

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA
DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E AUTOMAÇÃO INDUSTRIAL

ANIMADO:
Um Protótipo de um Sistema de Animação
Modelada por Dinâmica

autora: Maria Andréia Formico Rodrigues

orientador: prof. Léo Pini Magalhães *OK*

Dissertação submetida à Faculdade de Engenharia Elétrica da Universidade Estadual de Campinas, para preenchimento dos pré-requisitos parciais para obtenção do Título de Mestre em Engenharia Elétrica.

Este exemplar corresponde à redação final da tese
defendida por MARIA ANDRÉIA FORMICO
RODRIGUES e aprovada pela Comissão
Jugadora em 22 de 02 de 93.
LÉO PINI MAGALHÃES
Orientador

agosto 1993



Resumo

O objetivo geral desta dissertação é apresentar uma pesquisa que vem sendo realizada sobre o tema “Animação Modelada por Dinâmica” desde o ano de 1990. A partir desta pesquisa, o objetivo principal deste trabalho é a criação do **ANIMADO**¹, um Protótipo de um Sistema de Animação Modelada por Dinâmica, no âmbito do Projeto ProSim, no grupo de Computação de Imagens, no DCA - FEE - UNICAMP.

A Dinâmica refere-se à descrição do movimento como uma relação entre forças e torques atuando sobre massas. Desta forma, a Física é aplicada na intenção de modelar o movimento que será apresentado pelos objetos. Este movimento pode ser muito realista, dependendo do tipo e exatidão da modelagem envolvida. No **ANIMADO**, o animador poderá produzir movimentos selecionando forças e torques de controle que estarão diretamente relacionados às leis mecânicas que regem o movimento.

Basicamente, o **ANIMADO** é composto por cinco módulos totalmente independentes, escritos em linguagem C e C++, porém, fortemente inter-relacionados: o EGESP, um módulo contendo o Método de Gauss para Resolução de Sistemas Lineares com pivoteamento parcial - parâmetro T , o FRED, um módulo contendo o Método Numérico de Runge-Kutta Variante de quarta ordem para Resolução de Equações Diferenciais Ordinárias com m variáveis independentes, o COLISÃO, um módulo para Detecção de Colisões e Dinâmica de Impacto associada, o CONTROLE, um módulo para o controle da animação e o INTEREXPSERV, um interpretador de expressões baseado no modelo cliente-servidor. É importante salientar que a Animação Modelada por Dinâmica requer a presença de ferramentas que suportem cálculos matemáticos intensivos e precisos, pois, constantemente, está sujeita a erros de precisão numérica e grande consumo de CPU. As animações produzidas com o uso da Dinâmica podem ser usadas em vários campos de aplicação: educacional, industrial, Robótica, Propaganda, entre outros.

¹ANIMAção por Dinâmica de Objetos

Abstract

The general aim of this work is to present in details, a research that has been done about “Modeled Animation by Dynamics”, since 1990. The main objective is to introduce the **ANIMADO**², a first example of using Dynamic Simulation in the Computer Animation field, a research branch area of the group ProSim, in the DCA at UNICAMP.

Dynamics refers to the description of movement as a relation between forces and torques acting upon masses. In this way, we use Physics to find out the movement which will be performed by the objects. This movement can be very realistic and depends on the modelling. Using the **ANIMADO**, the animator will be able to produce movements by selecting control forces and torques which will be directly related to mechanical laws of motion.

Generally speaking, the **ANIMADO** is divided into five totally independent, though strongly inter-related modules, implemented using the C and C++ language: the **EGESP**, a module that solves Systems of Linear Equations by standard Gauss-Jordan elimination with maximal pivoting, the **FRED**, a module that contains a 4th variant Runge-Kutta Numerical Integration Method, the **COLISÃO**, a module to Detect and Response collisions, the **CONTROLE**, a module to control the animation, and the **INTEREXPSERV**, an interpreter of expressions based in the client-server model using sockets. It is important to say that the Modeled Animation by Dynamics needs appropriated mathematic tools which avoid inaccurate results and reduce time processing.

The produced animations will be able to be used in various fields of application and with a variety of objectives, as a helpful tool in the educational and industrial area, Robotic Engineering, Propaganda and many others.

²ANIMAção por Dinâmica de Objetos

Agradecimentos

Gostaria de agradecer a todos os que, mesmo silenciosamente, contribuíram positivamente para a realização deste trabalho.

Especialmente ao amigo Falcão, que se mostrou um braço direito invejável na implantação do ANIMADO, sempre bem humorado e dedicado.

Rubens, cujo interesse me ensinou a ser mais cuidadosa durante o desenvolvimento do trabalho.

Armando, Mário, Gorgônio e Cordeiro presentes no laboratório, resolvendo muitos “probleminhas” que fatalmente acabariam se tornando grandes bolas de neve.

Elô, cuja atenção para com os alunos é toda especial.

Sandra, com quem depurei muitos iLa[ica][k].

Jane Wilhelms, cujo contato via e-mail foi muito simpático e incentivador.

Ao professor Léo, cuja animação incentivou a realização deste trabalho.

André, que me ensinou a olhar “grandes mistérios” da vida e do trabalho, como “um conjunto de pequeninos misteriozinhos”, bem menos complicados do que insistimos em acreditar que sejam.

Tina e Terto, pacientes, fiéis e eternos amigos.

E a uns grandes olhos de criança que não sei como, nem por que, se tornaram raros doces cúmplices de outros olhos de criança.

Dedico este trabalho às figuras ímpares de minha vida: Tina e Tertó.

“As coincidências significativas espontâneas, de alto grau de improbabilidade, e que conseqüentemente parecem increditáveis, são pensáveis como puro acaso. Mas quanto mais elas se multiplicam, mais o espaço e o tempo perdem o seu significado, porque em tais circunstâncias, a casualidade que pressupõe o espaço e o tempo, torna-se quase impossível de ser determinada, ou é simplesmente impensável. Por isto, em tais casos, seria inadequado falar em “acazos” já que a magia e a emocionalidade da alma humana constitui (realmente) a causa principal de todas as coisas.”

(C. G. Jung - “Sincronicidade”)

Conteúdo

RESUMO	i
ABSTRACT	ii
AGRADECIMENTOS	iii
CONTEÚDO	ii
LISTA DE FIGURAS	v
1 INTRODUÇÃO GERAL	1
1.1 Objetivos	1
1.2 Motivação do trabalho	2
1.3 Descrição dos capítulos	5
1.4 Terminologia	7
2 A ANIMAÇÃO DE OBJETOS POR COMPUTADOR	9
2.1 Origens	9
2.2 Tipos de Objetos e seus Movimentos	12
2.2.1 Partículas	12
2.2.2 Corpos Rígidos	12
2.2.3 Corpos Flexíveis	14

2.2.4	Corpos Articulados	14
2.2.5	Corpos definidos algebricamente e Corpos em Metamorfose	15
2.3	Análise Mecânica do Movimento	15
2.3.1	A Análise Cinemática	15
2.3.2	A Análise Dinâmica	17
2.3.3	Animação Modelada por Dinâmica X Animação Modelada por Cinemática	18
2.4	Classificação da animação quanto aos Métodos de Controle de Movimento e Tipos de Interação entre Objetos	22
2.4.1	Métodos de Controle de Movimento	22
2.4.2	Tipos de Interação entre Objetos	23
2.5	A Animação Modelada por Dinâmica	24
2.5.1	Um método simplificado para a Animação por Dinâmica	25
2.5.2	Algumas limitações dos Modelos de Animação por Dinâmica	26
2.5.3	Trabalhos existentes na área	27
3	O ANIMADO	35
3.1	Origens	35
3.2	Modelagem do movimento	36
3.2.1	Formulações existentes	37
3.2.2	A formulação utilizada no ANIMADO: o Método de Euler	40
3.3	Etapas da animação utilizando o ANIMADO	50
3.3.1	A fase de inicialização	50
3.3.2	Resolução das equações de movimento: o EGESP	55
3.3.3	Integração das equações de movimento: o FRED	59
3.3.4	Deteção de Colisão e a Dinâmica de Impacto: o COLISÃO	61
3.3.5	Controle das equações de movimento e dos módulos: o CONTROLE	73

3.3.6	Comunicação entre os módulos: o INTEREXPSERV	74
3.3.7	Introdução de restrições físicas no modelo	78
3.3.8	Reprodução da Cinemática	80
3.3.9	O funcionamento do ANIMA _{DO}	80
4	Conclusão	83
4.1	Trabalhos Futuros	87
a	Animações Produzidas	89
b	Descrição funcional do ANIMA_{DO}	93
c	Exemplo de uso (arquivos de descrição ou script)	96
d	Trabalhos existentes no Brasil	125
	BIBLIOGRAFIA	128

Lista de Figuras

1.1	O Sistema ProSIIm.	3
2.1	Restrições cinemáticas no movimento de um chicote e de uma corrente [IC87].	11
2.2	Tipos de objetos existentes e seus graus de liberdade de movimento.	13
2.3	Animação Dinâmica X Animação Cinemática [TT89b].	21
2.4	MCMs e Tipos de Interação entre Objetos [TT91].	24
2.5	Processamento de um programa usando o EXPERTMIRA [TT86].	33
2.6	Cálculos de Interface e lógica (regras e fatos) [TT86].	34
3.1	O tensor de inércia de um objeto descreve a distribuição de massa do objeto. O vetor BP localiza o elemento massa, d_m [CRA89].	46
3.2	Um protótipo de um Sistema de Animação Modelada por Dinâmica.	51
3.3	Deteção do ponto de colisão [HAH88b].	62
3.4	Deteção da aresta de colisão [HAH88b].	63
3.5	Um modelo de colisão entre dois corpos rígidos [MW88].	66
3.6	Um exemplo de <i>bounding box</i>	71
3.7	Esquema geral de comunicação entre portmapper, cliente e servidor [SUN88].	74
3.8	Estabelecimento de uma conexão do tipo <i>stream</i> [SUN88].	76
3.9	A comunicação no ANIMADO.	79
3.10	Esquema de funcionamento do ANIMADO.	81

a.1	Lançamento de uma esfera, colidindo geometricamente com o chão.	90
a.2	Simulação do Sistema Solar.	91
a.3	Colisão de uma esfera com o chão.	91
a.4	Colisão de uma esfera e um cubo.	92
b.1	Comunicação entre os módulos do ANIMADO.	94
c.1	Um exemplo de colisão utilizando solução analítica.	124

Capítulo 1

INTRODUÇÃO GERAL

1.1 Objetivos

O principal objetivo deste trabalho de dissertação de mestrado é o estudo e implementação de um Protótipo de um Sistema de Animação Modelada por Dinâmica, chamado **ANIMADO** (ANIMAção por Dinâmica de Objetos) que introduzirá o uso da Simulação Dinâmica no âmbito da Animação de Objetos sintetizados por computador no **ProSim**¹. Desta forma, o animador produzirá os movimentos desejados selecionando **forças** e **torques** de controle que estarão diretamente relacionados a leis de Mecânica de Movimento.

