Marcelo Caniato Renhe

Modelagem e simulação física de uma aeronave

Juiz de Fora

12/10/2006 (2006)

Marcelo Caniato Renhe

Modelagem e simulação física de uma aeronave

Orientador:
Marcelo Bernardes Vieira

Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Juiz de Fora

12/10/2006 (2006)

Monografia submetida ao corpo docente do Instituto de Ciências Exatas da Universidade Federal de Juiz de Fora como parte integrante dos requisitos necessários para obtenção do grau de bacharel em Ciência da Computação

Marcelo Bernardes Vieira, D. Sc.
Orientador

Rubens de Oliveira, D. Sc.

Raul Fonseca Neto, D. Sc.

Sumário

1 Introdução

Quando falamos em processamento gráfico, não precisamos nos preocupar com os custos advindos da pipeline gráfica, uma vez que o trabalho gasto para realizar as operações da pipeline está concentrado nas GPU's (graphics processing units) das placas de vídeo atuais. O mesmo, entretanto, não se pode dizer do suporte de hardware para as operações de simulação física. Ainda não existe grande suporte nesse sentido, sendo que a primeira PPU (physics processing unit) foi lançada há pouco tempo, em 2005, pela AGEIA. O microprocessador, chamado de PhysX, implementa cálculos para dinâmica de corpos rígidos, detecção de colisão e dinâmica de fluidos, entre outros. Essa idéia veio evoluindo e a nova geração de placas de vídeo está vindo equipada com essas unidades para agilizar o processamento físico.

Existem diversas aplicações que justificam o estudo de métodos de modelagem computacional de processos e interações no campo da física. Dentre elas, podemos citar as simulações militares, nas quais a busca por realismo e precisão é de vital importância para que se possa obter resultados que sirvam de objeto para a realização de estudos que permitam prever as adversidades de uma operação militar e minimizar as possibilidades de fracasso. Outra aplicação viável se encontra na criação de ambientes de estudo de física, para que estudantes possam visualizar fenômenos que não possam ser demonstrados em laboratórios convencionais de física. Temos ainda aplicações em física de partículas, modelagem de interações moleculares, simulações aeroespaciais, projetos de engenharia e jogos eletrônicos, sendo que o último possui um apelo cada vez maior com relação ao realismo.

O objetivo **primário** desta monografia é estudar métodos de modelagem de física de corpos rígidos. Inicialmente é discutida toda a base matemática e física sobre a qual se fundamenta o modelo matemático, que é usado para a criação de um modelo computacional capaz de representar a dinâmica de corpos.

A partir deste objetivo primário, o modelo computacional e todos os conceitos apresentados serão usados para a construção de um motor físico capaz de simular a física de

corpos rígidos simples e corpos rígidos agregados, que são composições de vários corpos mais simples. O intuito é o de construir um motor independente de qualquer pacote gráfico, de forma que ele possa ser facilmente adaptado e estendido.

Por fim, um outro objetivo é o de utilizar o motor físico desenvolvido para a construção de um modelo computacional para a representação e simulação de uma aeronave genérica. Será explicado o funcionamento básico das aeronaves, juntamente com a definição do modelo para a simulação.

1.1 Notação

A notação usada neste trabalho é a seguinte:

- o **negrito** é usado para representar vetores: $\mathbf{v}$
- o *itálico* é usado para representar escalares: s
- o ponto (.) em cima de uma variável significa diferenciação com relação ao tempo, ou seja:

$$\dot{x} = \mathrm{d}x/\mathrm{d}t$$

1.2 Propedêutica

Este trabalho está estruturado da seguinte forma: inicialmente, é apresentado um modelo matemático para a solução do problema da modelagem física. Em seguida, é apresentado o modelo computacional, que serve de base para a construção do motor físico. Por fim, é elaborado o modelo de uma aeronave, utilizando o motor construído.

Para entender o modelo computacional, é necessário ler o Capítulo 2, principalmente a seção que trata das propriedades de massa dos corpos rígidos, pois sem elas não tem como simulá-los.

O entendimento do desenvolvimento do motor físico depende da leitura do Capítulo 3. A maioria dos conceitos discutidos naquele capítulo são utilizados no desenvolvimento do modelo do motor.

O capítulo referente à simulação de aeronaves possui duas partes. A primeira, que explica o funcionamento de um avião, pode ser compreendido independentemente dos outros capítulos desse trabalho. Já a segunda exige que a seqüência dos capítulos seja

seguida para ter um completo entendimento do assunto, sendo necessário, principalmente, ter lido o Capítulo 4.

2 Modelo Matemático

Antes de tecer qualquer comentário sobre os aspectos computacionais da modelagem física, é indispensável compreender primeiramente a base matemática que sustenta tudo isso. A idéia deste capítulo é apresentar alguns dos conceitos fundamentais da cinemática e da mecânica clássica de Newton e duas abordagens distintas para a modelagem matemática do movimento de corpos rígidos. Os assuntos discutidos neste capítulo servirão de base para o capítulo seguinte, no qual será tratada a questão da criação de um modelo computacional capaz de representar e simular a física de corpos.

2.1 Conceitos físicos

A primeira coisa que deve ficar clara é a diferença existente entre corpos rígidos discretos e contínuos. Basicamente, existem três tipos de corpos rígidos: partículas, sistemas de partículas e contínuos de massa.

Por definição, corpo rígido é todo corpo que não sofre deformações quando sob a ação de forças externas, ou seja, a distância entre as partículas que o constituem permanece sempre a mesma. Dessa forma, não existe translação ou rotação entre as partículas constituintes. Trata-se de um corpo ideal, visto que na prática isto não acontece. Na realidade, todo corpo é deformável, mesmo que essa deformação seja invisível ao olho humano.

O tipo mais simples de corpo rígido é uma **partícula**, cuja massa é finita, porém suas dimensões podem ser desprezadas, uma vez que são tão pequenas que não influem na análise do problema em questão. Quando temos duas ou mais partículas em conjunto, damos o nome de **sistema de partículas**.

Os dois casos anteriores são exemplos de material discreto, ou seja, existe um número finito de partículas. Normalmente, os problemas da vida real envolvem material contínuo, isto é, corpos que possuem uma quantidade infinita de partículas componentes. Este tipo

de corpo é chamado de **contínuo de massa**. Em materiais discretos, uma grandeza física qualquer pode ser obtida pelo somatório das grandezas das partículas que compõem o sistema. Já no caso de materiais contínuos, um simples somatório já não se aplica, sendo necessário integrar sobre a região do corpo desejada.

2.1.1 Cinemática

A cinemática é uma área da física que estuda o movimento de corpos na ausência de forças externas. Esta área se preocupa apenas com a determinação das componentes espaciais e temporais do movimento, sem considerar as forças que causam o movimento. Estas componentes estudadas pela cinemática são a posição, a velocidade e a aceleração. As três podem ser definidas através das equações abaixo, para uma partícula em duas dimensões:

$$\mathbf{r}(t) = x(t)\mathbf{i} + y(t)\mathbf{j}$$

$$\mathbf{v}(t) = \dot{\mathbf{r}}$$

$$\mathbf{a}(t) = \dot{\mathbf{v}} = \ddot{\mathbf{r}}$$

No caso acima, as componentes do movimento foram expressadas em termos dos eixos canônicos **i** e **j**. Os vetores posição, velocidade e aceleração são representados, respectivamente, por **r**, **v** e **a**. É possível também definir um sistema de referência móvel, que varia com a posição da partícula. Os eixos coordenados desde sistema seriam um vetor tangente à curva na posição corrente da partícula e um vetor normal, resultante de uma rotação de 90° do vetor tangente no sentido anti-horário. A Figura 1 ilustra este sistema de referência móvel.

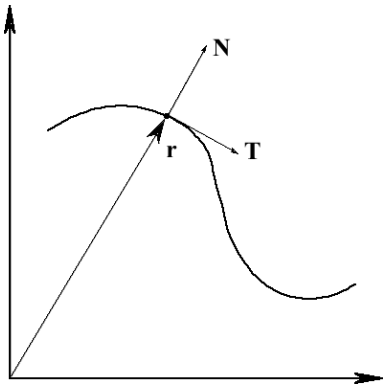


Figura 1: Sistema de referência $\{r(t); T(t), N(t)\}$ varia com o tempo

Para o caso em que a partícula está se movendo em três dimensões, basta apenas

adicionar uma coordenada à equação da posição. No caso de se definir um sistema de referência dependente da posição da partícula como no caso 2D, é necessário definir um novo vetor, chamado de vetor **binormal**, resultante do produto vetorial do vetor tangente pelo vetor normal.

Quando uma partícula se move em torno de um eixo, seja ele fixo ou não, deve-se considerar as componentes angulares do movimento. A Figura 2 ilustra o movimento de uma partícula em torno de um eixo fixo.

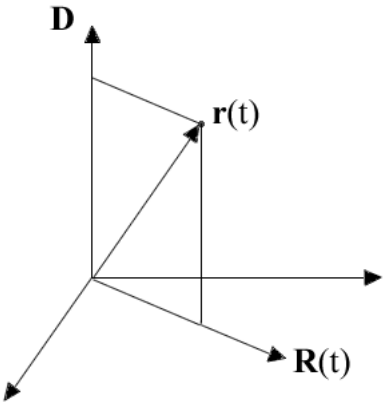


Figura 2: Movimento de uma partícula em torno de um eixo

Analogamente ao movimento linear, as componentes do movimento angular são dadas pelas equações abaixo, em que σ é o vetor posição, ω é o vetor velocidade angular, α é o vetor aceleração angular e θ é o ângulo de rotação.

$$\sigma(t) = \dot{\theta}$$

$$\omega(t) = \dot{\sigma}$$

$$\alpha(t) = \dot{\omega} = \ddot{\sigma}$$

Existe uma relação entre o movimento linear de um corpo e o seu movimento angular. Essa relação é definida pelas seguintes equações:

$$\mathbf{r}(t) = r_0\mathbf{R}(t) + h_0\mathbf{D}$$

$$\mathbf{v}(t) = \omega \times \mathbf{r}$$

$$\mathbf{a}(t) = -\sigma^2\mathbf{r} + \alpha \times \mathbf{r}$$

A posição da partícula em cada instante de tempo é determinada pela rotação de

$\mathbf{r}(t)$ em torno do eixo $\mathbf{D}$. Supondo um vetor $\mathbf{u} = (u_1, u_2, u_3)$ qualquer, podemos definir a seguinte matriz anti-simétrica:

$$\text{Skew}(\mathbf{u}) = \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix} \tag{2.1}$$

Essa matriz possui a propriedade de que $\text{Skew}(\mathbf{u})\mathbf{r} = \mathbf{u} \times \mathbf{r}$, para qualquer vetor $\mathbf{r}$. Sendo $R(t)$ a matriz de rotação correspondente à rotação em torno do eixo, e $\mathbf{w}(t)$ a velocidade angular da partícula, chega-se à seguinte equação:

$$\dot{R}(t) = \text{Skew}(\mathbf{w}(t))R(t) \tag{2.2}$$

Esta equação define a taxa de variação na rotação, em função da atual rotação da partícula e de sua velocidade angular. A demonstração da validade dessa equação pode ser encontrada em (GOLDSTEIN; POOLE; SAFKO, 2002). Podemos ainda usar essa equação e estendê-la para uma partícula girando em torno de um eixo móvel. Para isso, basta considerar o vetor $\mathbf{D}$ como um vetor variável no tempo e derivar as novas equações.

