para a velocidade terminal prevista analiticamente, evidenciando o correto comportamento do programa mesmo em um problema envolvendo a interação entre fluido e partícula. A energia cinética cresce em função da transformação da energia potencial em movimento. Entretanto esse crescimento é retardado até cessar aproximadamente em 30 s de simulação. A queda na energia mecânica da partícula, por sua vez, pode ser explicada pela dissipação provocada pelo arrasto. Como se observa, a força de arrasto é responsável por dissipar mais de 6 kJ em 40 s de queda.

Figura 4.18 – Soluções para a altura da partícula (a), sua velocidade vertical (b), sua energia cinética (c) e sua energia mecânica (d) da partícula no problema de queda com arrasto.



Fonte: do Autor.

A importância dessa simulação reside no fato de que, embora o problema considerado seja simples e de fácil solução analítica, ele representa o transporte de informação do fluido para o elemento discreto, como indicado na [seção 3.7](#).

5 Conclusão

5.1 Sumário

Neste trabalho, foram apresentados os fundamentos do Método de Elementos Discretos, a implementação de uma biblioteca de códigos e resultados de simulação.

No [capítulo 2](#), foram apresentados os aspectos matemáticos, como as equações de movimento baseadas nas leis de Euler para o movimento, os modelos de força de colisão e o algoritmo de integração de Gear. Dentre os modelos de força, foram detalhadas as expressões de cálculo e o significado físico dos parâmetros para os modelos de amortecimento linear e de esferas viscoelásticas, para a força normal, e os modelos de Haff e Werner e de Cundall e Strack, para a força tangencial. Também foi apresentado o algoritmo de Gear, responsável pela solução das equações diferenciais em cada passo de tempo.

Por sua vez, no [capítulo 3](#), foram apresentados os métodos de elementos discretos em maiores detalhes. Foram apresentados a discretização temporal e os principais elementos pertencentes às simulações. Cada etapa do algoritmo foi explicada, demonstrando, com base em fluxogramas e listagens, o seu funcionamento. O primeiro passo do algoritmo é a inicialização, em que a configuração da simulação é obtida de arquivos de entrada. Depois, inicia-se o laço temporal em que são resolvidas, para cada instante do tempo discretizado, as equações de movimento. A solução das equações de movimento foi descrita e uma listagem, apresentada. A exportação de dados, também, é um passo essencial na simulação visto que permite, de fato, a obtenção dos resultados simulados. O laço de simulação é executado repetidamente até que um critério de parada previamente estabelecido seja satisfeito, levando o programa à etapa de finalização.

Ainda, foi apresentada a importância dos métodos de monitoramento de vizinhanças para simulações com elevado número de partículas. Também foi discutida a limitação do passo de tempo que, embora necessária, não segue, necessariamente, um método, dependendo mais da experiência do usuário que de critérios matemáticos rigorosos. Além disso, foram indicadas as condições de contorno e de restrição mais usuais, como a condição de movimento prescrito aos elementos de contorno, de paredes reflexivas para as bordas do domínio e de restrição de movimento relativo entre subelementos de partículas compostas. Mais ainda, no [capítulo 3](#) foram apresentados aspectos mais específicos concernentes à simulação da fragmentação de partículas e à paralelização dos métodos de elementos discretos. Por fim, foi apresentado brevemente o acoplamento que pode ser feito entre o DEM e outras famílias de métodos computacionais, como o Método de Elementos Finitos e a Mecânica de Fluidos Computacional.

A implementação da biblioteca e do simulador utilizados na obtenção de resultados foi então apresentada no [capítulo 4](#). A partir disso, foram considerados quatro problemas: o problema do lançamento oblíquo, o da esfera quicando, o da colisão entre esferas e, por fim, o da queda com arrasto.

5.2 Conclusões

A análise dos resultados obtidos no [capítulo 4](#) indica uma grande concordância entre as simulações e as soluções analíticas dos problemas considerados.

O primeiro problema tratou do lançamento oblíquo de uma partícula. Embora relativamente simples e de fácil resolução analítica, esse problema permite a validação do simulador em problemas em que a força atuante sobre as partículas é constante. De fato, o erro apresentado pelos resultados numéricos é da ordem do erro de máquina, o que é a máxima exatidão que se pode exigir de um algoritmo computacional.