O **ANIMADO** contém uma biblioteca de ferramentas independentes, suficientes para a composição de Animações Modeladas por Dinâmica. Facilmente extensível e modular, foi idealizado com a intenção de servir como apoio para outras abordagens, tais como a Dinâmica Inversa e a Cinemática Inversa, visto que possui algumas ferramentas para Cálculo Numérico bastante interessantes.

Quanto às funcionalidades previstas para o **ANIMADO**, pretende-se oferecer as seguintes vantagens em relação ao modelo de Animação Modelada por Cinemática:

1. Maior realismo na animação - os objetos atuarão respeitando as leis físicas que regem os seus movimentos. No atual estado em que se encontra o **ANIMADO**, os objetos

¹Prototipação e Síntese de Imagens foto-realistas e animação

sofrerão influência do campo gravitacional. Entretanto, a mesma idéia é válida para o campo magnético, eletrostático, etc., bastando estender a biblioteca de funções que o compõe.

2. O usuário poderá acrescentar forças e/ou torques agindo sobre os objetos continuamente ($F(t), \tau(t)$), a força e o torque variam entre cada intervalo de tempo) ou instantaneamente (F_i, τ_i , a força e o torque permanecem constantes).
3. Interações entre objetos - abordando o estudo de colisões como um processo físico e não meramente geométrico.
4. Introdução e uso de ferramentas numéricas para resolução dos sistemas de equações, aos quais o ANIMADO está sujeito, estando também à disposição do Grupo de Computação de Imagens para cálculos intensivos, com boa precisão numérica.

As animações produzidas, com o auxílio do ANIMADO, poderão ser utilizadas nos mais diversos campos e com finalidades das mais variadas, dentre elas: atuando como ferramenta auxiliadora e estimulante na área educacional, processos industriais ou naturais, engenharia robótica, medicina, esportes, efeitos especiais em filmes, propaganda e publicidade, e até mesmo em coreografias.

O ANIMADO está inserido no projeto ProSlm, atualmente em desenvolvimento no DCA² - FEE³ - UNICAMP, como ilustrado na figura 1.1, cujo propósito é o de tornar-se um sistema de Computação de Imagens de domínio público.

1.2 Motivação do trabalho

Há cerca de quatro anos, surgiu o projeto ProSlm, orientado pelo professor Léo Pini Magalhães. Naquela época, ainda em fase de estruturação, o ProSlm já tinha como objetivo o estudo e a implementação de técnicas de visualização: Raytracing, Scanline, Radiosidade, Câmera Sintética, Modelos de Tonalização, Modelos de Iluminação, Modelos de Textura, produção de efeitos especiais, estudo de fractais, etc. À medida que alguns módulos foram crescendo e aperfeiçoando-se, sentiu-se a necessidade de agrupá-los de uma

²Departamento de Engenharia de Computação e Automação Industrial

³Faculdade de Engenharia Elétrica

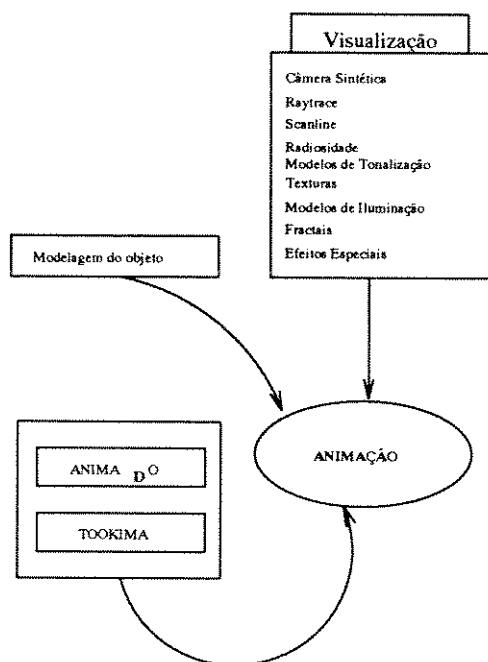


Figura 1.1: O Sistema ProSim.

forma funcional e coerente, dentro de um ambiente de Animação por Computador. Foi então que foram iniciadas as primeiras pesquisas sobre o tema [HOU92]. Como não havia ainda nenhuma ferramenta de animação implementada, era necessário que se começasse do zero. Partindo-se desta idéia, [HOU92] tentou agrupar a maior gama de informações sobre Animação por Computador (de uma forma bastante geral) e, baseado num sistema já implementado [TT85a], especificou o sistema TOOKIMA. Como primeiro protótipo de um sistema de animação por computador, o sistema estava em fase experimental e carecia de testes e correções. Foi então que, durante seis meses, foi testado e corrigido o maior número de funções possível do TOOKIMA. Como resultado deste trabalho, foram constatadas na prática as vantagens e desvantagens existentes entre a Animação Tradicional por Computador (*keyframe*) [ROD92b] e da Animação Modelada por Cinemática [ROD92a].

A partir da constatação de que um sistema de animação por computador deve abordar várias técnicas de animação, partiu-se para o estudo da Animação Modelada por Dinâmica, uma área ainda bastante recente de pesquisa e com vasto campo de aplicação. Aqui, o maior objetivo era a especificação de uma estrutura que comportasse a Animação

Modelada por Dinâmica⁴. Então, foram implementadas algumas situações clássicas da Física, tais como: queda-livre, lançamento oblíquo e pêndulos na solução de sistemas de *edo*⁵. Os dois primeiros modelos também foram implementados via *TOOKIMA*, sujeitos a restrições geométricas e constam do apêndice *a* desta dissertação.

Na Animação Modelada por Dinâmica, as leis físicas que regem o movimento dos corpos são traduzidas em forma de equações que formam um sistema de *edo* que deve ser resolvido para encontrar as acelerações e depois integrado para encontrar as velocidades e posições, durante a animação. Visto isso, era necessário desenvolver algumas ferramentas para a Animação Modelada por Dinâmica cujo desempenho fosse satisfatório, poupando tempo nos cálculos para a animação e com boa precisão numérica. Em especial, foram estudados e implementados Métodos Numéricos de Integração para resolução de *edo* com *m* variáveis independentes⁶ com análise de erro e Métodos de Eliminação para Resolução de Sistemas Lineares⁷ com pivoteamento parcial (parâmetro *T*), usando estrutura de dados para matrizes esparsas. Paralelamente a este trabalho, foi realizado um estudo sobre os tipos existentes de interação entre corpos rígidos (estudo de colisões) num ambiente de animação por computador. Nesse âmbito, foram avaliadas as interações do tipo objeto-ambiente e objeto-objeto, com Detecção de Choques. O estudo dos algoritmos existentes para a Detecção de Colisões entre Objetos (problema cinemático, essencialmente geométrico) fez com que o trabalho de dissertação [MAD93] fosse reavaliado, constatando-se que alguns de seus módulos poderiam ser utilizados para a elaboração de um programa que testa se dois objetos polédricos quaisquer, colidiram ou não. Este programa para a Detecção de Choques foi otimizado utilizando-se *bounding boxes*. O passo seguinte foi a elaboração de um programa que retratasse a situação que ocorre após o choque, um problema associado à Dinâmica de Impacto do choque. Como sugestão dos pesquisadores [MW88], foram utilizadas a inserção de “molas” e resolução analítica de sistemas de equações para a modelagem da Dinâmica de Impacto. Todas estas ferramentas foram agrupadas em bibliotecas de maneira que pudessem ser utilizadas na Animação Modelada por Dinâmica.

Como ilustração do trabalho, foram produzidas três pequenas seqüências de a-

⁴É importante salientar que, neste contexto, o mesmo estudo foi válido para que os modelos de Animação Modelada por Cinemática Inversa e por Dinâmica Inversa também fossem abordados.

⁵Equações Diferenciais Ordinárias.

⁶Runge-Kutta Variante de quarta ordem.

⁷Gauss.

nimação. Duas delas, ilustram o problema de Detecção de Choques e Dinâmica de Impacto Associada, e a outra corresponde a um Modelo de Simulação Gravitacional do nosso sistema solar, que ilustra a dissertação como um todo.

É importante salientar que a experiência ganha com a realização das animações produzidas no decorrer do mestrado, citadas no apêndice *a* desta dissertação, foi fundamental no processo de aprendizado de como se dá, na prática, a produção de uma Animação por Computador.

1.3 Descrição dos capítulos

Este trabalho é composto por quatro capítulos e quatro apêndices. No capítulo 1, são introduzidos, de forma bastante geral, os objetivos e a motivação que originaram o desenvolvimento da dissertação.