Todas as equações mostradas até agora se aplicam perfeitamente a partículas, porém para corpos rígidos compostos por mais de uma partícula, é preciso considerar algumas questões, como a diferença entre coordenadas do mundo e coordenadas do corpo.

As coordenadas do mundo são coordenadas expressas em termos dos eixos canônicos do espaço considerado. Elas variam de acordo com a posição do corpo, enquanto os eixos coordenados permanecem fixos. Já as coordenadas do corpo são fixas para cada objeto, visto que o que muda no tempo é a posição e a orientação dos eixos do sistema de coordenadas local, que tem origem no centro de massa do corpo.

2.1.2 Forças

Foi apresentado na seção anterior o movimento de corpos, porém sem considerar as forças que agem sobre eles. Entretanto, na prática, as forças estão presentes a todo momento e precisam ser levadas em consideração na construção de um modelo matemático que represente com exatidão a realidade.

As forças são grandezas capazes de provocar uma mudança no estado atual de um corpo. A força também pode ser interpretada como uma variação do momento linear de um corpo. Elas são regidas pelas já conhecidas Leis de Newton. A primeira e a terceira

leis - a lei de inércia e a lei de ação e reação, respectivamente - não são de grande relevância para os propósitos deste trabalho. A lei que realmente interessa para a criação do modelo matemático é a segunda lei de Newton. Esta lei define a famosa equação $F = ma$, que garante que a força é diretamente proporcional à aceleração do corpo sobre o qual ela age e é o resultado do produto dessa aceleração pela massa do corpo. É através desta equação que obteremos as equações do movimento para os corpos rígidos considerados.

Existem vários tipos de forças (BARAFF, 1997a). Temos as forças unárias, que são forças que agem de forma independente sobre cada corpo, como é o caso da gravidade e do atrito. A ação da gravidade sobre um corpo A, por exemplo, não influi na ação dessa mesma força sobre dois corpos B e C. As forças elásticas são consideradas forças n-árias, pois a ação delas depende de um conjunto fixo de corpos. No caso das forças elásticas, este conjunto normalmente é composto por dois corpos: um em uma extremidade da mola e o outro na outra extremidade. Por fim, temos as forças de interação espacial, que agem sobre todos os corpos simultaneamente dentro da área de influência da força. Um exemplo desse tipo de força é a força de atração ou repulsão entre os corpos.

Outro conceito que é importante ter em mente é o de força conservativa. Forças conservativas são aquelas que independem da trajetória. Ou seja, o trabalho realizado pelo objeto sobre o qual age a força não depende da trajetória percorrida pelo objeto. É chamada de conservativa, pois se o objeto for movido de um ponto A para um ponto B, por exemplo, e depois retornar ao ponto A, não terá ocorrido perda de energia mecânica. Já as forças dissipativas provocam perda de energia. O principal exemplo desse tipo de força é o atrito, que faz com que a energia mecânica se transforme em calor.

2.1.3 Propriedades de massa

Os conceitos a seguir são de suma importância, visto que eles são essenciais para a simulação física. Existem três propriedades de um corpo que em conjunto são chamadas de propriedades de massa: a massa propriamente dita, o centro de massa e o momento de inércia (BOURG, 2002).

A massa de um corpo pode ser vista como uma medida da quantidade de matéria presente no corpo. Uma outra possibilidade é considerar a massa como uma medida da resistência de um corpo ao movimento ou a uma mudança em seu movimento, o que faz mais sentido quando pensamos em mecânica.

O centro de massa, também chamado de centro de gravidade, é o ponto do corpo ao

redor do qual sua massa está igualmente distribuída. Qualquer força agindo nesse ponto não será capaz de colocar o corpo em rotação. Por isso, as partículas não possuem rotação em torno de si mesmas, visto que a partícula corresponde a um corpo rígido composto apenas por um único ponto, que é o seu centro de massa.

O cálculo do centro de massa pode ser feito através do seguinte procedimento: divide-se o corpo em um número infinito de elementos de massa, soma-se os produtos da massa pela posição de cada elemento e divide-se o resultado pela soma das massas.

$$\frac{\sum_{i=1}^{p} m_i x_i}{\sum_{i=1}^{p} m_i}$$

Esta fórmula se aplica a um sistema de partículas com p partículas em uma dimensão. Supondo que o sistema de partículas esteja em duas ou três dimensões, a única modificação seria no número de coordenadas na multiplicação da massa pelo ponto correspondente à posição do elemento de massa.

O somatório foi possível no caso anterior pois a quantidade de elementos de massa era finita. Caso o corpo seja um material contínuo, o número de elementos de massa obtido pela divisão é infinito. Logo, deve-se considerar a densidade de massa em cada ponto e integrar sobre uma curva, uma superfície ou um volume, dependendo do número de dimensões consideradas no problema. Suponha uma função $\delta(x)$ que retorna a densidade de massa do elemento na posição x. O cálculo do centro de massa ficaria da seguinte forma:

$$\frac{\int_R x\delta(x)\mathrm{d}x}{\int_R \delta(x)\mathrm{d}x}$$

onde a região de integração R pode ser uma curva, uma superfície ou um volume.

A última propriedade de massa, chamada de momento de inércia, é uma medida da distribuição radial da massa de um corpo com relação a um eixo de rotação determinado. O momento de inércia pode ser entendido também como uma medida da resistência de um corpo ao movimento rotacional.

O momento de inércia com relação ao centro de massa do corpo pode ser calculado pela equação abaixo:

$$I = I_0 - \bar{x}^2$$

em que I_0 é o momento de inércia com relação à origem e é definido, em uma dimensão, por:

$$I_0 = \sum_{i=1}^{p} m_i x_i^2$$

Em duas dimensões, x^2 seria substituído por $(x^2 + y^2)$. Porém, na marioria dos casos, estaremos lidando com três dimensões. Neste caso, o momento de inércia será definido relativamente a uma reta, ao contrário dos casos anteriores. Essa decisão faz sentido uma vez que os pontos de um corpo rígido se movem em conjunto. Se um ponto executa uma rotação em torno de uma reta, então todos os outros pontos do corpo também entram em rotação.

Dada uma reta parametrizada pela equação $O + t\mathbf{D}$, em que O é a origem do sistema de coordenadas e $\mathbf{D} = (d_1, d_2, d_3)$, o momento de inércia em relação a essa reta será:

$$I_L = d_1^2 I_{xx} + d_2^2 I_{yy} + d_3^2 I_{zz} - 2d_1 d_2 I_{xy} - 2d_1 d_3 I_{xz} - 2d_2 d_3 I_{yz} \tag{2.3}$$

As quantidades I_{xx}, I_{yy} e I_{zz} são os momentos de inércia com relação aos eixos coordenados. Já as quantidades I_{xy}, I_{xz} e I_{yz} são os chamados produtos da inércia. Estas seis quantidades são dadas pelas seguintes equações:

$$I_{xx} = \sum_{i=1}^{p} m_i(y_i^2 + z_i^2) \ , \ I_{yy} = \sum_{i=1}^{p} m_i(x_i^2 + z_i^2) \ , \ I_{zz} = \sum_{i=1}^{p} m_i(x_i^2 + y_i^2)$$

$$I_{xy} = \sum_{i=1}^{p} m_i x_i y_i \ , \ I_{xz} = \sum_{i=1}^{p} m_i x_i z_i \ , \ I_{yz} = \sum_{i=1}^{p} m_i y_i z_i$$

Escrevendo a equação 2.3 em notação matricial, temos:

$$I_L = \mathbf{D}^T J \mathbf{D}$$

A matriz J é uma matriz simétrica formada pelos momentos de inércia em relação aos eixos coordenados e pelos produtos da inércia. Esta matriz é de extrema importância e é conhecida como **tensor de inércia** ou **matriz de massa**. Um tensor é uma grandeza cuja magnitude pode variar de acordo com a direção. Materiais que possuem propriedades definidas por tensores são chamados de anisotrópicos. A matriz de massa é considerada um tensor de ordem 2, visto que é representado por uma matriz de duas dimensões.

$$J = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xx} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix} \tag{2.4}$$

Para o caso de materiais contínuos, mais uma vez, basta substituir o somatório por uma integração sobre a região ocupada pela massa e integrar com relação a um elemento infinitesimal de massa dm. Para poliedros convexos, cuja geometria é mais complexa,

Mirtich (1996) apresenta um método para o cálculo do centro de massa e do tensor de inércia, que efetua reduções de integrais de volume a integrais de linha para realizar a computação, dado um conjunto de vértices que descrevem o poliedro.

2.1.4 Energia

Os conceitos de trabalho e energia têm grande importância para a dinâmica de Lagrange, que será discutida na seção seguinte. O trabalho nada mais é do que uma grandeza escalar que mede a energia transferida pela aplicação de uma força ao longo de um deslocamento. Existem dois tipos de energia: a cinética e a potencial. A energia cinética está associada ao movimento de um corpo. É definida como a quantidade de trabalho realizado para que um objeto saia do repouso e adquira uma velocidade v.

Agora considere dois pontos distintos. Um é conhecido como sistema de referência e o outro como sistema geral. Através da aplicação de uma força no sistema geral, tem-se que a energia potencial corresponde ao trabalho realizado para transferir o sistema geral para o sistema de referência.

Nos casos em que o trabalho realizado independe da trajetória do corpo, dizemos que a força que produz o trabalho é uma força **conservativa**. Uma força conservativa possui a propriedade $F = -\nabla V$. Para saber se uma força é conservativa, basta realizar um teste de exatidão. Se ela for conservativa, as três equações abaixo serão satisfeitas:

$$\frac{\partial F_1}{\partial y} = \frac{\partial F_2}{\partial x} \ , \ \frac{\partial F_1}{\partial z} = \frac{\partial F_3}{\partial x} \text{ e } \frac{\partial F_2}{\partial z} = \frac{\partial F_3}{\partial y}$$

2.2 Equações do movimento

As equações do movimento são expressões matemáticas que definem o movimento de um corpo no decorrer do tempo, dadas as forças que agem sobre ele em cada instante. Com elas, é possível definir posição, velocidade e aceleração para cada instante de tempo, definindo dessa forma a trajetória do corpo.

Para isso, serão discutidas aqui duas abordagens existentes. A primeira abordagem é a mais utilizada, baseada na dinâmica clássica de Newton, que utiliza a fórmula da segunda lei de Newton para determinar o movimento. Essa abordagem, embora muito utilizada, tem suas desvantagens, que motivam a escolha de uma outra forma de obter as equações do movimento. Essa outra forma é conhecida como dinâmica lagrangiana e possui certas vantagens que podem tornar mais interessante a sua utilização do que o uso

da dinâmica newtoniana.