O segundo problema, ligeiramente mais complexo, inclui, além da gravidade, um elemento plano correspondente ao chão com o qual a partícula colide. Com esse problema, foi possível a análise da energia da partícula e do coeficiente de restituição resultante das colisões em um caso conservativo, em que não há componentes dissipativas na expressão para a força, e um caso dissipativo, em que essas componentes são incluídas. Em ambas as situações, os resultados numéricos reproduziram o comportamento analítico do sistema. Observou-se, porém, uma queda de energia no caso conservativo, o que não existe na solução analítica. Entretanto esse erro e o erro do coeficiente de restituição são, cada um, quatro ordens de magnitude inferiores aos valores esperados, o que pode ser explicado pela inexatidão das aproximações consideradas no método numérico empregado. Também é interessante notar a discrepância entre os coeficientes de restituição analítico e numérico no caso dissipativo quando a partícula se aproxima do repouso. Isso se deve ao fato de que a velocidade relativa entre os elementos colidentes tende a zero, e a computação do coeficiente de restituição deixa de fazer sentido. Esse repouso é atingido em decorrência da dissipação de energia mecânica nos choques como resultado da introdução das constantes de amortecimento no cálculo das forças.

No terceiro problema foi abordada a colisão entre duas partículas em três casos: um conservativo, um dissipativo sem rotações e outro dissipativo com rotações. O caso conservativo demonstrou um excelente desempenho, em que tanto a quantidade de movimento linear quanto a energia se conservaram, como esperado. O erro máximo do coeficiente de restituição atingiu, nessa situação, uma fração de bilionésimos da solução analítica. É interessante, ainda, observar-se que a força de contato entre as esferas ultrapassou 100 kN em módulo como consequência da rigidez das partículas, do modelo de interação usado e da configuração do problema. Também ficou evidente a transformação de energia cinética

em energia elástica e vice-versa durante o choque.

No caso dissipativo sem rotação do terceiro problema, observou-se, mais uma vez, a conservação da quantidade de movimento linear visto que esta independe do modelo de força usado, mas segue da Lei de Ação e Reação. A energia cinética, por sua vez, reduziu-se como resultado da dissipação na colisão. Novamente, a força máxima no contato ultrapassou os 100 kN em módulo, mas, dessa vez, apresentou um perfil diferente, assimétrico, resultante da dissipação de energia. Ainda assim, o coeficiente de restituição numérico aproximou-se bastante do analítico, evidenciando a corretude do algoritmo. É interessante ainda observar o caráter crescente do erro máximo do coeficiente de restituição em função do passo de tempo empregado. Esse erro vai da ordem de milionésimos para simulações com $\Delta t = 10^{-8}$ s para mais que 100 % quando $\Delta t = 10^{-5}$ s, caso em que o passo de tempo se aproxima do tempo de colisão.

No caso dissipativo com rotação, observou-se, mais uma vez, o comportamento esperado para as grandezas translacionais: a quantidade de movimento linear se conservou e houve dissipação de energia, enquanto o coeficiente de restituição manteve-se próximo do analítico. Além disso, notou-se, nessa situação, a conservação da quantidade de movimento angular e uma queda na energia rotacional. Por não haver componente elástica no modelo de força tangencial, não houve recuperação de energia rotacional ao fim da colisão. Percebe-se, ainda, que a força tangencial consequente da rotação dá início a um movimento vertical, sendo que, também nessa direção, a quantidade de movimento linear se conserva. Mais uma vez, a força de contato horizontal atinge a casa dos 100 kN, ao passo que a força de contato vertical ultrapassa os 20 kN.

Por fim, foi apresentado o problema da queda com arrasto, em que uma partícula foi liberada no espaço a 3 km de altura sob ação da força da gravidade e da resistência do ar. Da comparação do histórico da velocidade da partícula com a velocidade terminal analiticamente calculada, infere-se a corretude do algoritmo mesmo em situações em que a natureza da força é bem distinta das forças de colisão e da força constante da gravidade. A energia cinética da partícula cresce até se estabilizar em, aproximadamente, 1 kJ aos 30 s de simulação, quando as forças de arrasto praticamente se igualam à força gravitacional. A energia mecânica, por outro lado, é continuamente dissipada em virtude da resistência do ar.

A importância desse último problema reside no fato de que o mesmo fenômeno físico é considerado na simulação acoplada entre o DEM e a mecânica de fluidos, a diferença consistindo apenas na expressão para o cálculo do coeficiente de arrasto que, no caso geral, depende do número de Reynolds, enquanto, neste trabalho, foi considerado constante.