No capítulo 2, são discutidas as origens da Animação por Computador, incluindo os tipos de objetos que podem ser animados via computador e seus respectivos movimentos. Em seguida, serão claramente diferenciados os tipos de Análise do Movimento existentes. Primeiramente, considerando-se cada método separadamente, em seguida, comparando-se o Método por Cinemática (usado no **TOOKIMA**) com o Método por Dinâmica (usado no **ANIMADO**), incluindo as vantagens e desvantagens de cada método. Uma vez bem definido o escopo da abordagem por Dinâmica, ela será detalhada e, em seguida, será proposto um método simplificado de funcionamento. Nesse âmbito, serão colocadas algumas das limitações do método e os trabalhos existentes na área, procurando ser o mais abrangente e diversificado possível, incluindo pesquisas recentes na área. Para finalizar, é apresentada uma breve classificação da animação abordada no **ANIMADO** quanto a alguns Métodos de Controle de Movimento e Tipos de Interação entre Objetos de acordo com a classificação proposta por [TT91].

No capítulo 3, é apresentada a especificação do **ANIMADO**, de uma forma bastante detalhada. A partir de algumas considerações iniciais a respeito da proposta envolvida na Animação Modelada por Dinâmica, presente no **ANIMADO**, serão discutidas uma a uma as etapas envolvidas na produção de uma animação utilizando tal sistema. A partir de uma breve discussão sobre o movimento de corpos rígidos sob a influência de forças e torques,

será apresentada a técnica utilizada para o estudo da Dinâmica (método de Euler) e quais variáveis estão envolvidas neste processo (momento de inércia, eixos principais, etc). Em seguida, discutir-se-á o problema da Colisão entre objetos e a Dinâmica de Impacto associada à colisão, bem como o método escolhido para esta abordagem no ANIMADO. Acompanhando estas abordagens, serão apresentadas cinco ferramentas importantes para o bom funcionamento do sistema: o EGESP, uma ferramenta para Resolução de Sistemas Lineares esparsos, o FRED, uma ferramenta para Resolução de Equações Diferenciais Ordinárias, o COLISÃO, que detecta e responde a colisões, o CONTROLE, que literalmente “controla” a comunicação entre os outros módulos, e um INTEREXPSERV, que interpreta expressões, armazena dados e executa a tarefa de comunicação entre os diferentes módulos do sistema, baseado na teoria de cliente-servidor.

No capítulo 4, são apresentadas as conclusões sobre o ANIMADO, finalizando com algumas perspectivas futuras em relação ao tema aqui discutido.

No apêndice *a*, são detalhadas algumas animações produzidas durante o período do desenvolvimento desta dissertação, incluindo três exemplos ilustradores do funcionamento do ANIMADO.

No apêndice *b*, é apresentada a descrição funcional do ANIMADO.

No apêndice *c*, são apresentados alguns exemplos de utilização do sistema (arquivos de descrição).

Finalmente, no apêndice *d*, a partir de uma análise da literatura disponível, consta uma lista dos trabalhos de animação atualmente existentes no Brasil.

1.4 Terminologia

Segue abaixo, uma relação dos termos usados nesta dissertação.

Matrizes

J = matriz do tensor de inércia

${}^B_A R$ = matriz de rotação do SC {B} para o SC {A}

${}^A_B R$ = matriz de rotação do SC {A} para o SC {B}

D_k = matriz de rotação = direção dos co-senos

I = matriz identidade

Vetores

f = força

f_{gr} = força gravitacional

f_{ext} = força externa aplicada

$f_{elastica}$ = força elástica

$f_{amortecimento}$ = força de amortecimento

P = impulso

P_{real} = impulso real

τ = torque

τ_{ext} = torque externo aplicado

p = posição

v = velocidade linear

w = velocidade angular

a = aceleração linear

a_{gr} = aceleração gravitacional

$\dot{\omega}$ = aceleração angular

c = centro de massa do objeto

d = distância

ρ = distância do ponto de colisão ao centro de massa do objeto

d_{limite} = distância limite para ativar a força de amortecimento

pc = ponto de colisão

i, j, k = vetores unitários ortogonais que definem o plano de colisão

Escalares

m = massa

dim = dimensões do objeto

δt = intervalo de tempo

I_{xx}, I_{yy}, I_{zz} = momentos de inércia

I_{xy}, I_{xz}, I_{yz} = produtos de inércia

ϵ = coeficiente de restituição

ϵ_{real} = coeficiente de restituição real

$k_{elastica}$ = constante elástica

$k_{amortecimento}$ = constante de amortecimento

μ = coeficiente de atrito

δ = deslocamento

γ = ângulo de rotação em x

β = ângulo de rotação em y

α = ângulo de rotação em z

Capítulo 2

A ANIMAÇÃO DE OBJETOS POR COMPUTADOR

2.1 Origens

Inicialmente, reportaremos-nos às origens da animação de objetos por computador e a algumas técnicas de modelagem de animação existentes.

Os primeiros modelos de animação por computador trabalhavam somente com corpos meramente geométricos. Este fato foi decorrente da própria origem da animação por computador, derivada diretamente da **animação tradicional**, desenvolvida nos estúdios Walt Disney entre 1925 e 1930.

Como na animação tradicional, os *frames* principais¹ (que compõem toda uma animação) eram criados e os restantes (intermediários) eram interpolados [TJ84], [LAS87], [ROD92b]. Desta forma, gerava-se um conjunto de imagens estáticas que, vistas em sequência, originavam a animação. Esta técnica tradicional de animação por computador conhecida como *keyframing*, é puramente espacial - os modelos representam formas, cores e posições dos objetos que formam uma lista de *frames* para compor a animação [ST91]. Embora seja muito fácil de ser utilizada, à medida que os modelos tornam-se mais complexos, a especificação manual de cada detalhe de cada *frame*, no computador, torna-se tediosa e, às vezes, quase impraticável. Até hoje em dia, os princípios e regras básicas de animação

¹Quadros-chaves da animação

são largamente utilizados [LAS87] no âmbito da animação por computador.

Nos últimos anos, a animação modelada por computador vem se tornando um campo de pesquisa fascinante. “Aqui, o computador é mais do que um suporte, atuando na criação de um mundo 3D” [TT85a]. Uma abordagem típica e simples para o processo de animação por computador consiste na especificação da posição e orientação de uma coordenada geométrica 3D do objeto, como uma função do tempo, sem levar em consideração as **forças** ou **torques** que causaram o seu movimento. Ou seja, os objetos participantes de uma cena são considerados entidades geométricas desprovidas de propriedades físicas. Isto faz com que o animador tenha que usar sua intuição sobre o mundo físico existente, objetivando um maior realismo no movimento dos objetos.

Infelizmente, esta técnica é pouco satisfatória, visto que, movimentos realistas geralmente são complexos de serem representados. Este modelo de animação é conhecido no ambiente de Computação de Imagens como Animação Modelada por Cinemática [HOU92].

Tentando-se amenizar o problema de “intuir” o comportamento físico de um corpo, o animador necessita considerar as entidades participantes da cena de animação como objetos do mundo real, tendo **massa**, **momento de inércia**, **elasticidade**, **coeficiente de fricção**, etc, características estas, advindas da **Dinâmica**.

Alguns sistemas de simulação [HAH88a] são capazes de modelar, realisticamente, a Dinâmica de uma classe geral de **corpos rígidos**² considerando o mundo computacional como um conjunto de massas atuando sob a influência de forças e torques.

Alguns temas muito interessantes de serem abordados em sistemas de Simulação Dinâmica são: **análise de colisão e movimentos articulados**. [WIL87b] trabalhou com um sistema de animação chamado DEVA um sistema gráfico para simulação de movimento de figuras articuladas. Sua abordagem explora o uso de Análise Dinâmica na intenção de gerar animação de corpos articulados.

Dadas as forças e torques atuantes num corpo, a Análise Dinâmica provê as acelerações que as partes dos corpos terão. Por intermédio de Métodos de Integração, as acelerações são usadas para encontrar as novas posições e então, gerar o movimento.

²Definidos como corpos que se movem como uma unidade. O estado dinâmico de um corpo rígido é completamente descrito por sua massa, inércia rotacional, posição, orientação, momento linear e momento angular.

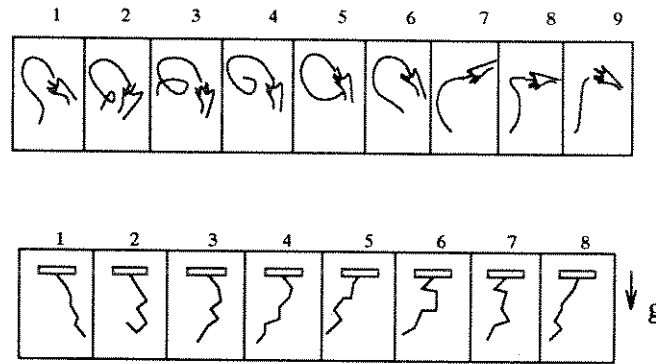


Figura 2.1: Restrições cinemáticas no movimento de um chicote e de uma corrente [IC87].

[IC87] também implementaram um sistema de animação de figuras chamado DYNAMO (DYNAMIC MOTION SYSTEM). O sistema introduz três formas diferentes para o controle do movimento:

- **“Restrições Cinemáticas”**, permitindo que os sistemas de animação tradicional estejam embutidos num ambiente de Análise Dinâmica. Restrições cinemáticas também podem ser impostas em juntas limites em movimentos articulados (por exemplo, não estendendo além de uma certa junta limite, ou permitindo a rotação de um certo segmento somente de 10 graus, durante os próximos 10 segundos, etc), como ilustrado na figura 2.1. Neste exemplo, as rotações da mão são pré-especificadas como restrições cinemáticas e os segmentos que compõem o chicote reagem ao movimento da mão devido às forças de interação (Análise Dinâmica) existentes entre eles. A mesma idéia é válida no movimento da corrente.
- **“Funções Comportamentais”**, que relacionam o estado momentâneo do sistema dinâmico com as forças e acelerações desejadas na figura. Esta habilidade de definir “funções comportamentais” permite que o objeto reaja de acordo com o ambiente em que está posicionado.
- **“Dinâmica Inversa”**, que provê meios para a determinação das forças necessárias para a realização de um movimento específico.