2.2.1 Dinâmica de Newton

A dinâmica, ou cinética, é o estudo do movimento de corpos quando estes estão na presença de forças externas. Na dinâmica newtoniana, estas forças são todas incluídas na equação $F = ma$. Após analisar a situação-problema e determinar todas as forças agindo sobre o corpo, basta integrar a equação com relação ao tempo para obter a velocidade e a posição do corpo. Isso faz sentido, uma vez que $\mathbf{a} = \dot{\mathbf{v}}$ e $\mathbf{v} = \dot{\mathbf{x}}$.

Suponha um corpo em queda livre. A princípio, a única força que age sobre ele é a da gravidade. Podemos considerar também a resistência do ar. Desprezando a ação de qualquer outra força externa, como, por exemplo, o vento, o problema simplifica bastante. Neste caso, deduzir as equações do movimento que determinam o movimento do corpo nas direções dos eixos coordenados é trivial. Contudo, a maioria dos problemas não apresentam essa simplicidade.

Para situações mais elaboradas, em que existem várias forças agindo sobre o corpo, restringindo o seu movimento de maneiras diferentes, a simplificidade da equação da segunda lei de Newton deixa de existir. Conseqüentemente, as equações do movimento obtidas pela dinâmica newtoniana tornam-se extremamente complexas. Computacionalmente, essa complexidade se traduz em maior tempo de processamento e maior propensão a erros numéricos. Como a maioria das aplicações que envolvem simulação física são aplicações de tempo real, estas conseqüências podem ser drásticas.

2.2.2 Dinâmica de Lagrange

Diferentemente da newtoniana, essa abordagem é baseada no conceito de energia. A partir da segunda lei de Newton e da equação do trabalho, chega-se na seguinte equação, chamada de equação de D'Alembert:

$$m\ddot{\mathbf{x}} \cdot \mathrm{d}\mathbf{x} = \mathbf{F} \cdot \mathrm{d}\mathbf{x} \tag{2.5}$$

onde o lado esquerdo da equação representa uma variação da energia cinética e o lado direito representa o trabalho realizado.

Uma das vantagens de se usar Lagrange é que as equações lagrangianas do movimento conseguem separar forças restritivas das demais forças. Com Newton, todas as forças

tinham que obrigatoriamente figurar no termo F da equação. Com Lagrange, isso não é mais necessário. As forças de restrição podem ser tratadas separadamente. Isso leva a equações mais simples, permitindo criar modelos mais robustos e com menos custo computacional.

Existe, porém, uma desvantagem. A dinâmica de Lagrange é específica para cada aplicação. Ou seja, não é possível usar o mesmo modelo para várias aplicações. É claro que aplicações semelhantes podem se aproveitar dos modelos já criados, mas, em geral, isso é muito difícil. Isto eleva os custos de desenvolvimento, o que pode atrapalhar na hora de decidir entre uma ou outra abordagem. As questões referentes à decisão de quando se usar um ou outro modelo serão discutidas mais adiante.

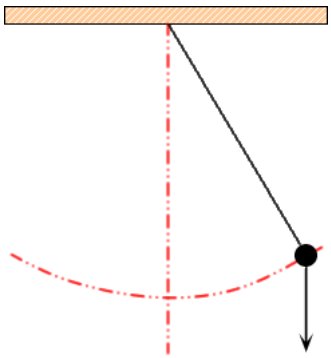


Figura 3: Pêndulo simples

Para ilustrar a situação, suponha um pêndulo simples, composto por uma pequena esfera e um fio que a liga a um suporte, como ilustra a Figura 3. A esfera sofrerá a ação da gravidade, porém, ela também sofre uma força de tensão do fio, que restringe o movimento dela a uma curva específica. Se a curva for parametrizada por uma variável q qualquer, a restrição da esfera se manter sobre a curva $x(q)$ pode ser definida como abaixo, através da aplicação da equação de D'Alembert:

$$m\ddot{\mathbf{x}} \cdot \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}q} = \mathbf{F} \cdot \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}q}$$

Repare que o vetor $\mathrm{d}\mathbf{x}/\mathrm{d}q$ é tangente à curva no ponto x. À medida que o corpo se move, o vetor tangente vai determinando a curva à qual a esfera está restrita, em intervalos infinitesimais de tempo. Seja $T = m\ddot{\mathbf{x}}/2$ a energia cinética do sistema, podemos, a partir da equação acima, chegar à forma geral da equação lagrangiana do movimento:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial T}{\partial \dot{q}}\right) - \frac{\partial T}{\partial q} = F_q \tag{2.6}$$

onde $F_q = \mathbf{F} \cdot \mathrm{d}\mathbf{x}/\mathrm{d}q$ é chamada de **força genérica**, apesar de não ser uma força no sentido físico, uma vez que suas unidades diferem das unidades de força. A grande diferença desta equação para a newtoniana é que nesta o termo F_q inclui apenas as forças externas. As forças de restrição estão todas representadas no lado esquerdo da equação.

Normalmente, tem-se uma equação lagrangiana do movimento para cada grau de liberdade presente no problema. Também é perfeitamente possível aplicar as equações de Lagrange a parâmetros adicionais aos já existentes, mesmo que eles não variem livremente no tempo. A única diferença, neste caso, é que as forças de restrição não farão parte do lado esquerdo da equação, e sim do lado direito, somando-se às forças gerais.

Como é uma extensão natural da segunda lei de Newton, a dinâmica de Lagrange se aplica a todos os problemas que podem ser resolvidos com a dinâmica newtoniana. Contudo, alguns problemas são simples demais para que haja necessidade de se usar a dinâmica lagrangiana.

Para o caso de forças conservativas, pode-se fazer uma pequena modificação na equação do movimento. Lembrando que, para forças conservativas, $F = -\nabla V$, a componente F_q pode ser substituída pela derivada parcial com relação ao tempo da energia potencial. Dessa forma, a equação fica assim:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial T}{\partial \dot{q}}\right) - \frac{\partial T}{\partial q} = -\frac{\partial V}{\partial q}$$

Pode-se ainda definir uma função $L = T - V$, chamada de **função lagrangiana**, e reescrever a equação acima usando essa função. Neste caso, chegamos à equação abaixo para forças conservativas:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} = 0 \tag{2.7}$$

Como o propósito desse trabalho é simular uma aeronave em movimento irrestrito, ou seja, isento de restrições, a dinâmica de Lagrange não será utilizada no desenvolvimento do motor físico no Capítulo 4. Porém, o motor possuirá a liberdade para a inserção da dinâmica lagrangiana em trabalhos futuros, caso haja o interesse.

3 Modelo Computacional

Os conceitos físicos e matemáticos expostos no capítulo anterior são a base para a definição do modelo computacional. Neste capítulo, serão apresentadas algumas formas de se representar computacionalmente o modelo matemático apresentado. Serão discutidas também as dificuldades que surgem nesse processo, as simplificações que precisam ser feitas, o que são motores físicos e o papel desses na simulação física.

3.1 Movimento irrestrito

Movimento irrestrito é aquele em que os corpos movem-se livremente no ambiente em que se encontram, sem nenhum tipo de colisão ou qualquer restrição ao seu movimento. Tudo que existe são forças externas agindo sobre os corpos, provocando uma aceleração em cada um deles. Dado um movimento com essas características, torna-se suficiente a utilização da dinâmica de Newton para a determinação das equações do movimento.

3.1.1 Partículas

Sabe-se, da segunda lei de Newton, que $\mathbf{F}(t) = m\mathbf{a} = m\ddot{\mathbf{x}}$. Contudo, os métodos numéricos para solução de equações diferenciais normalmente lidam com sistemas de equações de primeira ordem. Para contornar isto, define-se um vetor $\mathbf{S}(t) = [\mathbf{x}\ \mathbf{v}]^T$, chamado de **vetor de estados**. Podemos, então, definir o sistema de equações diferenciais da seguinte forma (BARAFF, 1997a):

$$\frac{\mathrm{d}\mathbf{S}}{\mathrm{d}t} = \begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{v}} \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ \frac{\mathbf{F}}{m} \end{bmatrix} \tag{3.1}$$

Basicamente, a simulação física consistirá em atualizar este vetor de estados a cada instante de tempo. Se houver a necessidade de se representar um sistema de n partículas, basta acrescentar ao vetor de estados os vetores posição e velocidade de cada uma das n

partículas.

3.1.2 Corpos sólidos

O vetor de estados apresentado acima é suficiente para representar partículas, porém, normalmente trabalha-se com corpos rígidos que não são apenas formados por um único ponto. Para estes corpos, usamos o centro de massa como sendo o vetor posição do corpo e adicionamos as informações referentes à orientação do corpo através de uma matriz de rotação. Um corpo com essas características pode sofrer não só translações, mas também rotações. Dessa forma, o vetor de estados é alterado para incluir o momento linear e o momento angular do corpo. A velocidade linear não é mais necessária, visto que ela pode ser obtida através do momento linear. O mesmo vale para a velocidade angular.

Para manter a simulação consistente, o tensor de inércia e o seu inverso devem ser recalculados a cada intervalo de tempo. Entretanto, isso é um processo muito custoso. Para contornar este problema, pode-se utilizar a seguinte relação:

$$J(t) = R(t)J_{corpo}R(t)^T \tag{3.2}$$

A derivação da fórmula acima pode ser obtida em (EBERLY, 2004). Como o tensor de inércia em coordenadas do corpo é fixo em relação ao tempo, o cálculo do tensor em coordenadas do mundo se resume a multiplicar as matrizes da expressão acima, dispensando a integração. O mesmo vale para o inverso do tensor.

3.1.3 Quaternions

Outra alternativa que torna a simulação mais rápida e robusta é o uso de quaternions no lugar de matrizes de rotação. Quaternions são vetores de dimensão 4, formulados por William Rowan Hamilton (HAMILTON, 1866). Eles fazem parte de uma classe mais geral de números hipercomplexos. Possuem a característica de que a multiplicação não é comutativa, apenas associativa.

Analogamente aos números complexos, podemos representá-los como a soma de partes reais e imaginárias. De fato, os quaternions são uma generalização dos números complexos, obtida pela adição dos elementos i, j e k aos números reais, tais que:

$$i^2 = j^2 = k^2 = ijk = -1$$

Todo quaternion é uma combinação linear dos quaternions de base 1 com i, j e k. Pode ser interpretado também como a combinação de um escalar e um vetor. A vantagem de utilizá-los é que uma simples soma de quaternions equivale a uma rotação em 3D. Isso torna a computação da orientação dos corpos muito mais rápida e menos propensa a erros numéricos.