Com isso, foram cumpridos os objetivos propostos para este trabalho:

- a) o Método de Elementos Discretos e sua fundamentação matemática foram

- apresentados e detalhados;
- b) foi implementada uma biblioteca para simulação DEM capaz de gerar simuladores a partir de modelos de interação de natureza qualquer entre partículas de geometria, tipo e propriedades arbitrários;
 - c) com base nos métodos expostos nas [capítulo 2 e 3](#), um simulador foi gerado com sucesso;
 - d) a corretude do programa foi atestada por meio da análise de resultados de simulação e de sua comparação com soluções analíticas.

5.3 Recomendações para Trabalhos Futuros

Para trabalhos futuros nessa linha de pesquisa, sugere-se:

- a) remover as restrições indicadas na [seção 4.1](#) da biblioteca. Essa tarefa pode ainda ser dividida em etapas:
 - permitir o uso de algoritmos de integração distintos, e mesmo da escolha de diferentes métodos de extração no algoritmo de Gear;
 - possibilitar a escolha de critérios de parada arbitrários;
 - acrescentar suporte a métodos de monitoramento de vizinha quaisquer;
 - generalizar o procedimento de entrada e saída de dados para permitir formatos de arquivo diferentes;
 - permitir que não apenas a posição e a orientação sejam submetidas à integração, mas também graus de liberdade quaisquer.
- b) melhorar o pós-processamento. Embora a linguagem Python forneça diversas ferramentas apropriadas para o processamento dos resultados das simulações, existem bibliotecas e programas especializados que o fazem de maneira muito mais eficiente. Por exemplo, sugere-se, aqui, o uso do Paraview ([AYACHIT, 2015; AHRENS; GEVECI; LAW, 2005](#)), que possui ferramentas de visualização bastante completas e suporte para scripts em Python;
- c) embora a classe template **Simulator** definida na biblioteca aceite tipos de entidades e de interações quaisquer, ainda pode ser trabalhoso de se implementarem as classes de interesse a serem inseridas no simulador. Propõe-se a implementação de tipos de partículas adicionais, como os superelipsoides, descritos em [Sampaio \(2017\)](#), as partículas poliédricas e as compostas, descritas por [Schwager e Pöschel \(2005\)](#);
- d) introduzir a definição de materiais. Até o momento, todas as propriedades físicas de interesse devem ser informadas manualmente nos arquivos de entrada. Entretanto esses valores podem ser obtidos se se souberem os materiais de que

são compostos os elementos da simulação. Para tanto, o conceito de *materiais* deve ser introduzido na biblioteca, e um banco de dados deve ser inserido;

- e) adicionar unidades de medida às propriedades. Na implementação corrente, o programa opera as variáveis independentemente de suas unidades de medida, as quais devem ser rastreadas manualmente pelo usuário;
- f) estudar o acoplamento entre DEM e CFD como indicado na [seção 3.7](#). Propõe-se, ainda, que se considere o uso do Método de Volumes Finitos Baseado em Elementos para a resolução do escoamento;
- g) incluir modelos de troca de calor como aquele descrito por [Simsek et al. \(2009\)](#).