2.2 Tipos de Objetos e seus Movimentos

Os tipos de objetos existentes variam de acordo com sua descrição, o que diretamente influencia o movimento de cada objeto que participa de uma determinada ação.

Tipicamente, o movimento é descrito em termos dos graus de liberdade de cada objeto, ou seja, do **número de coordenadas independentes** necessário para a especificação das posições de todas as componentes do sistema.

Para um sistema de animação com n graus de liberdade, ativo durante f frames, devemos especificar nxf quantidades. Ex.: Supondo uma taxa de exposição de 30 frames por segundo e um tempo total de animação de 3 segundos e que exista um conjunto de objetos que totalizem 50 graus de liberdade. É necessário um total de 4500 valores para a especificação completa do movimento [WIL87a]. Naturalmente, o uso de técnicas tradicionais de animação (*keyframing*) reduz a porcentagem dos números especificados pelo usuário, mas mesmo assim, os dados necessários para o controle do movimento são bastante consideráveis. Alguns tipos de objetos estão ilustrados na figura 2.2 e serão discutidos a seguir.

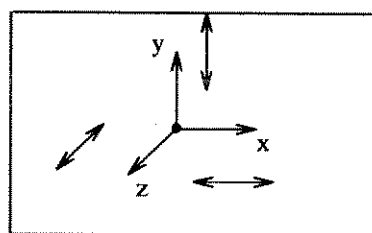
2.2.1 Partículas

Uma partícula pode ser descrita como **um ponto** no espaço $3D(x, y, z)$. A localização e o movimento deste ponto é designado por três variáveis, ou seja, o ponto tem três graus de liberdade de movimento. A animação deste ponto requererá uma tripla de números para cada *frame* da animação ou três funções descritoras da variação de x , y e z no tempo.

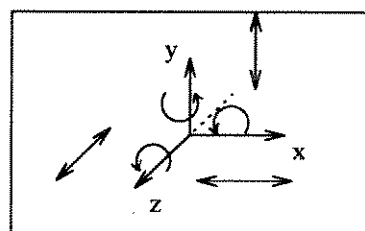
[REE83], [RB85] e outros autores têm usado partículas na simulação de fenômenos naturais.

2.2.2 Corpos Rígidos

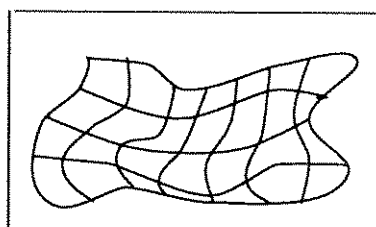
Um corpo rígido é definido por um número de pontos que se movem como uma unidade. Eles não podem se mover relativamente a cada um deles (por exemplo, ser



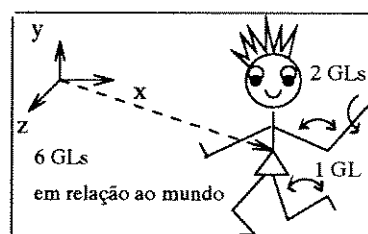
Partículas
(3 graus de liberdade)



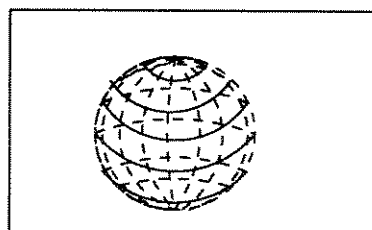
Corpos Rígidos
(6 graus de liberdade)



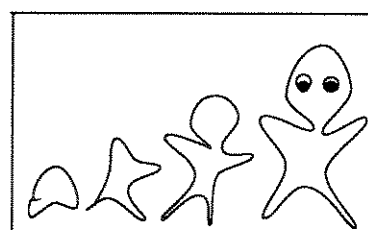
Corpos Flexíveis
(3 a 6 graus de liberdade)



Corpos Articulados
(Graus de liberdade variados)



Corpos definidos algebricamente



Corpos em Metamorfose

Figura 2.2: Tipos de objetos existentes e seus graus de liberdade de movimento.

deformado), entretanto, podem se mover conjuntamente, em relação ao espaço mundo.

Estes pontos podem definir um polígono ou uma superfície de forma livre. O movimento é especificado por seis graus de liberdade: x , y e z translações e x , y e z rotações (orientação). Este tipo de movimento é frequentemente visualizado como o movimento de um SC fixo ao corpo rígido, com todos os pontos que definem o corpo movendo-se com o SC. A maioria dos sistemas de animações concentram-se neste tipo de modelagem de objetos.

2.2.3 Corpos Flexíveis

Um corpo flexível consiste num número infinito de pontos que se movem relativamente uns aos outros, com o passar do tempo. Na prática, pode ser simplesmente definido como “um conjunto de pontos que se movem”. Estes pontos podem ser representados por vértices de polígonos, mas conseguimos maior flexibilidade se os representarmos como pontos de controle da superfície [WEI86]. Ex.: Um corpo flexível em movimento, por exemplo uma ameba, é definido por uma superfície de forma livre usando p pontos de controle, tem $3xp$ graus de liberdade variando com o tempo. Isto explica o fato da maioria dos sistemas de animação existentes utilizar corpos rígidos como personagens da animação. O estudo de corpos flexíveis é uma área recente de pesquisa e que promete muito no campo da animação.

2.2.4 Corpos Articulados

Corpos articulados são compostos por segmentos cujo movimento relativo à cada parte componente é um pouco restrito. Ex.: Um corpo humano é frequentemente representado por segmentos rígidos ligados por articulações (juntas) que têm de um a três graus de liberdade. O número total de graus de liberdade que deve ser especificado é a soma dos graus de liberdade de cada junta [ZEL82], [BOK80] e [BAD87].

Corpos articulados também podem ser flexíveis. Neste caso, a junta entre dois segmentos flexíveis pode ser modelada como uma junta entre dois SC coordenados, um atracado a cada segmento adjacente. Dois tipos de movimento são possíveis: os movimentos nas articulações (consiste na mudança de seis números que especificam a relação entre os SC) e o movimento dentro de cada segmento flexível (consiste no movimento de pontos que definem o segmento relativo ao seu SC local). O número total de graus de liberdade

necessário para a especificação de um corpo flexível articulado é o número de graus de liberdade em cada junta mais três vezes o número de pontos definidos.

2.2.5 Corpos definidos algebricamente e Corpos em Metamorfose

Alguns corpos podem ser especificados algebricamente. Uma esfera pode ser modelada como um centro (três números) e um raio, e uma esfera “pulsante” pode ser animada somente mudando-se a dimensão do raio. O número de graus de liberdade do sistema dependerá da natureza das equações que definem estes objetos algébricos.

A complexidade do sistema aumenta à medida que os objetos modelados vão tornando-se mais sofisticados. Um corpo que muda de forma durante a animação está mudando o seu número de graus de liberdade com o passar do tempo. Alguns exemplos deste tipo de corpos são montanhas em formação [FFC82], crescimento de plantas [KAW82], [SMI84] e sistemas de partículas [REE83] utilizando fractais³.

2.3 Análise Mecânica do Movimento

Nesta seção serão discutidas as diferenças entre os dois tipos de Análise Mecânica do Movimento existentes: a **Cinemática** e a **Dinâmica**, ambas aplicadas à animação.

2.3.1 A Análise Cinemática

A Análise Cinemática descreve quantitativamente a geometria de um movimento em relação a uma variável, por exemplo, o tempo (posições e velocidades de pontos). Há dois tipos de Análise Cinemática: a **Direta** e a **Inversa**.

³O conceito de fractal é ainda uma questão em aberto e a discussão em torno do assunto ainda não chegou a um consenso geral, mas segundo Mandelbrot, “Fractais são curvas cuja dimensão é estritamente maior que sua dimensão Euclidiana” [STA93].

2.3.1.1 A Cinemática Direta

A Cinemática Direta envolve a especificação de uma seqüência de posições em cada intervalo de tempo da animação. Um exemplo de aplicação da Cinemática Direta é a animação *keyframe*. Este tipo de animação envolve uma especificação incompleta do movimento, bem como a subsequente determinação matemática do movimento completo. O maior problema desta abordagem é o tempo gasto para a descrição do movimento, com o intuito de dar realismo à animação. A qualidade da animação dependerá muito mais da habilidade do animador do que da qualidade do sistema de animação. [HOU92] abordou a Análise Cinemática de movimento no sistema de animação chamado **TOOKIMA**⁴. Uma aplicação desta análise pode ser vista na animação realizada no trabalho [ROD92a] que objetivou agrupar o maior número de funções de movimento que exemplificassem a Análise Cinemática Direta.

Uma descrição bastante simples deste tipo de animação seria: “Existe um corpo posicionado na origem no tempo igual a zero. Ele move-se com uma aceleração constante na direção (2,-2,3)”. Por intermédio de transformações afins básicas (translação, rotação e escala), os objetos vão adquirindo movimento. Maiores detalhes podem ser vistos na citação anterior.

2.3.1.2 A Cinemática Inversa

A Cinemática Inversa, muito usada em Robótica, também requer a especificação de uma seqüência temporal de posições. Contudo, este tipo de análise somente necessita da especificação dos pontos extremos a serem atingidos. As demais posições são calculadas baseadas na localização destes pontos extremos. Por exemplo, a localização especificada para a mão determina a orientação do braço. Isto reduz a tarefa computacional do animador, mas requer a existência de restrições internas (para uma dada posição da mão) com o objetivo de obter uma única orientação do corpo.