Erros numéricos são inevitáveis quando se realiza integrações numéricas, de forma que torna-se necessário realizar uma correção na saída do solucionador de equações diferenciais. Essa correção é realizada periodicamente, em um intervalo de tempo definido na simulação. Matrizes de rotação podem ser corrigidas através da ortonormalização de Gram-Schmidt, que gera vetores ortonormais que se tornam as colunas da matriz de rotação. Seja $R(t) = [\mathbf{u}_0\ \mathbf{u}_1\ \mathbf{u}_2]$. A ortonormalização será dada por:

$$\mathbf{u}_0 = \frac{\hat{\mathbf{u}_0}}{|\hat{\mathbf{u}_0}|}\ ,\ \mathbf{u}_1 = \frac{\hat{\mathbf{u}_1} - (\hat{\mathbf{u}}_1 \cdot \mathbf{u}_0)\mathbf{u}_0}{|\hat{\mathbf{u}_1} - (\hat{\mathbf{u}}_1 \cdot \mathbf{u}_0)\mathbf{u}_0|}\ ,\ \mathbf{u}_2 = \mathbf{u}_0 \times \mathbf{u}_1$$

Para o caso dos quaternions, se q é o quaternion que representa a matriz $R(t)$ e ω é o quaternion que representa a velocidade angular do corpo, a derivada de $q(t)$ com relação ao tempo poderá ser obtida por $\frac{1}{2}\omega(t)q(t)$. Uma integração numérica sobre essa equação novamente produzirá erros numéricos, após algumas iterações do simulador, o que precisa ser corrigido através de uma normalização. Porém, usando quaternions, o número de erros é bem menor, permitindo aplicar a normalização com uma freqüência menor do que a que foi escolhida para matrizes de rotação. Além disso, a normalização de quaternions é bem menos custosa do que a ortonormalização de Gram-Schmidt, sendo dada por:

$$q = \frac{\hat{q}}{|\hat{q}|}$$

3.1.4 Equações do movimento

Podemos, então, definir um novo vetor de estados para atender às necessidades de um corpo rígido maior que uma partícula. O novo vetor terá a seguinte forma:

$$\mathbf{S}(t) = \begin{bmatrix} \mathbf{x}(t) \\ q(t) \\ \mathbf{p}(t) \\ \mathbf{L}(t) \end{bmatrix}$$

As equações do movimento, por sua vez, serão definidas da seguinte forma:

$$\frac{\mathrm{d}\mathbf{S}}{\mathrm{d}t} = \begin{bmatrix} \dot{\mathbf{x}} \\ \dot{q} \\ \dot{\mathbf{p}} \\ \dot{\mathbf{L}} \end{bmatrix} = \begin{bmatrix} m^{-1}\mathbf{p} \\ \omega q/2 \\ \mathbf{F} \\ \tau \end{bmatrix} \tag{3.3}$$

onde $\mathbf{F}$ e τ são, respectivamente, a força e o torque. Para n corpos rígidos, assim como ocorreu com as partículas, basta adicionar ao vetor as variáveis de cada corpo que compõem o vetor de estados.

A simulação consistirá em manter estas variáveis de estado atualizadas a cada intervalo de tempo. As únicas variáveis que podem ser fornecidas diretamente em cada instante são a força e o torque. O momento linear, o momento angular e o quaternion de orientação são calculados pelo método de solução de equações diferenciais. Todas as outras quantidades, como velocidade linear e angular, podem ser atualizadas a partir dessas variáveis.

3.2 Movimento restrito

Na realidade, não existe movimento totalmente irrestrito em aplicações físicas. Para que a física possa parecer o mais realista possível, algumas restrições têm que ser impostas ao movimento dos corpos. Por exemplo, os corpos devem interagir entre si, não sendo permitida a interpenetração de ambos, de modo que caso um corpo atinja outro, um efeito de resposta deve ser produzido para impedir que os dois corpos ocupem a mesma posição naquele instante de tempo.

A abordagem proposta por Baraff (1997b) e Mirtich (1996) é baseada em impulsos, que serão explicados um pouco mais adiante. Baraff (1997b) ainda propõe o uso de restrições que reforcem a idéia de não-penetração. Apesar de antiga, a abordagem de Baraff (1997b) ainda é bastante popular na solução de problemas envolvidos com simulação física, principalmente na área de jogos. Entretanto, existem outras alternativas, como o uso da dinâmica lagrangiana, que será discutida mais à frente neste capítulo.

3.2.1 Colisões

A simulação física do movimento restrito de corpos pode ser dividida em duas fases principais. A **detecção de colisões**, que é um problema de geometria, visa identificar os corpos que sofreram algum tipo de contato entre si e em que ponto ocorreu este contato.

A segunda fase é a **resposta às colisões**, que vai determinar os efeitos que a colisão causará no estado de cada um dos corpos no próximo instante da simulação. A resposta às colisões caracteriza um problema tipicamente físico, já que os efeitos aplicados aos corpos serão derivados a partir de equações baseadas nas leis que regem a física de corpos rígidos.

Existem dois tipos possíveis de contatos. Um deles é o contato de colisão, que ocorre quando um corpo tende a penetrar o outro corpo. Já quando um corpo está em repouso sobre outro corpo, ou a velocidade de um corpo faz com que ele deslize sobre o outro, sem afastamento nem interpenetração, diz-se que o contato é um contato de repouso.

Uma forma simples de se determinar o tipo de contato é verificar o resultado do produto escalar da velocidade do corpo A pela normal da superfície do corpo B no ponto de contato P. A Figura 4 ilustra esta situação:

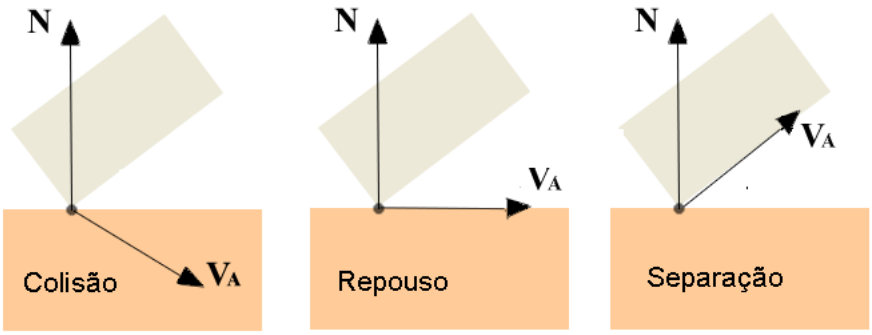


Figura 4: Tipos de contato entre dois corpos

O produto escalar $\mathbf{N} \cdot \mathbf{V}_A$ representa a velocidade do corpo A na direção normal. Se o produto escalar for menor que zero, então temos um contato de colisão, ou seja, a velocidade do corpo A na direção normal está em sentido contrário ao do vetor normal, isto é, o corpo A está indo de encontro ao corpo B. Analogamente, se o produto é igual a zero, temos um contato de repouso, e se o produto escalar é maior que zero, então não temos um contato, e os corpos estão se separando.

O conjunto de pontos de contato será determinado pelo tipo de interseção que ocorre. Podemos ter quatro tipos distintos de interseção entre dois corpos rígidos:

- vértice-face
- aresta-aresta

- aresta-face
- face-face

Os dois primeiros tipos são fáceis de se tratar. O problema surge quando há a necessidade de se computar o conjunto de pontos de uma interseção aresta-face ou face-face. O conjunto de pontos nesses dois casos é infinito, o que torna o cálculo mais custoso. Em aplicações de tempo real, utiliza-se um conjunto reduzido de pontos de contato, no qual interseções desses dois tipos são reduzidas a interseções vértice-face e aresta-aresta. Dessa forma, o conjunto fica reduzido a um número finito de pontos, tornando a computação mais rápida.

3.2.2 Impulsos

Impulsos são forças que agem por períodos de tempo muito curtos. Agem sobre os corpos somente o suficiente para causar uma alteração na velocidade deles. A idéia do impulso é que ele seja utilizado no instante de contato entre os corpos para causar uma mudança na velocidade relativa de ambos e impedir a interpenetração. O problema é que essa mudança deve ser feita de maneira descontínua, o que não é fisicamente possível, já que os corpos, quando se colidem, alteram sua velocidade suavemente, mesmo que essa alteração ocorra em uma fração de segundo.

Suponha que uma partícula possua uma velocidade inicial v_0 e que em determinado ela atinja um objeto. Algum tempo após a colisão, a velocidade da partícula será $-v_0$ (assumindo uma colisão perfeitamente elástica). A mudança da velocidade ocorre em uma intervalo de tempo $\epsilon > 0$. Fazendo ϵ tender a zero, obtemos uma mudança na velocidade de forma descontínua. Dessa forma, obtemos a distribuição abaixo, conhecida como **delta de Dirac**, que nos ajuda na obtenção da força impulsiva.

$$\delta(t - t_0) = \begin{cases} 0, & t \neq t_0 \\ \infty, & t = t_0 \end{cases}$$

Observe a integração da segunda lei de Newton:

$$m\mathbf{v}(t) - m\mathbf{v}(0) = \int_0^t \mathbf{F}(\tau)\mathrm{d}\tau \tag{3.4}$$

Braun (1984) mostra que se incluirmos forças impulsivas no lado direito da equação 3.4, podemos causar uma descontinuidade no momento linear do corpo. Uma forma mais

simples, entretanto, de resolver essa questão é, ao invés de construir uma função impulsiva como citado acima, escolher a velocidade desejada após a aplicação do impulso no corpo. Essa velocidade será dada em função da velocidade inicial, da normal à superfície do corpo no ponto de contato e do **coeficiente de elasticidade** ϵ. Esse coeficiente representa a quantidade de energia cinética que é perdida no momento da colisão. Se $\epsilon = 0$, tem-se uma colisão perfeitamente inelástica, onde ocorre perda total de energia cinética, fazendo com que os corpos permaneçam em contato após a colisão. Quando $\epsilon = 1$, trata-se de uma colisão perfeitamente elástica, quando não há perda de energia cinética e ocorre uma reflexão perfeita do corpo através da normal. Na prática, as colisões não são perfeitamente elásticas, de forma que ϵ sempre variará entre 0 e 1. Uma forma de se determinar o coeficiente de elasticidade é através das velocidades dos dois corpos antes e depois da colisão.

$$\epsilon = -\frac{v_1^+ - v_2^+}{v_1^- - v_2^-} \tag{3.5}$$

O símbolo + indica os atributos do corpo após a colisão e o − indica os atributos antes da colisão. Suponha uma força impulsiva $\mathbf{F} = f\mathbf{N}_0$ agindo sobre dois corpos A e B no instante de contato entre eles. As velocidades linear e angular após o impulso para o corpo A podem ser relacionadas pelas seguintes equações:

$$\mathbf{v}_A^+ = \mathbf{v}_A^- - \frac{f\mathbf{N}_0}{m_A}$$

$$\mathbf{w}_A^+ = \mathbf{w}_A^- - J_A^{-1}(\mathbf{r}_A \times f\mathbf{N}_0)$$