Referências

- AHRENS, J.; GEVECI, B.; LAW, C. *ParaView: An End-User Tool for Large Data Visualization*. 1. ed. [S.l.], 2005.
- AI, J. et al. Assessment of rolling resistance models in discrete element simulations. *Powder Technology*, 2010.
- ALLEN, M. P.; TILDESLEY, D. J. *Computer Simulation of Liquids*. Oxford: Clarendon Press, 1987.
- AYACHIT, U. *The ParaView Guide: A Parallel Visualization Application*. [S.l.]: Science, 2015.
- BIĆANIĆ, N. Discrete element methods. In: STEIN, E.; BORST, R. de; HUGHES, T. J. R. (Ed.). *Encyclopedia of Computational Mechanics*. Glasgow, Scotland: John Wiley & Sons, Ltd., 2007. v. 1, cap. 11, p. 266–273.
- BREZINSKI, C.; ZAGLIA, M. R. *Extrapolation Methods: Theory and Practice*. [S.l.]: Elsevier Science, 2013. v. 2. (Studies in Computational Mathematics, v. 2).
- BRILLANTOV, N. V. et al. Model for collisions in granular gases. *Physical Review E*, v. 53, n. 5, p. 5382–5392, 1996.
- CHU, K. W.; YU, A. B. Numerical simulation of complex particle-fluid flows. *Powder Technology*, p. 104–114, 2008.
- COMELLI, R. C. *ParticleSimulator*. On-line: GitHub, 2017. Disponível em: <<https://github.com/ruancomelli/ParticleSimulator>>. Acesso em: 09 dez. 2017.
- CUNDALL, P. A. A computer model for simulating progressive large scale movements in blocky rock systems. In: *Proceedings of the Symposium on International Society of Rock Mechanics*. France: Society for Rock Mechanics, 1971. (II, 8).
- CUNDALL, P. A.; HART, R. D. Numerical modeling of discontinua. In: MUSTOE M. HENRIKSEN, H. P. H. G. W. (Ed.). *First U.S Conference on Discrete Element Methods (DEM)*. Golden: Colorado School of Mines, 1989.
- CUNDALL, P. A.; STRACK, O. D. L. A discrete numerical model for granular assemblies. *Géotechnique*, p. 47–65, 1979.
- DANG, H. K.; MEGUID, M. A. An efficient finite-discrete element method for quasi-static nonlinear soil-structure interaction problems. *INTERNATIONAL JOURNAL FOR NUMERICAL AND ANALYTICAL METHODS IN GEOMECHANICS*, Wiley Online Library, 2011.
- DEEN, N. G. et al. Review of discrete particle modeling of fluidized beds. *Chemical Engineering Science*, Elsevier Science, v. 62, p. 28–44, 2007.
- DONZÉ, F. V.; RICHEFEU, V.; MAGNIER, S.-A. Advances in discrete element method applied to soil, rock and concrete mechanics. *Electronic Journal of Geotechnical Engineering*, 2009.

- FOX, R. W.; PRITCHARD, P. J.; McDONALD, A. T. *Introdução à Mecânica dos Fluidos*. 7. ed. Rio de Janeiro (RJ): LTC, 2009.
- FRIES, L. et al. DEM–CFD modeling of a fluidized bed spray granulator. *Chemical Engineering Science*, Elsevier Science, v. 66, p. 2340–2355, 2011.
- GALLAS, J. A. C. et al. Molecular dynamics simulation of size segregation in three dimensions. *Journal of Statistical Physics*, v. 82, 1996.
- GALLAS, J. A. C.; HERRMANN, H. J.; SOKOLOWSKI, S. Convection cells in vibrating granular media. *Physical Review Letters*, v. 69, n. 9, 1992.
- GEAR, C. W. *Numerical Initial Value Problems in Ordinary Differential Equations*. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1971.
- GOMES, A. V. S. *Estudo do Acoplamento do Método dos Elementos Finitos e do Método de Elementos Discretos na Análise de Interação Solo-Estrutura*. Dissertação (Mestrado) — Universidade Estadual de Campinas, Campinas, 2014.
- HAFF, P. K.; WERNER, B. T. Computer simulation of the mechanical sorting of grains. *Powder Technology*, p. 239–245, 1986.
- HERTZ, H. Ueber die berührung fester elastischer körper. *Journal für die reine und angewandte Mathematik*, v. 92, p. 156–171, 1881.
- KARLSSON, B. *Beyond the C++ Standard Library: An Introduction to Boost*. Boston: Addison-Wesley, 2005.
- LOHMANN, N. *JSON for Modern C++*. On-line: GitHub, 2017. Disponível em: <<https://github.com/nlohmann/json>>. Acesso em: 09 dez. 2017.
- MACHADO, M. et al. Compliant contact force models in multibody dynamics: Evolution of the hertz contact theory. *Mechanism and Machine Theory*, Elsevier Science, v. 53, p. 99–121, 2012.
- MAIO, F. P. D.; RENZO, A. D.; TREVISAN, D. Comparison of heat transfer models in DEM-CFD simulations of fluidized beds with an immersed probe. *Powder Technology*, Elsevier Science, v. 193, p. 257–265, 2009.
- MALISKA, C. R. *Transferência de Calor e Mecânica dos Fluidos Computacional*. Rio de Janeiro, (RJ): LTC Editora, 2004.
- MARIGO, M.; STITT, E. H. Discrete element method (DEM) for industrial applications: Comments on calibration and validation for the modelling of cylindrical pellets. *KONA Powder and Particle Journal*, 2015.
- NAKASHIMA, H.; OIDA, A. Algorithm and implementation of soil–tire contact analysis code based on dynamic fe–de method. *Journal of Terramechanics*, Elsevier Science, v. 41, p. 127–137, 2004.
- OÑATE, E.; ROJEK, J. Combination of discrete element and finite element methods for dynamic analysis of geomechanics problems. *Computational methods in applied mechanics and engineering*, Elsevier Science, v. 193, p. 3087–3128, 2004.