Um exemplo simples que utiliza esta análise é o seguinte: “qual deve ser a velocidade inicial (constante) do objeto para que ele atinja a posição (10, 20, 35) em 10 segundos?”. Ela envolve dois problemas intimamente relacionados:

⁴A TOOL Kit for Modeled Animation.

- Encontrar uma solução que atinja a meta desejada,
- Encontrar a melhor solução entre as factíveis.

Para um corpo complexo, tal como uma figura humana, o uso da Cinemática Inversa tende a tornar-se extremamente problemático, pois a vantagem de sua utilização é inversamente proporcional ao aumento do número de graus de liberdade do objeto.

2.3.2 A Análise Dinâmica

A Análise Dinâmica assume a existência de uma relação causal entre a força produzida e o correspondente comportamento cinemático do objeto. Este tipo de análise requer a derivação das equações de movimento que representam uma descrição matemática do modelo dinâmico. Há dois tipos de Análise Dinâmica: a **Dinâmica Direta** e a **Dinâmica Inversa**.

2.3.2.1 A Dinâmica Direta

Na Dinâmica Direta, calcula-se um conjunto de forças e torques a partir de equações algébricas de movimento. Portanto, levamos em consideração as leis físicas que regem o movimento. Um objeto com massa m move-se com uma aceleração proporcional às forças que nele atuam. “No tempo inicial igual a zero, o objeto está posicionado em (0, 5, 10) e possui massa igual a 20 kg . Ele está sujeito apenas à força gravitacional.” Este exemplo de abordagem de Simulação⁵ nada mais é do que o movimento de queda-livre.

A análise é centrada nestas forças ou nas transformações matemáticas das forças (trabalho ou potência). A Dinâmica Direta envolve o cálculo do movimento baseado na especificação de forças e técnicas de integração das equações de movimento. Este tipo de Análise Dinâmica é comumente referenciado como **Simulação Dinâmica**.

⁵A Simulação pode ser baseada em soluções analíticas explícitas de equações de movimento ou em Métodos Numéricos de Integração para resolução de equações diferenciais que representam uma lei natural.

2.3.2.2 A Dinâmica Inversa

A Dinâmica Inversa pode ser usada indiretamente com o objetivo de melhorar a entrada de forças numa simulação dinâmica. Isto sugere que a Dinâmica Inversa tenha uma aplicação um pouco mais direta na animação onde a produção do movimento é o **objetivo**, e não a análise da **causa** do movimento. Neste caso, dados o movimento do objeto (aceleração) e sua massa, devemos descobrir “quais forças devem ser aplicadas para que ele atinja uma certa posição (15, -10, 12) em 10 segundos ?” Esta técnica é muito usada no movimento de figuras articuladas.

Enfim, a compreensão dos diferentes tipos de Análise Mecânica do Movimento é de vital importância no âmbito da produção de animações por computador. Um exemplo bastante conhecido de aplicação desta análise é o filme: “2001: Uma Odisséia no Espaço”.

2.3.3 Animação Modelada por Dinâmica X Animação Modelada por Cinemática

Nesta seção, serão comparadas as técnicas de Animação Modelada por Dinâmica e por Cinemática com o objetivo de salientar as vantagens e desvantagens existentes em cada um dos métodos.

Primeiramente é interessante relembrar que a Animação por Computador é definida por um conjunto de parâmetros que descrevem os seguintes aspectos:

- a estrutura da cena, os objetos e seus atributos (cor, textura, posições, etc),
- a posição do observador, posição da câmera e seu estado local,
- a posição das fontes de luz e seus atributos (intensidade, localização, etc).

Todos estes parâmetros devem ser **sincronizados** de tal forma que produzam a animação desejada. A animação resultante dependerá intimamente do Método de Análise Mecânica utilizado (citado na seção 2.3) para gerar o movimento.

Basicamente, há quatro tipos de Análise Mecânica de Movimento e, de uma forma abrangente, os modelos por Dinâmica e os modelos por Cinemática. A Animação

Modelada por Cinemática produz movimentos a partir de velocidades e posições, enquanto que, a Animação Modelada por Dinâmica descreve o movimento por intermédio de um conjunto de forças e torques de onde os dados cinemáticos são derivados.

O método de controle de movimento mais frequentemente utilizado em animação é o cinemático porque é mais intuitivo e consome menos tempo de CPU. A vantagem sobre outros métodos está no controle total de todas as posições e orientações ocupadas pelos pontos no decorrer do tempo. Por outro lado, movimentos rápidos e complexos podem ser perfeitamente abordados utilizando-se o método de controle por Dinâmica. Sendo assim, é natural que cada um destes métodos possua suas vantagens e desvantagens que serão salientadas a seguir.

Segundo [TT89b], na Animação Modelada por Cinemática, a qualidade final da animação depende das leis que regem o movimento e, este tipo de animação, frequentemente produz resultados pouco realistas. O tempo de CPU gasto para a geração da animação também é dependente das leis de movimento envolvidas no processo, mas geralmente esta abordagem não é muito custosa. Quanto à intervenção humana no processo de animação, ela é muito grande, podendo sofrer algumas dificuldades. Em geral, a versatilidade deste tipo de animação é muito boa, porém, a principal dificuldade desta abordagem está em se descobrir leis realistas capazes de realizar o movimento desejado. Infelizmente, elas não são fáceis de serem encontradas dependendo da habilidade, criatividade e paciência do animador. Estes sistemas forçam o usuário a escolher o movimento desejado a partir de uma gama muito grande de movimentos variados. Por exemplo, os movimentos de alguns seres vivos que habitam em nosso mundo podem ser analisados e “copiados” no computador, guardando-se as posições assumidas por todos os seres em cada *frame*. Estas posições serão utilizadas para direcionar a figura que está sendo animada. Como pode ser observado, esta abordagem possui algumas dificuldades e, entre elas, o fato do movimento ser altamente especificado pelo animador, não podendo ser alterado sem impunidade, nem tão pouco permite a descrição de animais e/ou condições imaginárias (por exemplo, simular corpos caindo na Terra sem a presença do campo gravitacional).

Uma alternativa é dar aos seres vivos a serem animados uma realidade física um pouco maior, por meio da descrição de massas, forças e torques ativos, usando a Análise Dinâmica para diagnosticar o movimento. A Análise Dinâmica também é usada em

aplicações CAD/CAM e em Robótica (na simulação e controle de veículos ou manipuladores mecânicos), porém também possui algumas dificuldades que serão discutidas no item 2.5.3 deste trabalho: custo, instabilidade numérica (dependendo da complexidade do objeto e do movimento) e controle do movimento.

Na Animação Modelada por Dinâmica, a qualidade da animação é bastante realista, porém, os movimentos são muito regulares (não levam em consideração a personalidade e a individualidade de cada objeto). O tempo de CPU gasto e o custo para a sua geração são bastante elevados e aumentam de acordo com a complexidade da animação desejada. O maior problema desta abordagem é decorrente da não-existência de soluções analíticas para o problema da animação e, devido aos numerosos graus de liberdade que podem existir num mesmo objeto, o sistema de equações diferenciais não-lineares torna-se muito grande. Mesmo utilizando Métodos Numéricos Recursivos de Otimização [AG85b], o custo computacional continua alto, sem contar a possibilidade da existência de muita instabilidade numérica. A intervenção humana neste tipo de processo de animação pode ser limitada, o animador apenas inicia o processo de animação setando as condições iniciais para a animação. Em geral, a versatilidade é boa e as maiores dificuldades encontradas são consequência direta da não-trivialidade em se modelar o mundo físico existente pois as forças e torques (grandezas pouco intuitivas de serem estimadas, em N^6 e $N.m^7$) devem ser especificadas como dados de entrada para iniciar e gerar o movimento. Estas considerações estão sucintamente evidenciadas na figura 2.3.

A simulação de movimentos de corpos articulados, usando a modelagem por Dinâmica, é trabalhosa devido aos muitos graus de liberdade presentes nas articulações, à possibilidade da existência de movimentos complexos coordenados e também decorrente da nossa pré-disposição em não aceitar visualmente movimentos pouco realistas de corpos que nos são tão familiares, tais como, seres humanos e animais. As duas últimas considerações também são válidas na animação de corpos mais simples.

Enquanto na modelagem por Cinemática os movimentos são definidos por posições no espaço e o animador deve ajustar parâmetros até que o movimento pareça correto, na modelagem por Dinâmica, o movimento parecerá perfeitamente real (se bem modelado), mas o animador terá que experimentá-lo até obter o efeito desejado.

⁶Newton, unidade de medida de força

⁷Newton.metro, unidade de medida de torque

	Qualidade da animação	Tempo de CPU	Intervenção Humana	Versatilidade	Dificuldades
CINEMÁTICA	<i>dependerá das leis que regem o movimento, mas freqüentemente é pouco realista.</i>	<i>depende das leis que regem o movimento, mas freqüentemente é não-custosa.</i>	<i>muito dependente das leis de animação, podendo sofrer algumas dificuldades</i>	<i>muito boa</i>	<i>leis realistas não fáceis de serem encontradas</i>
DINÂMICA	<i>muito realista . se bem modelada</i>	<i>muito cara</i>	<i>pode ser limitada</i>	<i>boa</i>	<i>modelos completamente baseados em dinâmica são muito caros, especialmente em seqüências grandes</i>

Figura 2.3: Animação Dinâmica X Animação Cinemática [TT89b].