De forma análoga, as velocidades do corpo B são dadas por:

$$\mathbf{v}_B^+ = \mathbf{v}_B^- - \frac{f\mathbf{N}_0}{m_B}$$

$$\mathbf{w}_B^+ = \mathbf{w}_B^- - J_A^{-1}(\mathbf{r}_B \times f\mathbf{N}_0)$$

A partir destas equações, podemos chegar à fórmula para a obtenção do módulo da força impulsiva, como demonstrado em (EBERLY, 2004).

$$f = \frac{-(1+\epsilon)(\mathbf{N}_0 \cdot (\mathbf{v}_A^- - \mathbf{v}_B^-) + (\mathbf{w}_A^- \cdot (\mathbf{r}_A \times \mathbf{N}_0) - \mathbf{w}_B^- \cdot (\mathbf{r}_B \times \mathbf{N}_0)))}{m_A^{-1} + m_B^{-1} + (\mathbf{r}_A \times \mathbf{N}_0)J_A^{-1}(\mathbf{r}_A \times \mathbf{N}_0) + (\mathbf{r}_B \times \mathbf{N}_0)^T J_B^{-1}(\mathbf{r}_B \times \mathbf{N}_0)} \tag{3.6}$$

Como podemos ver, a força de impulso é calculada em função da massa dos corpos. Porém, se quisermos simular corpos imóveis, não podemos permitir que as forças impulsivas apliquem qualquer tipo de movimento neles. Sendo assim, uma solução é considerar o corpo como tendo massa infinita. Dessa forma, o inverso da massa será 0 e o inverso do

tensor de inércia será a matriz nula. Uma vez que na simulação apenas os inversos dessa grandezas serão utilizados diretamente nos cálculos, torná-los iguais a zero é suficiente para impedir que estes corpos adquiriram movimento linear ou angular.

3.2.3 Múltiplos pontos de contato

Quando um corpo faz contato com outro em mais de um ponto simultaneamente, o tratamento que deve ser dado a essa colisão não é tão simples como nos casos anteriores. Como os pontos de contato são processados seqüencialmente, a aplicação de uma resposta ao contato de um ponto pode afetar de maneira não desejada o outro contato.

Figura 5: Corpos com mais de um ponto de contato

Por exemplo, suponha uma caixa em contato com outros dois corpos através de dois pontos P_0 e P_1, como ilustra a Figura 5. Processando o contato em P_0, obteremos uma resposta à colisão que causará uma alteração nos momentos linear e angular da caixa. A partir daí, temos duas opções: processar o contato em P_1 ou executar o algoritmo para solução da equação diferencial. Se a segunda opção for escolhida, pode ser que P_1 não seja mais um ponto de contato após a execução do algoritmo, ou, o que é pior, pode ser que a caixa tenha sofrido uma rotação de forma que a extremidade do ponto P_1 penetrou no corpo, quebrando a condição de não-penetração dos corpos. Se, por outro lado, escolhermos processar o ponto P_1, teríamos que usar as velocidades obtidas com o processamento de P_0 ao invés da velocidade atual do corpo, uma vez que o processamento de P_0 alterou o estado do corpo. Isso pode fazer com que o impulso aplicado em P_1 não seja adequado, causando um comportamento físico irreal.

O mais simples a ser feito é processar um ponto e alterar o estado dos corpos que compartilham aquele ponto. A seguir a simulação avança um passo e verifica-se novamente se houve colisões. Caso haja alguma interpenetração, o algoritmo deve ser executado

novamente, a partir do estado original, porém com um intervalo de tempo menor, até que não ocorram mais interpenetrações ou que um número máximo de tentativas seja atingido.

De forma a tornar este processo simples e automático, podemos determinar algumas condições para as forças de contato. Para isso, a força de contato $\mathbf{C} = f\mathbf{N}_0$ exercida por um corpo B em um corpo A deve atingir às seguintes condições:

- agir somente no instante de contato
- ser repulsiva
- prevenir a interpenetração dos corpos
- não implicar em ganho de energia cinética para o sistema

Para satisfazer às duas primeiras condições, f deve ser, em primeiro lugar, maior ou igual a zero. Além disso, f deve ser escolhida em função da velocidade dos corpos. Seja d a componente normal da velocidade do corpo A. Se $d < 0$, o corpo A está tentando penetrar no corpo B, como visto na seção 3.2.1. Dessa forma, f deve ser escolhida de tal forma que impeça essa interpenetração. Por fim, para impedir que ocorra ganho de energia cinética para o sistema, essa força deve ser limitada por um valor máximo em função da velocidade inicial do corpo.

Essa situação pode ser resolvida através da definição de uma função quadrática convexa sujeita às restrições acima. Dessa forma, define-se um problema de complementaridade linear, que tem como objetivo determinar f de forma que minimize a função objeto. Esse problema pode ser resolvido utilizando o algoritmo de Lemke-Howson (LEMKE; HOWSON, 1964).

Essa solução pode ser adaptada para casos em que ocorre uma colisão de repouso. Como na colisão de repouso a distânca e a velocidade relativa entre os corpos é zero, nós utilizamos a aceleração dos corpos para calcular uma força de contato que atenda às restrições impostas anteriormente. Neste caso específico, não precisamos nos preocupar com o ganho de energia cinética, pois a velocidade dos corpos é zero.

3.2.4 Lagrange

Observando a Figura 6, vemos que o corpo inicialmente está sobre uma mesa. Logo, existe um contato de repouso entre ambos. A partir do momento em que ele cai da mesa,

esse contato de existir. Ao tocar o chão, inicialmente o contato é um contato de colisão. Após algum tempo, o corpo finalmente pára sobre o chão e restabele-se o contato de repouso.

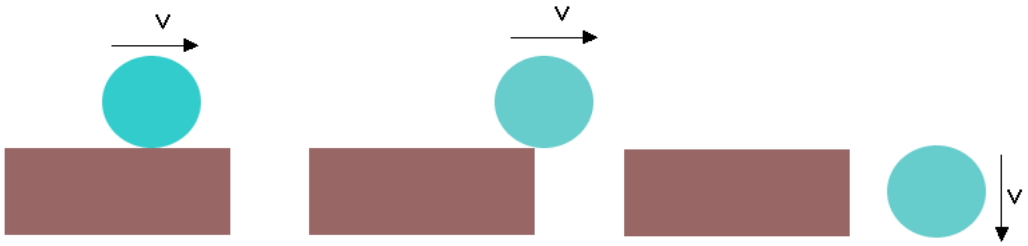


Figura 6: Um corpo alternando entre contato de repouso e de colisão

Podemos aproveitar situações como essas para utilizar a dinâmica de Lagrange. A grande vantagem do uso de Lagrange, como dito no Capítulo 2, é a facilidade de se tratar restrições com essa abordagem. O chão na figura, por exemplo, é um tipo de restrição conhecida em tempo de projeto. É possível tirar proveito disso e definir equações lagrangianas do movimento para impedir a interpenetração entre os corpos e o chão.

A desvantagem dessa idéia é que as equações lagrangianas serão específicas para a aplicação, ou seja, se o chão ou alguma outra superfície de restrição de outra aplicação física for diferente, as equações terão que ser reescritas. Isso pode ser um problema, uma vez que não é possível construir componentes reutilizáveis para aplicações físicas em geral, o que aumenta os custos de desenvolvimento.

Existe, contudo, a possibilidade de se utilizar um sistema híbrido, capaz de alternar, quando necessário, entre a dinãmica de Newton e a dinâmica de Lagrange. O único porém é a maior complexidade para projetar e implementar um motor físico que suporte esse tipo de funcionalidade.

3.3 A simulação física

Basicamente, a simulação consiste em detectar as colisões entre os corpos, determinar a resposta a essas colisões e executar um passo do algoritmo que soluciona as equações diferenciais, atualizando os estados dos corpos. A detecção e a resposta às colisões foram discutidas na seção anterior. Serão apresentados agora, sucintamente, alguns métodos para resolução das equações diferenciais.

3.3.1 Método de Euler

Este é o método mais simples de se implementar para resolução de equações diferenciais. Ele simplesmente calcula o estado no instante de tempo $(t + \delta t)$, multiplicando a derivada do estado atual no tempo t pelo intervalo δt, e somando o resultado ao estado inicial.

Apesar de sua simplicidade, o método de Euler é bastante impreciso, como mostrado por Baraff e Witkin (1997). À medida que a simulação avança, o erro acumulado vai aumentando cada vez mais. Pode-se diminuir o intervalo para reduzir os erros, porém os erros nunca irão desaparecer completamente.

Além disso, o método de Euler é bastante instável. Para intervalos bastante pequenos, o comportamento do método é aceitável, mas à medida que esse intervalo cresce, o método começa a divergir, até perder totalmente a solução.

Para finalizar, este método tem um custo computacional elevado. O método de Euler utiliza apenas um cálculo por iteração e ainda assim é bastante impreciso. Isso nor força a reduzir bastante o intervalo para o cálculo em cada iteração, tornando o método mais lento.

Todos esses problemas ocorrem pois o método de Euler é obtido pelo truncamento de todos os termos da série de Taylor após o segundo termo. Isso faz com que ele gere um erro grande para funções não-lineares.

3.3.2 Método do ponto médio

O método do ponto médio, também chamado de Runge-Kutta de segunda ordem, simplesmente evita o truncamento de mais um termo na série de Taylor. Este método realiza primeiro um cálculo igual ao do método de Euler e, em seguida, realiza um segundo cálculo usando o ponto médio do intervalo considerado. Porém, este método ainda não é a melhor opção.

3.3.3 Método de Runge-Kutta

Expandindo a série de Taylor um pouco mais, chegamos ao método de Runge-Kutta de quarta ordem, que é o método mais utilizado nas simulações físicas. É relativamente simples de implementar e possui uma estabilidade e uma precisão muito superior às dos métodos anteriores.

Ele consiste em realizar quatro cálculos, usando intervalos distintos, eliminando cada vez ordens mais altas de derivações. Seja uma função f e um valor inicial x_0, temos que (PRESS et al., 1986):

$$k_1 = hf(x_0, t_0)$$
$$k_2 = hf(x_0 + \frac{k_1}{2}, t_0 + \frac{h}{2})$$
$$k_3 = hf(x_0 + \frac{k_2}{2}, t_0 + \frac{h}{2})$$
$$k_4 = hf(x_0 + k_3, t_0 + h)$$
$$x(t_0 + h) = x_0 + \frac{1}{6}k_1 + \frac{1}{3}k_2 + \frac{1}{3}k_2 + \frac{1}{6}k_4$$

3.4 Motores físicos

Um motor físico é um software que define um conjunto de funcionalidades para a simulação de modelos físicos. Ele permite simular e prever diversos efeitos sob diferentes condições. É possível simular a vida real, por exemplo, ou um mundo imaginário qualquer imaginado pelo desenvolvedor. Ele fornece uma interface para que o desenvolvedor da aplicação física possa configurar os parâmetros necessários para a simulação. Idealmente, o motor físico deve ser independente de qualquer sistema gráfico, de modo que as aplicações que venham a ser desenvolvidas com o uso dele possam ter liberdade de escolha nesse sentido.