- RANTANEN, J.; KHINAST, J. The future of pharmaceutical manufacturing sciences. *Journal of Pharmaceutical Sciences*, 2015.
- SAMPAIO, M. A. B. *O Método dos Elementos Discretos com Superelipsóides usando a Parametrização das Rotações de Rodrigues*. Tese (Doutorado) — Universidade de São Paulo, 2017.
- SCHWAGER, T.; PÖSCHEL, T. *Computational Granular Dynamics*. 1. ed. Berlin, Germany: Springer, 2005.
- SCHWAGER, T.; PÖSCHEL, T. Coefficient of restitution and linear dashpot model revisited. *Granular Matter*, p. 465–469, 2007.
- SCHWAGER, T.; PÖSCHEL, T. *Coefficient of Restitution for Viscoelastic Spheres: The Effect of Delayed Recovery*. 2008. arXiv:07081434.
- SIMSEK, E. et al. Numerical simulation of grate firing systems using a coupled CFD/discrete element method (DEM). *Powder Technology*, Elsevier Science, v. 193, p. 266–273, 2009.
- SPIVAK, M. *Calculus*. 3. ed. Houston, Texas: Publish or Perish, Inc., 1994.
- TSUJI, T.; KAWAGUCHI, T.; TANAKA, T. Discrete particle simulation of two-dimensional fluidized bed. *Powder Technology*, Elsevier Science, v. 77, p. 79–87, 1993.
- TSUJI, T.; YABUMOTO, K.; TANAKA, T. Spontaneous structures in three-dimensional bubbling gas-fluidized bed by parallel DEM–CFD coupling simulation. *Powder Technology*, Elsevier Science, v. 184, p. 132–140, 2008.
- VARGAS, W. L.; MCCARTHY, J. J. Conductivity of granular media with stagnant interstitial fluids via thermal particle dynamics simulation. *International Journal of Heat and Mass Transfer*, Elsevier Science, v. 45, p. 4847–4856, 2002.
- WALTON, O. R. Discrete element methods. In: JENKINS, J. T.; SATAKE, M. (Ed.). *Mechanics of Granular Materials: New Models and Constitutive Relations*. Amsterdam: Elsevier Science, 1983. p. 327–338.
- WITTENBURG, J. *Dynamics of Multibody Systems*. 2. ed. Berlin, Germany: Springer, 2008.

Apêndices

APÊNDICE A – Caso Geral do Algoritmo de Gear

No caso de partículas quaisquer, o algoritmo de Gear precisa levar em consideração as parametrizações para as orientações das partículas. O algoritmo inicia com a escolha da ordem de derivação k para a predição conforme a equação (2.27):

$$\begin{aligned}\mathbf{r}_{i,\text{pr}}^{[k]}(t_n) &= \underline{\mathbf{A}}_k(\Delta t_n) \cdot \mathbf{r}_i^{[k]}(t_{n-1}) \\ \boldsymbol{\varphi}_{i,\text{pr}}^{[k]}(t_n) &= \underline{\mathbf{A}}_k(\Delta t_n) \cdot \boldsymbol{\varphi}_i^{[k]}(t_{n-1})\end{aligned}, \quad \mathcal{P}_i \in \mathcal{P}.$$

O cálculo de forças e torques se dá de maneira idêntica ao caso para partículas esféricas, isto é,

$$\begin{aligned}\mathbf{F}_i &= \sum_{\mathcal{E}_j \in \mathcal{N}_i} \mathbf{F}_{ji}(\mathcal{P}_i, \mathcal{E}_j) + \mathbf{F}_i^{\text{ext}}(\mathcal{P}_i) \\ \mathbf{M}_i &= \sum_{\mathcal{E}_j \in \mathcal{N}_i} \mathbf{M}_{ji}(\mathcal{P}_i, \mathcal{E}_j) + \mathbf{M}_i^{\text{ext}}(\mathcal{P}_i)\end{aligned}, \quad \mathcal{P}_i \in \mathcal{P}.$$