O ideal para um sistema de animação modelada é a integração de várias técnicas. Nos trabalhos [WB85] e [WIL86] foi criado um sistema de animação chamado VYRIA que suporta ambas as técnicas combinadas. É importante ressaltar que nenhum método é superior a outro. Em particular, “a maioria dos métodos que possuem baixo tempo de CPU não produzem animações realistas” [TT89b].

Para finalizar, pretendemos estudar e implementar um primeiro protótipo de um Sistema de Animação Modelada por Dinâmica chamado ANIMA_{DO}.

“A Simulação é geralmente definida como o processo do estudo do comportamento de um sistema físico sob várias condições, usando um modelo computacional. Em Computação de Imagens, o termo “Simulação” é frequentemente usado quando um sistema é descrito por leis físicas que nele atuam (Dinâmica) e o termo “Animação” é frequentemente associado ao termo Cinemática” [BRU90].

Nossa intenção é que o termo “Simulação” seja tratado como uma técnica de animação particular, onde o controle do movimento é determinado por **equações dinâmicas de movimento**, e não explicitamente pelo animador, como no caso cinemático.

2.4 Classificação da animação quanto aos Métodos de Controle de Movimento e Tipos de Interação entre Objetos

[TT91] propõe uma nova classificação para os modelos de Animação por Computador que depende dos Métodos de Controle de Movimento e dos Tipos de Interação entre Objetos.

2.4.1 Métodos de Controle de Movimento

“Um Método de Controle de Movimento (MCM) especifica como um objeto é animado. A caracterização de um MCM é baseada no tipo de informação privilegiada na animação”. Exemplificando, num sistema *keyframe* (para um corpo articulado), a informação privilegiada é o **ângulo** e, num sistema modelado por Dinâmica, são as **forças e torques**. Segundo os autores, podemos distinguir três categorias de MCMs: **geométrico, físico e comportamental**.

2.4.1.1 MCMs Geométricos

O movimento é definido por meio de coordenadas, ângulos e outras características dependentes da forma do objeto. As informações privilegiadas são de natureza geométrica.

2.4.1.2 MCMs Físicos

O movimento é definido por meio de leis físicas. As informações privilegiadas são: **massa, momento de inércia, rigidez**, etc. Ex.: movimento de balanço.

2.4.1.3 MCMs Comportamentais

O movimento é definido de acordo com o **comportamento** da figura (modo pelo qual humanos e animais agem). Por exemplo, uma figura sorrindo para uma outra figura. As informações privilegiadas são os parâmetros para a descrição do comportamento

(personalidade das figuras: mudança no humor, emoção e estilo), por intermédio de uma linguagem natural. Neste MCM, [TT91] estão considerando a **semântica** e a **sintaxe** do comportamento.

2.4.2 Tipos de Interação entre Objetos

Os tipos de interação entre objetos, ou seja, a **relação entre o objeto e o resto do mundo**, pode ser dividida em quatro casos: um único objeto, objeto-ambiente, objeto-objeto e objeto-animador.

2.4.2.1 Um único objeto

O objeto sintetizado está sozinho na cena e não interage com outros objetos. Somente o animador pode prever colisões pré-calculando as trajetórias apropriadas.

2.4.2.2 Objeto-Ambiente

O objeto sintetizado está em movimento num ambiente e consciente deste ambiente (Simulação Dinâmica). Ex.: colisão (MCM Físico).

2.4.2.3 Objeto-objeto

As ações realizadas por um objeto são conhecidas por outro objeto e podem influenciar o comportamento deste segundo objeto. Isto significa que ocorre algum tipo de **comunicação** entre os objetos (respostas $\longleftrightarrow$ estímulos).

2.4.2.4 Objeto-Animador

O animador pode comunicar-se com o objeto e vice-versa, em alguns casos, em tempo real. Ou seja, o animador ordena e a figura responde. A tabela ilustrada pela figura 2.4 compara concisamente, a relação entre os MCMs e os Tipos de Interação entre Objetos.

	MCMs GEOMÉTRICOS	MCMs FÍSICOS	MCMs COMPORTAMENTAIS
1 ÚNICO OBJETO	Rotoscopia Keyframe Restrições Cinemáticas	Dinâmica	Modelo de cansaço muscular Sistema de codificação da ação facial
OBJETO-AMBIENTE	Desvio de obstáculos Detecção de intersecção	Modelos de Colisão Modelos deformáveis	Sensores Desvio de obstáculos baseados na visão
OBJETO-OBJETO	Colisões geométricas	Colisões Dinâmicas	Comunicação de emoções entre figuras
OBJETO-ANIMADOR	Desenho de trajetórias	Entrada de forças	Entrada visual utilizando uma câmera de vídeo

Figura 2.4: MCMs e Tipos de Interação entre Objetos [TT91].

Respeitando a classificação proposta por [TT91], o objetivo desta dissertação será o de estudar os **MCMs Físicos** cuja interação seja do tipo: um único objeto, objeto-ambiente e objeto-objeto, e os **MCMs Geométricos** cujas interações sejam do tipo: objeto-ambiente e objeto-objeto. Sendo que os objetos considerados são corpos rígidos, sem nenhum grau de deformação. Neste âmbito, pretende-se abordar: detecção de colisões (detecção de intersecções) e respostas à colisão (modelos de colisão).

2.5 A Animação Modelada por Dinâmica

Nesta seção serão abordados mais profundamente os conceitos que estão embutidos dentro do escopo da Animação Modelada por Dinâmica.

Como já mencionado no item 2.3.2, a Dinâmica refere-se à descrição do movimento como uma relação entre **forças** e **torques** atuando sobre **massas**. Assim, podemos usar a física para definir o movimento que essas massas irão sofrer. Este movimento, se bem modelado, imita o que realmente acontece a tais massas no mundo real. “Portanto, a Dinâmica **simula** o movimento ao invés de somente animá-lo” [BRU90]. Na Animação por Dinâmica, “o que acontece **entre cada frame** é mais importante do que o que acontece **em cada frame**” [KOC90].

O uso da Dinâmica é importante por uma série de razões: alguns movimentos complexos podem ser gerados com o mínimo de dados do usuário (por exemplo, movimentos

devido à gravidade), pode impor vários tipos de restrições (impedindo a intersecção de corpos colidentes), pode alcançar um maior realismo e naturalidade no movimento de corpos articulados mais complexos (movimento humano), desde que seja modelável etc.

O problema do controle da animação por Dinâmica, no caso das figuras articuladas, é encontrar os torques que serão aplicados nas juntas visando o movimento desejado. Vários métodos têm surgido e alguns deles sugerem que o animador especifique os torques diretamente, no caso, utilizando a Dinâmica. Outra alternativa é o uso da Dinâmica Inversa para descobrir quais serão os torques necessários para a produção de uma aceleração conhecida. Uma solução bastante interessante para o problema utiliza controladores de espaço-estado [PFZ90].

2.5.1 Um método simplificado para a Animação por Dinâmica

O primeiro passo para a construção de um modelo de animação por Dinâmica é construir um modelo físico por meio de equações diferenciais. Neste modelo físico, cada objeto participante da animação será considerado como um corpo rígido, cujo movimento será conveniente para efeitos de inércia, forças e torques que nele atuarão durante a animação. Muitas das forças e torques aplicados serão externos, outros serão derivados de restrições geométricas, que poderão ser aplicadas ao modelo.

Utilizando-se de **Métodos Numéricos de Integração**⁸ para a resolução das equações diferenciais, obtemos as posições e velocidades em cada instante de tempo da animação. Estes dados poderão ser utilizados pelo TOOKIMA para mover o objeto para uma determinada posição referente ao modelo simulado. O TOOKIMA também poderá ser utilizado para definir alguns parâmetros da animação referentes à câmera, à iluminação, etc.

Caso o modelo esteja correto e não haja problemas de instabilidade numérica, a animação retratará com fidelidade o processo físico.

Infelizmente, este processo de animação por computador requer que o animador saiba **modelar** o movimento físico desejado, bem como **programar** o modelo. Isto não significa que o animador tradicional deixe de ser importante no processo de animação, muito

⁸Por exemplo, Runge-Kutta Variante de quarta ordem

pelo contrário, o ideal é que haja um grupo de animação, cada um com suas atribuições específicas, porém inter-relacionadas, cujo interesse final é comum: o processo de animação.

Várias formulações da Mecânica Clássica [GOL50] têm sido desenvolvidas com o objetivo de representar uma lei de evolução de um sistema físico através de um conjunto de soluções de um sistema de equações diferenciais. Como já foi dito, as soluções destas equações podem ser encontradas por intermédio de Métodos Numéricos de Integração específicos que serão usadas para a derivação da seqüência de animação.

Muitos destes modelos são extremamente complexos (modelagem de músculos) pois as equações diferenciais envolvidas nesta abordagem podem conter centenas de variáveis inter-relacionadas, dificultando e muito, a convergência da solução.

De um modo geral, a maioria dos pesquisadores, dentre eles, [AG85b], [WB85] e [GM85] têm modelado corpos rígidos articulados simples (com poucas juntas) e corpos geométricos cilíndricos e esféricos, sem qualquer grau de deformação.

Somente pouquíssimas seqüências de animação têm sido produzidas devido à complexidade da especificação deste tipo de movimento e por causa do tempo de CPU exigido.

No caso da Simulação de Movimentos Humanos, [BB88], [WIL87b], [WK88] e [TT91] concordam que a naturalidade pode ser obtida desde que se considere simultaneamente a **geometria**, a **física**, e o **comportamento individual** de cada personagem. Contudo, isto não é tarefa nada fácil de ser concretizada.

2.5.2 Algumas limitações dos Modelos de Animação por Dinâmica

Três grandes problemas que os pesquisadores de animação por computador têm enfrentado na abordagem por Dinâmica serão enumerados a seguir.

- **Difíceis de serem implementados**

Geralmente, os gastos para a construção de um modelo físico específico é elevado e, na maioria das vezes, o modelo não é reaproveitável.