Existem basicamente dois tipos de motores físicos: os motores de alta precisão e os de tempo real. Os motores de alta precisão são mais utilizados em aplicações científicas e em animações. Não existe uma grande preocupação com velocidade, e sim com estabilidade e fidelidade aos fenômenos sendo simulados, buscando aproximá-los o máximo possível da realidade. Já os motores de tempo real são mais utilizados para entretenimento e aplicações iterativas, nos quais a velocidade é fator preponderante, às vezes sendo necessário sacrificar um pouco a precisão da simulação. O motor que será desenvolvido no próximo capítulo é um motor de tempo real.

4 Desenvolvimento

Neste capítulo, serão tratados os detalhes do desenvolvimento de um motor físico de tempo real simplificado, capaz de simular o movimento de corpos rígidos. Serão discutidas as estruturas de dados utilizadas na representação dos corpos e das geometrias de colisão, os relacionamentos entre as estruturas, o funcionamento geral do motor e os métodos utilizados.

4.1 Classes

As seções a seguir mostram uma estrutura orientada a objetos para o desenvolvimento de um motor físico, com as classes que foram usadas para representar todas as entidades necessárias no momento da simulação.

4.1.1 Corpos

A classe central no modelo do motor é a classe `RigidBody`. Ela é a extensão de uma classe abstrata chamada `AbstractBody` que representa um corpo genérico com todos os atributos necessários para a simulação física e os métodos que manipulam esses atributos durante a execução do motor. Os atributos podem ser categorizados em valores constantes, variáveis de estado e quantidades derivadas. Os atributos constantes são definidos no momento da inicialização do corpo rígido. São eles:

- `mass`: define a massa do corpo rígido
- `inertiaTensor`: define o tensor de inércia do corpo
- `invMass`: armazena o inverso da massa, para reduzir os cálculos
- `invTensor`: armazena o inverso do tensor de inércia, também com o objetivo de reduzir os cálculos

Existem atributos, porém, que precisam ser atualizados a cada instante de tempo na simulação. Estes atributos são as variáveis de estado, que determinam o estado do corpo rígido no instante corrente. Estas variáveis são as que compõem o vetor de estados usado na resolução da equação diferencial do movimento. As quatro variáveis de estado são as que seguem:

- `position`: define a posição corrente do centro de massa do corpo no mundo
- `orientation`: define a orientação do corpo em relação ao mundo através de um quaternion
- `linearMomentum`: define o momento linear do corpo
- `angularMomentum`: define o momento angular do corpo

Os atributos `rotationMatrix`, `linearVelocity` e `angularVelocity` são quantidades derivadas das variáveis de estado. Os outros atributos que não se enquadram nas categorias acima são: `dynamicsMode`, que tem como única finalidade definir se as equações do movimento utilizadas para o cálculo das variáveis de estado do corpo serão equações newtonianas ou lagrangianas, e `coefficientOfRestitution`, que define o coeficiente de elasticidade do corpo, que influencia diretamente nas colisões. Abaixo, uma breve descrição de cada um deles:

- `rotationMatrix`: define uma matriz de rotação do corpo rígido, derivada a partir do quaternion de orientação
- `linearVelocity`: define a velocidade linear corrente do corpo
- `angularVelocity`: define a velocidade angular corrente do corpo

A classe `RigidBody`, derivada da classe acima, possui como atributo adicional apenas `collisionGeometry`, que é um ponteiro para uma geometria de colisão genérica, que pode ser associada ao corpo para fins de detecção de colisão. A classe `ComplexBody`, por sua vez, define um corpo agregado por vários corpos. Com as geometrias associadas aos corpos menores, é possível criar corpos de geometria arbitrária e tratá-los como um só.

4.1.2 Mundo físico

Para que os corpos rígidos possam fazer parte da simulação e interagir uns com os outros, eles devem fazer parte de um mundo, que é uma entidade que serve de base no

modelo. Essa entidade é representada pela classe `PhysicsWorld`, que mantém uma lista dos corpos que fazem parte do mundo, bem como uma lista dos contatos identificados entre os corpos na execução atual do simulador. Essa classe possui ainda outros atributos de estado, como o método numérico utilizado para a resolução das equações diferenciais, o intervalo de tempo utilizado e dois vetores de estados, que são utilizados para guardar os estados de entrada e de saída de todos os corpos num determinado instante de tempo. A classe possui ainda um vetor de estados de *backup*, para que o simulador possa retornar a um estado anterior, caso haja necessidade.

4.1.3 Geometrias

As classes apresentadas acima são suficientes para a simulação do movimento irrestrito de partículas. Porém, nem a classe `RigidBody` nem a `ComplexBody` definem uma geometria para o corpo. Para que eles possam deixar de ser simples partículas, é necessário que uma geometria de colisão seja associada aos corpos. É essa mesma geometria que permitirá que eles interajam entre si, podendo ocorrer interseções entre eles.

A classe `CollisionGeometry` representa essas geometrias. É uma classe abstrata, que possui quatro atributos: um para definir a geometria específica da classe derivada, um atributo estático que guarda o ponto de interseção encontrado na última detecção de colisão, e dois ponteiros que apontam para a posição atual e a posição anterior do corpo associado à geometria, de forma que a posição da geometria fique sempre atualizada com a do corpo. Sendo abstrata, essa classe torna trivial estender o motor para qualquer tipo de geometria, bastando, para isso, criar uma classe derivada que defina seus próprios atributos e implemente os métodos abstratos da classe base.

4.1.4 Contatos

Identificadas as colisões, é necessário dar-lhes uma resposta. Para isso, existe a classe Contact, que define uma interseção entre dois corpos. Possui como atributos os dois corpos rígidos que sofrem a colisão, o ponto em que ocorreu o contato, o tipo de contato ocorrido, que pode ser de colisão ou de repouso, e um atributo que define se o tipo de interseção é vértice-face ou aresta-aresta. Possui ainda um atributo que armazena a normal unitária da superfície de contato e, para o caso de um contato aresta-aresta, armazena as arestas dos dois corpos.

4.1.5 Noções matemáticas

Existem ainda três classes matemáticas, que dão suporte às operações matemáticas. Essas classes são as classes `Vector`, `Matrix` e `Quaternion`, que representam, respectivamente, vetores, matrizes e quaternions. Cada uma delas define as operações básicas necessárias. A classe `Quaternion` possui ainda funções que relacionam quaternions a matrizes de rotação.

4.2 Hierarquia de classes

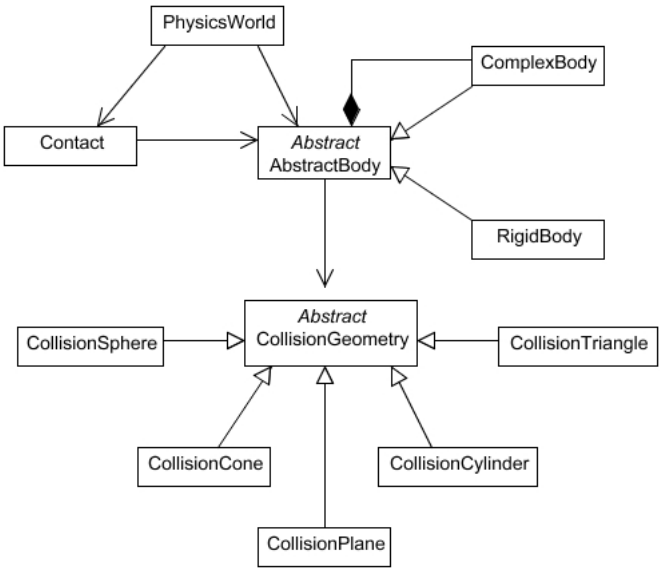


Figura 7: Diagrama da hierarquia de classes

A Figura 7 mostra a hierarquia das classes do motor físico. Como pode ser visto, as classes `RigidBody` e `ComplexBody` herdam da classe abstrata `AbstractBody`, de forma que qualquer objeto, seja ele um corpo simples ou um corpo agregado, possa ser incluído na lista de corpos do mundo. Este mundo é representado pela classe `PhysicsWorld`, que possui não só a lista de corpos como a lista de objetos da classe `Contact`. Esta classe é composta de duas instâncias de `AbstractBody`, como explicado na seção anterior.

Cada corpo possui uma geometria de colisão associada, que é representada no diagrama pela classe `CollisionGeometry`. Esta também é uma classe abstrata, que serve de base para classes concretas, como `CollisionSphere`, `CollisionCylinder` e qualquer outra geometria que seja de interesse incluir no motor.

4.3 Funcionamento geral do motor

Para que se possa criar a mais simples das aplicações físicas usando o motor desenvolvido, é necessário, antes de tudo, criar um mundo. Esse mundo será o objeto ao qual todos os elementos da sua aplicação serão associados. É possível criar vários mundos diferentes, todos independentes entre si.

É possível configurar o intervalo de tempo utilizado na simulação e o método numérico utilizado para resolução das equações diferenciais do movimento, o que permite escolher entre aplicações mais realistas e estáveis ou aplicações com menos acurácia, porém com mais velocidade de processamento. Todos os métodos implementam a dinâmica newtoniana, mas foi deixada a liberdade de se estender futuramente o projeto para a inclusão da dinâmica lagrangiana para dar suporte às colisões de repouso. É possível também definir a aceleração da gravidade no mundo, além de criar e adicionar novos corpos a ele.

4.3.1 Funcionamento interno

Internamente, o motor se utiliza de uma série de métodos privados para simular o mundo de acordo com a configuração determinada e com o estado corrente de cada corpo. O motor determina o tamanho dos vetores de estado e executa um passo da simulação através do método `simulate()`. Ele prepara o vetor de estados de entrada, resolve as equações diferenciais e atualiza todos os corpos com o vetor de saída.

Em seguida, é chamada a rotina de detecção de colisões. Caso ela encontre alguma colisão, o método de resposta será chamado, determinando os impulsos aplicados aos corpos em contato. Se houve alguma interpenetração de corpos, o método de detecção de colisões pára imediatamente e o simulador carrega o estado anterior dos corpos, e reexecuta o mesmo passo, porém com um intervalo de tempo menor. Isso é repetido até que a interpenetração não ocorra mais ou que se atinja um número limite de tentativas.

O fato de o método `simulate()` executar apenas um passo da simulação permite que o motor fique independente de qualquer sistema gráfico, permitindo que o desenvolvedor da aplicação se utilize do estado de cada corpo entre cada passo da simulação para exibir a saída da simulação da maneira que lhe aprouver. Na Figura 8 é possível visualizar o grafo de transição de estados da simulação.