As equações de movimento (2.4) e (2.9), então, são utilizadas para se corrigirem as derivadas de segunda ordem da posição e da orientação das partículas:

$$\begin{aligned}\ddot{\mathbf{r}}_{i,\text{corr}} &= \frac{1}{m_i} \cdot \mathbf{F}_i \\ \ddot{\boldsymbol{\varphi}}_{i,\text{corr}} &= f^\varphi(\mathbf{M}_i, \boldsymbol{\varphi}_{i,\text{pr}}, \dot{\boldsymbol{\varphi}}_{i,\text{pr}})\end{aligned}, \quad \mathcal{P}_i \in \mathcal{P},$$

sendo que f^φ depende da parametrização escolhida para a orientação.

Definem-se, então, os erros

$$\begin{aligned}\Delta \ddot{\mathbf{r}}_i(t_n) &= \ddot{\mathbf{r}}_{i,\text{corr}}(t_n) - \ddot{\mathbf{r}}_{i,\text{pr}}(t_n) \\ \Delta \ddot{\boldsymbol{\varphi}}_i(t_n) &= \ddot{\boldsymbol{\varphi}}_{i,\text{corr}}(t_n) - \ddot{\boldsymbol{\varphi}}_{i,\text{pr}}(t_n)\end{aligned}, \quad \mathcal{P}_i \in \mathcal{P}.$$

Por fim, resta executar a correção nos vetores de derivadas. Tanto para a translação quanto para a rotação, a equação diferencial é de ordem $p = 2$. Com isso, aplica-se a equação (2.32) para se escrever

$$\begin{aligned}\mathbf{r}_{i,\text{corr}}^{[k]}(t_n) &= \mathbf{r}_{i,\text{pr}}^{[k]}(t_n) + \mathbf{c}_{2,k} \cdot \frac{\Delta t_n^2}{2} \Delta \ddot{\mathbf{r}}_i(t_n) \\ \boldsymbol{\varphi}_{i,\text{corr}}^{[k]}(t_n) &= \boldsymbol{\varphi}_{i,\text{pr}}^{[k]}(t_n) + \mathbf{c}_{2,k} \cdot \frac{\Delta t_n^2}{2} \Delta \ddot{\boldsymbol{\varphi}}_i(t_n)\end{aligned}, \quad \mathcal{P}_i \in \mathcal{P}.$$

Dessa forma, estão calculadas a posição, a orientação e as suas derivadas, e o instante de tempo t_n é considerado solucionado.

APÊNDICE B – Arquivo de Entrada para o Problema do Lançamento Oblíquo

Listagem B.1 – Arquivo de entrada para o problema do lançamento oblíquo.

```

1 { // Parâmetros do simulador:
2   "InitialInstant": 0.0,
3   "TimeStep": 1e-3,
4   "FinalInstant": 3,
5   "StepsForStoring": 50, // Número de passos de tempo para armazenamento
6   "StoragesForWriting": 1, // Número de passos armazenados para a escrita em ...
... arquivo
7   "MainOutputFolder": "output", // Diretório principal de exportação
8   "ParticleOutputFolder": "particles", // Diretório para exportação de partículas
9   "BoundaryOutputFolder": "boundaries", // Diretório para exportação de ...
... elementos de contorno
10  "IntegrationAlgorithm": "Gear", // Algoritmo de integração das equações
11
12 // Definição das interações
13  "Interactions": "GravityForce", // A única interação que ocorre é a da gravidade
14
15 // Definição das entidades
16  "Particles": { // Definição das partículas
17    "SphericalParticle": { // Tipo de partículas esféricas
18      "Name": "Particle",
19      "Color": "Red",
20      "TaylorOrder": 7, // Ordem de interpolação
21      "Mass": 1,
22      "Radius": 0.03,
23      "MomentOfInertia": 3.6e-4,
24      "Position": [0.0, 0.0, 0.0], // Vetor posição inicial
25      "Velocity": [20.0, 10.0, 0.0], // Vetor velocidade inicial
26      "Acceleration": [0.0, -9.81, 0.0] // Vetor aceleração inicial
27    }
28  },
29  "Boundaries": { // Definição de outras entidades
30    "GravityField": { // Campo gravitacional
31      "Name": "Gravity",
32      "Gravity": [0.0, -9.81, 0.0]
33    }
34  }
35 }
```