- **Difíceis de serem controlados**

O comportamento do modelo é frequentemente determinado indiretamente a partir de parâmetros não muito intuitivos (especificação de forças ou torques que são grandezas newtonianas difíceis de serem estimadas no modelo), além de estarem frequentemente sujeitos a problemas de instabilidade numérica.

– **Podem ser lentos e custosos**

Alguns modelos são computacionalmente complexos, levando o sistema a cálculos intensivos.

2.5.3 Trabalhos existentes na área

Nos últimos anos, tem havido um aumento de interesse no estudo de Simulação Dinâmica em ambientes de animação computacional.

No movimento de corpos rígidos, a ênfase tem sido na simulação de figuras articuladas. [AGL87] e [WB85] têm usado estratégias similares incorporando controle cinemático em algumas articulações. [WK88] e outros autores têm investigado restrições específicas, puramente geométricas, direcionadas a algumas classes de problemas de restrição. [BB87] vêm usando uma estratégia similar, porém, no contexto das Simulações Dinâmicas.

Alguns pesquisadores vêm dando maior ênfase ao movimento de corpos humanos, uma complicada tarefa no campo da animação. “O corpo humano tem forma irregular e é difícil de ser modelado” [TPBF87]. “A especificação e cálculo dos movimentos humanos não são nada triviais já que suas complicadas articulações geralmente envolvem mais do que 200 graus de liberdade⁹” [ZEL82]. Além disso, o movimento humano é intuitivo. “É praticamente impossível se considerar o movimento de todas as juntas que envolvem um movimento bastante simples, seu balanço natural e suas interações com o ambiente, porque na realidade, a coordenação destes objetos ocorre inconscientemente” [BAD89]. Sem contar a diferença existente entre duas pessoas que possuem aspectos físicos bastante semelhantes porém, com personalidades completamente diferentes. Neste contexto, [BL86] vem estudando um método de animação realista do corpo humano objetivando encontrar expressões computadorizadas que expressem a personalidade humana.

⁹Correspondem ao número de parâmetros independentes (não incluindo o tempo), necessários para se especificar completamente a posição de cada parte do sistema.

Quanto às principais aplicações da animação do corpo humano, podemos salientar:

1. **a produção de filmes de diversão/cartoons** (“HUNGER” - National Film Board Canada 1974) e “Rendez Vous a Montreal” - MIRA Lab 1987), didáticos [HER82] ou publicitários (comerciais ou políticos).
2. **em simulação de situações onde os seres humanos estão envolvidos** : processos industriais ou naturais, engenharia (robótica, detecção de colisão, simulação de choques), medicina, coreografia, etc.
 - **Aeronáutica/Eng. Automobilística** Simulação de Choques - [WIL82], [FET82], Detecção de Colisão - [BAD80], [DOO82].
 - **Robótica** Visualização de Mecanismos ou robôs em movimento.
 - **Medicina** Reconstrução do cérebro humano - [BRE84].
 - **Coreografia** Simulação de ballets - [HER82].

Os objetivos destas aplicações são completamente diferentes: No primeiro caso, a qualidade artística dos *frames* é extremamente importante (a animação é pré-designada pelo animador) e no segundo caso, as ações não são designadas previamente pelo animador, mas por **equações**.

[GQB89] pesquisaram a Antropometria¹⁰ para a animação de figuras humanas. Discutem os conceitos necessários para o uso de modelos computacionais de animação humana em engenharia, incluindo a análise a respeito de juntas, juntas limites, massa, momento de inércia e volume das figuras.

Segundo [BAD87], “o movimento é a essência da comunicação”. É fácil observar que na animação convencional, a fidelidade do movimento é muito mais importante do que a fidelidade da forma; no âmbito da expressão, humor e caráter do personagem. [BAD87] descreveu um sistema de posicionamento interativo que permite a inclusão de restrições cinemáticas múltiplas nas juntas de uma figura e a definição de uma posição global do corpo, computada por um processo iterativo. Entretanto, nenhuma animação é envolvida

¹⁰ “A Ciência da medida do corpo humano”.

e o método pode ser usado para descobrir posições complexas num contexto tradicional de posicionamento dos *frames*. Com o controle dinâmico, o corpo é manipulado por forças ao invés de posições (na modelagem dos músculos internos relacionados aos torques e à resistência ao movimento). Neste contexto, os segmentos que compõem o corpo possuem massa, e as juntas possuem molas para a simulação de saltos.

[WIL87b] descreveu um sistema que controla a animação de figuras articuladas usando controle de forças em conjunto com certas restrições espaciais, tais como a presença do chão. Ela também abordou a interação entre controle de movimento dinâmico e cinemático.

[AGL87] simplificaram algumas suposições sobre a estrutura da figura a ser animada e produziram um algoritmo dinâmico quase em tempo-real. Eles discutem algumas destas questões complexas do uso da Dinâmica, no âmbito da animação.

[GM85] combinaram o posicionamento cinemático das pernas das figuras com o controle dinâmico de todo o movimento e criaram um sistema chamado PODA que facilita a animação de uma figura bípede ou múltipede. O usuário pode definir o caminho e o tipo de locomoção que desejar e, em seguida, combinar Dinâmica, Cinemática e Interpolação com o intuito de alcançar o tão almejado objetivo: as posições dos objetos, *frame-a-frame*.

[BAD89] trabalhou com Inteligência Artificial, Linguagem Natural e Simulação de Animação Humana. Sua preocupação principal foi direcionada ao processo de comunicação animador-computador. Visto que a “Animação é um meio de comunicação, ela deve ser compreensível, não-ambígua, auto-contida, convincente e fácil de ser criada”. A produção de uma animação sem a presença de um *expert* em animação requer um sistema por computador que compreenda tarefas, movimento e suas “semânticas”. Resumindo, um *expert* sintético. A intenção [BAD89] é estender as capacidades dos engenheiros de projeto, analistas de objetos humanos, ou até mesmo o usuário do sistema de animação para criar, animar e calcular as performances humanas.

Mais especialmente no ambiente de engenharia do que no artístico, os usuários necessitam de um projeto de análise de movimento efetivo e de ferramentas de análise sem que se sintam pressionados com o processo de produção da animação. Assim, foi criado em 1988, um sistema chamado JACK que engloba Inteligência Artificial, construção de ações humanas e modelos de performance, provendo ferramentas analíticas e oferecendo

interações gráficas e linguagem natural ao usuário. Abrange tanto a Análise de Movimento por Cinemática quanto por Dinâmica. No âmbito da Dinâmica, tem a capacidade de especificar forças e torques internos ou externos e aplicá-los a figuras articuladas para movimentá-las. O controle dinâmico é muito usado em movimentos rápidos e respostas a forças externas, por exemplo, devido à gravidade. *Experts* em animação podem reproduzir a ilusão da naturalidade despendendo grande esforço manual. O objetivo do sistema JACK é tornar o processo de animação mais direto, dando comandos para uma figura sintética modestamente inteligente, ou no mínimo, obediente.

Pesquisadores como [ZEL82], [CAL90], [TT89a], têm direcionado esforços no campo da animação humana por intermédio da simulação de alto-nível, através de representações do **comportamento** de um modo quase-humano.

[ADH⁺89] e [TT89a] discutem as vantagens e desvantagens do controle do movimento por Cinemática e Dinâmica no contexto dos movimentos de passeio (Dinâmica) e atracamento de objetos (Cinemática Inversa). Comentam o processo da escrita de uma simples carta, um processo puramente físico. Alguns problemas sobre deformações de superfícies também são mencionados. [WEI86] tem simulado o movimento de tecidos, [TPBF87] e [HAU87] têm estudado a Dinâmica de objetos flexíveis e a modelagem de suas deformações.

Também no âmbito da Simulação Humana, agora no campo das Expressões Faciais, [GUE89] descreve um sistema com facilidades interativas para atracar músculos e rugas (vincos) a malhas faciais retangulares arbitrárias e então, simular as contrações destes músculos. Técnicas de análise estrutural são usadas para dar maior elasticidade ao movimento, resolvendo com maior eficiência as equações especificadas para a animação facial. Segundo o autor, cada ação muscular facial abstrata teve um procedimento associado a 24 parâmetros (usados para o controle preciso dos músculos). Como se pode concluir, a desvantagem desta abordagem é o grande número de parâmetros que devem ser especificados.

[TT89a] apresentam as técnicas mais importantes de animação facial existentes. Vários destes modelos são baseados em músculos e por consequência, eles introduzem alguns conceitos gerais sobre a estrutura facial. Enfatizam as vantagens e desvantagens da separação dos **parâmetros faciais das expressões** e da **animação facial**. Discutem também, a geração de expressões decorrentes de fonemas e emoções que são interpretados

(simulados) pelos personagens.

A animação realista da face humana é extremamente difícil de ser representada pelo computador. Os principais problemas são:

- **A própria representação da face:** É muito difícil modelar uma imagem realística da face, principalmente modelar uma textura de pele que pareça ser natural.
- **A modelagem do movimento:** É muito complexa, por causa dos numerosos músculos específicos que a face possui e de suas interações com a estrutura óssea. Esta complexidade faz parte do **estudo das expressões faciais**. Segundo [TT89b], as propriedades destas expressões faciais foram estudadas, aproximadamente, por um período de 25 anos, pelo psicologista Ekman, que propôs uma parametrização dos músculos relacionando-os com suas emoções, num sistema chamado FACS.

[WAT87] representa a ação dos músculos usando uma topologia não-específica de deformação da face. As ações do músculo são testadas no sistema FACS que usa as unidades de ações diretamente num músculo ou num grupo pequeno de músculos. Todas as diferenças encontradas entre as unidades reais de ação e àquelas performadas nos modelos computacionais são facilmente corrigidas trocando os parâmetros musculares, até que se obtenha resultados razoáveis.