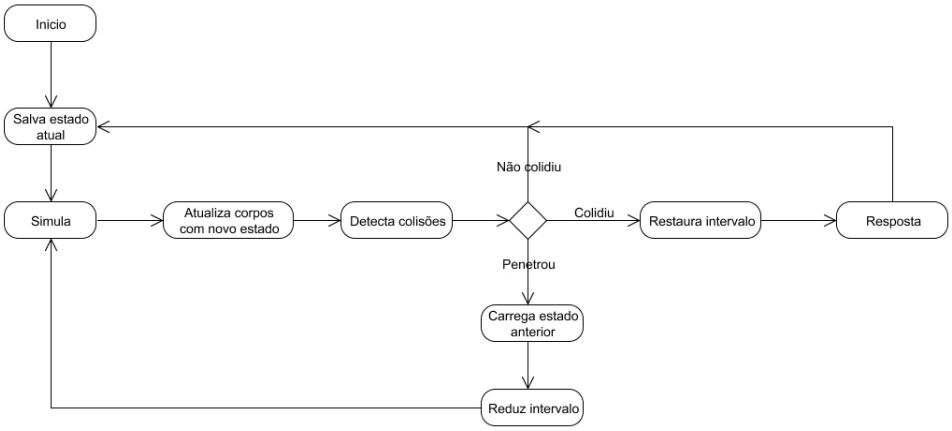


Figura 8: Grafo de transição de estados

4.3.2 Configuração dos corpos

Os corpos criados e adicionados ao mundo podem ter a maioria dos seus atributos modificados. Alguns, entretanto, não é recomendado que se modifiquem manualmente, deixando esse trabalho para o motor. Um exemplo é o atributo `position`, que define a posição do centro de massa do corpo no mundo. Para partículas, não há problemas em alterar essa posição manualmente, mas para corpos que possuem geometria, isso deve ser evitado, pois o centro de massa é calculado automaticamente em função da geometria associada a ele, no momento em que essa geometria é associada ao corpo. O ideal é que se defina os parâmetros da **geometria** antes de associá-la ao corpo, para que todos os cálculos sejam efetuados corretamente. O mesmo vale para o tensor de inércia, que é calculado no mesmo instante do centro de massa.

Forças externas devem ser associadas ao corpo quando necessário e retiradas quando não houver mais a necessidade de aplicá-las. Isso ocorre pois o motor não tem como saber qual o intervalo de tempo em que a força terá efeito. Por isso, ele considera todas as forças como sendo de ação constante. Forças inerentes ao mundo, que agem sobre todos os corpos simultaneamente, são calculadas automaticamente pelo motor, desde que os parâmetros do mundo estejam configurados corretamente, como é o caso da força exercida pela gravidade. Analogamente, o mesmo pode ser feito com os torques.

4.3.3 Detecção e resposta às colisões

As geometrias de colisão são as entidades que determinam as dimensões dos corpos. Sem elas, os corpos são apenas partículas. Além de determinar a forma e a dimensão dos corpos, são elas que permitem verificar se dois corpos colidiram em determinado instante de tempo. É possível chamar a função de detecção de colisões explicitamente, mas normalmente, é mais simples deixar essa tarefa a cargo do motor, que identifica e responde às colisões automaticamente. Cada geometria de colisão possui um método que identifica qual a geometria dos corpos em contato e chama um outro método para tratar o caso específico. Vários métodos de detecção de colisão para diferentes geometrias podem ser obtidos em (BERGEN, 2004) e (GOMEZ, 1999).

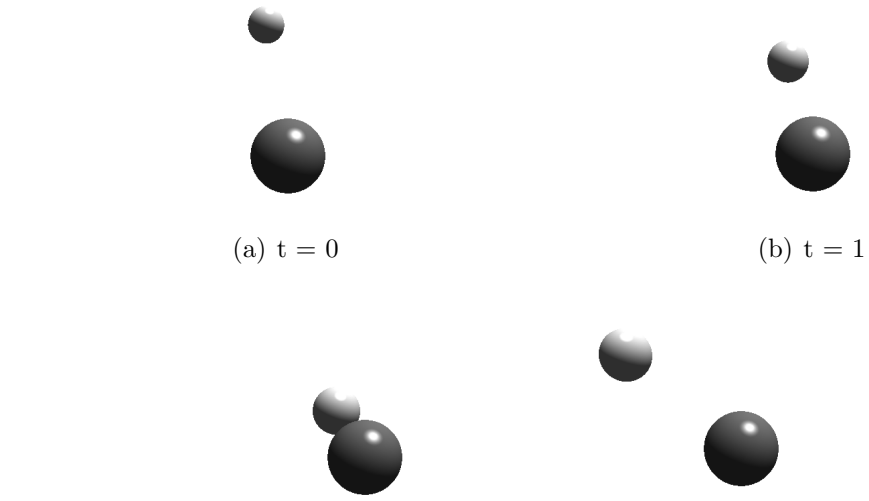
Por fim, a classe `Contact` é a responsável pelo tratamento das colisões. Ela identifica o tipo de contato que ocorreu e chama o método apropriado. Ela também calcula o coeficiente de elasticidade, baseado nos coeficientes dos corpos em contato.

4.3.4 Exemplos

Para testar o motor desenvolvido e comprovar o correto funcionamento do mesmo, foram realizadas algumas simulações simples. Como primeiro exemplo, foram simuladas duas esferas que se colidem num instante de tempo t. Na Figura 9, é possível observar a posição de cada esfera em quatro instantes diferentes.

A esfera mais escura, que está embaixo, possui massa infinita, o que significa que ela não irá se mover em hipótese alguma. A esfera mais clara está sobre a ação da gravidade e cai com uma aceleração de $9.8m/s^2$ na direção da outra esfera. Observe que, como o ponto de interseção é um pouco deslocada para a esquerda do centro da esfera de baixo, a força impulsiva gerada pela colisão causará uma reflexão na esfera mais clara para a esquerda.

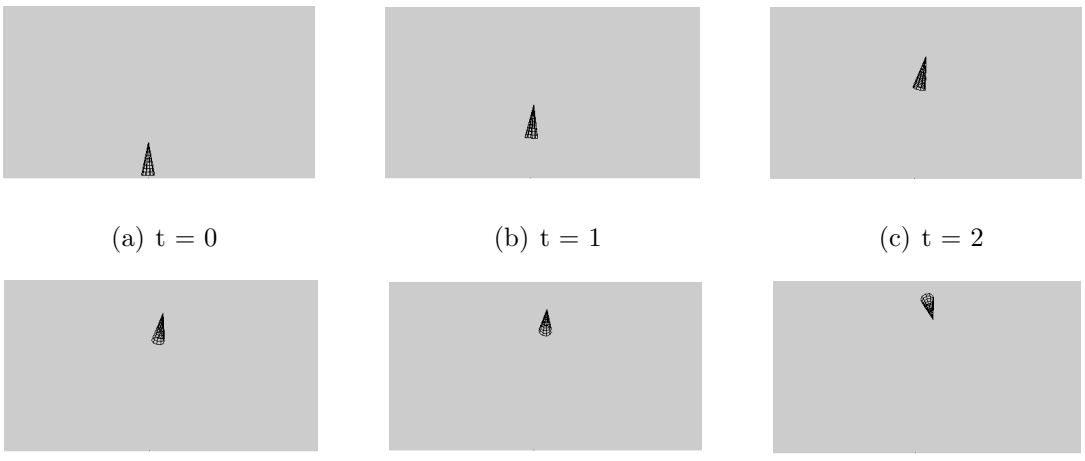
Outra simulação realizada foi a de um pequeno foguete, com uma força agindo constantemente na sua base contra a força da gravidade. Além disso, foi apliacada uma velocidade linear inicial de $100m/s$ para a direita, causando um torque no foguete que o leva a um movimento de rotação no ar, como pode ser observado na Figura 10, que mostra o foguete em seis instantes diferentes de tempo. Nos três primeiros, ele está subindo. Na Figura 10(d), o foguete começa a girar, perdendo o equilíbrio das forças.



(a) t = 0 (b) t = 1

(c) t = 2 (d) t = 3

Figura 9: Simulação de duas esferas se colidindo



(a) t = 0 (b) t = 1 (c) t = 2

(d) t = 3 (e) t = 4 (f) t = 5

Figura 10: Simulação simplificada do lançamento de um foguete

5 Aplicação

Este capítulo trata da utilização do motor físico detalhado no capítulo anterior, visando definir um modelo para o desenvolvimento de uma aplicação que simule a física de uma aeronave genérica. O objetivo aqui é mostrar basicamente como funciona a física de um avião e como isso pode ser implementado usando os conceitos discutidos até agora. Aviões de modelos específicos exigiriam maior riqueza de detalhes no momento da implementação, para que ele se comporte realisticamente. O objetivo aqui é simular, de forma satisfatória, uma aeronave de caráter geral.

5.1 Princípios básicos de aeronaves

As aeronaves podem ser resumidas em algumas partes fundamentais. Em primeiro lugar, temos a fuselagem, que é o corpo principal do avião. Acopladas a ela, temos as asas. Cada asa possui um aileron e um flape. Os ailerons são partes móveis das asas e cauda dos aviões que têm a função de controlar a direção da aeronave. Os flapes, por sua vez, são superfícies existentes nos bordos de fuga das asas que permitem aumentar a capacidade de sustentação quando abaixados. Temos ainda os profundores, que são pequenas superfícies de sustentação parecidas com asas localizadas próximas à cauda da aeronave. Por fim, existe o leme, uma superfície vertical localizada na extremidade da cauda.

5.1.1 Forças

Toda aeronave é governada basicamente por quatro forças: a de empuxo, a resistência do ar (ou força de arraste), a da gravidade e a de sustentação (BOURG, 2002). Pode-se ver essas forças na Figura 11.

O empuxo é uma força produzida pelo propulsor do avião, que faz com que ele adquira uma aceleração e comece a se mover. Normalmente, podemos considerar o empuxo como

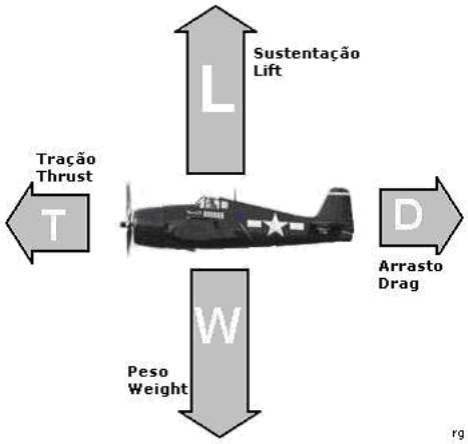


Figura 11: Forças que agem sobre uma aeronave

uma força que age na linha que passa pelo centro de gravidade do avião. Contrária ao empuxo, temos a força de arraste, produzida pelo atrito com o ar. Essa força tende a impedir o movimento do avião.

Perpendicularmente às duas forças anteriores, temos a força da gravidade e a força de sustentação. A primeira é a mesma força já abordada nos capítulos anteriores. A força de sustentação, entretanto, é uma força gerada pelas asas que tem como objetivo anular a ação da gravidade, mantendo o avião suspenso no ar. É o empuxo que permite às asas gerarem a força de sustentação.

5.1.2 Aerofólios

O aerofólio é uma superfície construída de forma a produzir uma força de sustentação a partir do movimento do ar através dele (ABBOT; DOENHOFF, 1959). Tem sua forma definida pela cambra, que é a medida da curvatura do aerofólio, e pela corda, que é a linha reta que liga o bordo de fuga ao bordo de ataque. Além disso, os aerofólios contêm os flapes, que quando possuem um ângulo de deflexão, alteram a curvatura da asa e outras propriedades do aerofólio, aumentando a sustentação.

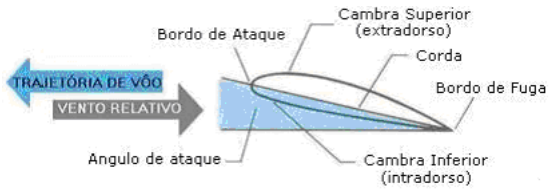


Figura 12: Esquema de um aerofólio

O aerofólio também possui um ângulo de ataque, que é medido entre a direção de viagem do aerofólio e a corda. Quanto maior esse ângulo, maior a sustentação da aeronave. Porém, se esse ângulo for muito grande, ele pode causar uma separação do fluxo de ar na parte superior do avião, causando um fenômeno conhecido como **estol**, em que as asas param de produzir sustentação e o avião começa a perder altitude.

A força de sustentação gerada é explicada pelo Princípio de Bernoulli. Este princípio diz que à medida que a velocidade de um fluido aumenta, a sua pressão interna diminui. No caso de uma aeronave, o fluido é o ar. O ar, ao chegar ao aerofólio, se divide. O fluxo de ar que se move sobre o aerofólio adquire maior velocidade do que aquele que passa por baixo do aerofólio, uma vez que ambos se encontram ao mesmo tempo no bordo de fuga. Com isso, o ar na parte de cima gera uma pressão menor, causando uma diferença de pressão entre as partes superior e inferior do aerofólio. Essa diferença de pressão faz com que surja uma força de sustentação na superfície.

Podemos definir os coeficientes de sustentação e arraste de um aerofólio pelas fórmulas a seguir:

$$C_L = \frac{L}{(1/2)\rho V^2 S}$$

$$C_D = \frac{D}{(1/2)\rho V^2 S}$$

onde ρ é a densidade do ar, V é a velocidade do aerofólio em relação ao ar, S é a área da superfície de sustentação, L é a força de sustentação e D é a força de arraste.

5.1.3 Eixos de vôo

Uma aeronave possui três eixos: o eixo longitudional, que vai da cauda ao bico do avião, o eixo lateral, que vai da extremidade de uma asa à outra, e o eixo vertical. Os três eixos se encontram no centro de gravidade do avião. Uma aeronave pode executar três tipos de movimento, que são rotações em torno de cada um desses eixos: a rolagem, a arfagem e a guinada.

Os ailerons são responsáveis pelo movimento de rolagem. Eles alteram a linha de corda e modificam a cambra de cada asa. Os ailerons sempre são movidos em sentido contrário. Dessa forma, o ângulo de ataque de uma das asas aumenta e o da outra diminui, causando uma diferença nas forças de sustentação, que faz o avião girar em torno do eixo longitudinal. Esse movimento pode ser visualizado na Figura 13. A figura mostra o movimento como seria observado pelo piloto na cabine da aeronave.

(a) (b) (c)

Figura 13: Movimento de rolagem de um avião

Os profundores produzem o movimento de arfagem. Quando eles são abaixados, a força de sustentação na parte traseira do avião aumenta, fazendo ele girar em torno do eixo lateral. Assim, a parte frontal da aeronave desce, num movimento conhecido como picada. O movimento inverso, ou seja, o movimento de subida do avião é conhecido como cabrada. A Figura 14 ilustra essa situação.

(a) (b) (c)

Figura 14: Movimento de arfagem de um avião

O leme, por sua vez, executa o movimento de guinada do avião. Esse movimento produz uma rotação em torno do eixo vertical. Se o leme for aplicado para a esquerda, a cauda irá se mover para a direita, fazendo a frente do avião girar para a esquerda. Analogamente, a aplicação do leme para a direita faz o avião girar para a direita. A guinada pode ser vista na Figura 15.

(a) (b) (c)

Figura 15: Movimento de guinada de um avião

5.2 Modelagem de uma aeronave

Tendo como base os princípios apresentados na seção anterior, é possível modelar uma aeronave, de modo a simular a física de um avião genérico. As classes que definem a aeronave possuem métodos específicos que se utilizam do motor físico apresentado no capítulo anterior para implementar o seu movimento.

5.2.1 A classe Airfoil

Uma vez que um avião possui vários aerofólios, com propriedades diferentes, que influem em sua sustentação, é válido definir uma classe para representá-los. No modelo elaborado, foi criada uma classe `Airfoil`, que possui atributos como o ângulo de ataque, o comprimento da linha de corda, a área da superfície e um campo que define se o flape está ou não abaixado.

Um objeto da classe também possui um atributo que representa a superfície de sustentação do aerofólio. Essa superfície é nada mais que um ponteiro para um objeto da classe `RigidBody` do motor apresentado no capítulo anterior.

5.2.2 A classe Airplane

Para representar o avião, foi definida a classe `Airplane`. Esta classe é uma especialização da classe `ComplexBody`. É composta por oito atributos que representam as partes principais do avião: fuselagem, as duas asas, os dois ailerons, os dois profundores e o leme. A fuselagem é uma instância de `RigidBody`, podendo ser representada, por exemplo, por um cilindro. Caso houvesse a necessidade de uma precisão maior na representação do corpo do avião, a fuselagem poderia ser dividida em partes menores, e cada parte ser associada a uma geometria adequada.

Os outros atributos são instâncias da classe `Airfoil`, descrita na Seção 5.2.1, uma vez que cada aerofólio compõe uma parte da aeronave. Todos esses atributos, assim como a fuselagem, compõem a lista de corpos da classe `ComplexBody`, definindo, assim, um corpo de geometria arbitrária (nesse caso, a geometria de um avião) para ser associado a um mundo. Assim, é possível aplicar as equações do movimento à aeronave modelada.

5.2.3 Simulação da aeronave

As duas classes demonstradas acima definem métodos que permitem alterar todas as propriedades da aeronave, incluindo aquelas definidas pelas classes `AbstractBody`, `RigidBody` e `ComplexBody`.

Existem métodos, por exemplo, que permitem executar rotações em torno dos três eixos do avião. Os métodos `pitch()`, `yaw()` e `roll()` executam, respectivamente, os movimento de arfagem, guinada e rolagem, a partir dos ângulos passados como parâmetro. É possível também abaixar e levantar os flapes e modificar a força de propulsão do avião.

Cada aerofólio também permite, a partir de suas propriedades, calcular os coeficientes de sustentação e de arraste. Todos esses dados são utilizados posteriormente para o cálculo das forças que agem sobre o avião, permitindo configurar os atributos definidos pelas classes base. Após a definição desses atributos, todo o trabalho de determinação do movimento do avião cabe apenas ao simulador implementado pelo motor físico do capítulo anterior, que trata o avião como qualquer outro corpo, abstraindo do tipo e da forma dos corpos adicionados ao mundo.

6 Conclusão

Tendo chegado ao término deste trabalho, pode-se ver que a tarefa de modelar e simular um ambiente sujeito a leis físicas na maioria das vezes não é trivial. Em simulações nas quais várias simplificações são feitas, ganha-se em velocidade e em facilidade de desenvolvimento, mas, na maioria das vezes, perde-se em acurácia. Como foi visto, existem situações de difícil solução, principalmente no que tange ao processamento de pontos de contato, em que vários motores comerciais tendem a ter sérios problemas. Além disso, existem os problemas de ordem numérica, que precisam ser tratados com o devido cuidado.

Em contrapartida, a simulação de corpos específicos, como a aeronave trabalhada no capítulo anterior, fica relativamente fácil, uma vez que um motor físico dê suporte a todas as operações básicas para a simulação. Basta definir propriedades e funções adicionais, e utilizar as funções já disponibilizadas pelo motor para configurar os parâmetros necessários à simulação.

O campo da simulação física é bastante amplo e, obviamente, não se resume à simulação de simples corpos rígidos. Dentre os possíveis tópicos que podem ser explorados e desenvolvidos, temos a modelagem da dinâmica de corpos deformáveis, uma vez que todo corpo, na realidade, é de alguma maneira deformável. Ainda em relação a corpos rígidos, pode-se também desenvolver tópicos relativos à simulação de outros corpos mais complexos, bem como utilizar métodos alternativos a Newton e Lagrange para a modelagem da dinâmica de corpos. Enfim, esses são apenas alguns exemplos do que pode vir a ser explorado em futuros trabalhos.

Referências

ABBOT, I. H.; DOENHOFF, A. E. V. *Theory of Wing Sections*. New York: Dover, 1959.

BARAFF, D. *An Introduction to Physically Based Modeling: Rigid Body Simulation I - Unconstrained Rigid Body Dynamics*. 1997. Disponível em: <http://www.cs.cmu.edu/ baraff/sigcourse/notesd1.pdf>.

BARAFF, D. *An Introduction to Physically Based Modeling: Rigid Body Simulation II - Nonpenetration Constraints*. 1997. Disponível em: <http://www.cs.cmu.edu/ baraff/sigcourse/notesd2.pdf>.

BARAFF, D.; WITKIN, A. *Physically Based Modeling: Principles and Practice - Differential Equation Basics*. 1997. Disponível em: <http://www.cs.cmu.edu/ baraff/sigcourse/notesb.pdf>.

BERGEN, G. van den. *Collision Detection in Interactive 3D Environments*. San Francisco, CA: Morgan Kaufmann, 2004.

BOURG, D. M. *Physics for Game Developers*. Sebastopol, CA: O'Reilly & Associates, 2002.

BRAUN, M. *Differential Equations and Their Applications, Applied Mathematical Sciences*. [S.l.]: Springer-Verlag, 1984.

EBERLY, D. H. *Game Physics*. San Francisco, CA: Morgan Kaufmann, 2004.

GOLDSTEIN, H.; POOLE, C.; SAFKO, J. *Classical Mechanics*. 3. ed. San Francisco, CA: Addison-Wesley, 2002.

GOMEZ, M. Simple intersection tests for games. 1999. Disponível em: <http://www.gamasutra.com/features/19991018/Gomez_1.htm>.

HAMILTON, W. R. *Elements of Quaternions*. London: Longman, 1866.

LEMKE, C. E.; HOWSON, J. T. Equilibrium points of bimatrix games. 1964.

MIRTICH, B. *Impulse-based Dynamic Simulation of Rigid Body Systems*. Tese (Doutorado) — University of California at Berkeley, 1996.

PRESS, W. H. et al. *Numerical Recipes: The Art of Scientific Computing*. [S.l.]: Cambridge University Press, 1986.