[TPBF87] projetaram um modelo baseado na estrutura facial. A textura da pele é simulada por um conjunto de pontos, e os músculos e ossos são simulados por um conjunto de malhas 3D. A pele está num nível mais externo, representado por um conjunto de pontos 3D que definem uma superfície que pode ser modificada. Os ossos representam um nível inicial que não pode ser modificado e os músculos são grupos de pontos com arcos elásticos residentes entre estes dois níveis. Nenhum modelo facial parece ter sido completamente desenvolvido usando esta representação.

[HW88] introduziram uma abordagem automática para a animação de discurso sintetizado por **regras**. Outros parâmetros necessários para o controle dos lábios, queixo e expressão facial são adicionados à uma tabela de parâmetros necessários para o controle do próprio discurso. [LP87] automatizaram a sincronização dos lábios de imagens geradas pelo computador por intermédio da gravação de um discurso real de um ser humano.

Os [TT89b], com a intenção de ilustrar os princípios da animação facial, criaram um modelo de animação baseado na abstração das ações do músculo e não nos próprios músculos.

[AGL87] partindo da análise de um dos maiores problemas da Análise Dinâmica, que é o cálculo das forças e torques necessários para a simulação de alguns movimentos particulares, conhecido como **problema de controle da Análise Dinâmica**, criaram uma interface interativa para a abordagem Dinâmica. Esta interface, juntamente com uma coleção de processos de movimento de baixo-nível, pode ser usada para o controle do movimento de uma figura humana quase em tempo real.

Os [TT86] discutem a problemática do desenvolvimento de uma interface inteligente homem-máquina no campo da Síntese de Imagens e da Animação, decorrente da aplicação da tecnologia originária de Inteligência Artificial e mais especificamente, da Engenharia do Conhecimento.

Desenhistas podem se comunicar com sistemas de animação por intermédio de uma **linguagem natural**. Regras geométricas básicas podem ser dadas ao computador de tal forma que o modelo 3D do objeto seja quase perfeito. Os autores comentam que linguagens baseadas no conceito de *frame* simplificam a descrição inteligente de objetos complexos. Estes sistemas permitem ao desenhista a especificação de certos parâmetros de uma imagem, mais precisamente das câmeras e luzes. *Scripts* e objetos são particularmente apropriados para a descrição dos movimentos. Finalmente, o complexo problema de animação de corpos articulados pode ser resolvido usando **Animação baseada em Conhecimento**. Para tal, os autores criaram uma nova linguagem chamada EXPERTMIRA baseada no sistema MIRA e PROLOG e incorpora conceitos de Inteligência Artificial, Síntese de Imagens e Animação.

– A linguagem EXPERTMIRA

É uma linguagem que compreende dois tipos completamente diferentes de programação: algorítmica e simbólica. Assim, ela é baseada em dois elementos:

1. **objetos procedurais** definidos por meio de tipos de dados de alto nível.
2. **regra e fatos** expressos por uma sintaxe como a do PASCAL mas com semântica parecida com a do PROLOG.

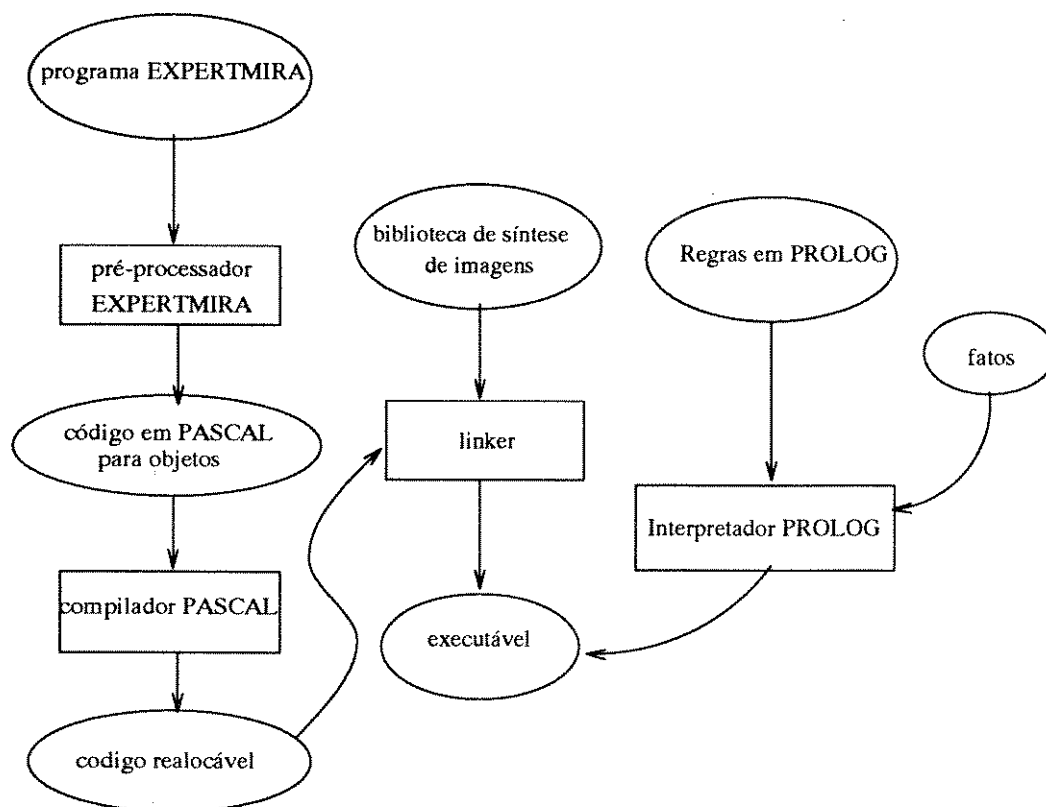


Figura 2.5: Processamento de um programa usando o EXPERTMIRA [TT86].

Assim, a estratégia para o desenvolvimento desta linguagem foi baseada na criação de um pré-processador (escrito em PASCAL), que translada objetos procedurais como faria o PASCAL e ativa regras e fatos como acontece no PROLOG que estão ilustradas na figura 2.5.

O interpretador PROLOG é chamado diretamente pelo programa executável que analisa as regras definidas no programa e os fatos interativamente dados pelo usuário. O interpretador provê suas conclusões ao usuário, mas o programa também pode prosseguir os cálculos baseado nestes fatos. Algumas poucas conclusões podem gerar novos fatos e mesmo, novas regras. A figura 2.6 ilustra o processamento de um programa qualquer, usando o EXPERTMIRA.

Trabalhando com a pesquisa realizada por [WK88] e outros autores, [BRE89] apresenta uma técnica de animação por Dinâmica empregando **funções de custo** para a

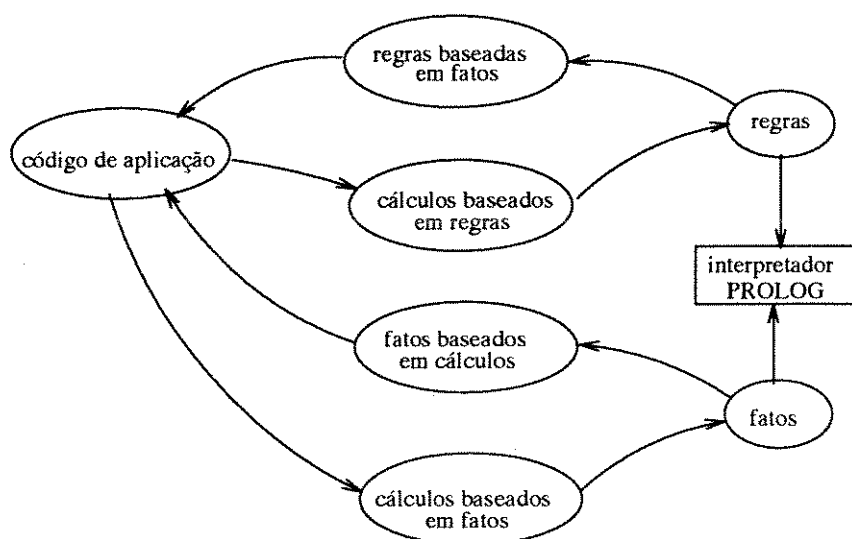


Figura 2.6: Cálculos de Interface e lógica (regras e fatos) [TT86].

produção de movimentos complexos. Estas funções podem ser usadas para a definição de um objetivo para o movimento. As variáveis desta função serão os parâmetros do movimento que serão modificados na intenção de se minimizar um certo custo. Experiências com esta abordagem têm sido realizadas para a determinação de caminhos livres de colisão (evitando obstáculos) e em movimentos de alcance de objetos em braços articulados, sujeitos à restrição.

Enfim, os tipos de problemas que podem ser abordados com o estudo da animação são bastante complexos e ainda são uma área muito recente de pesquisa. Contudo, os autores acreditam que um sistema de animação baseado em conhecimento tem grandes probabilidades de resolver um grande número destes problemas. “Cada objeto pode conhecer seus componentes, suas limitações e as limitações dos outros objetos” [TT86].

Sob o ponto de vista do animador, todos os sistemas de animação existentes têm empecilhos que limitam o processo da animação de uma maneira ou outra. Em geral, gasta-se muitíssimo tempo para a produção do movimento desejado. Por exemplo, foram gastos 14 meses para se produzir o filme de 13 minutos intitulado “Dream Flight” [TT85b]. Ou seja, é necessário ser **preciso** na modelagem da animação e **paciente** em sua produção para que se obtenha um produto final de boa qualidade.

Capítulo 3

O ANIMADO

3.1 Origens

Esta primeira proposta de construção do ANIMADO surgiu a partir da detecção de algumas necessidades fundamentais da abordagem de Animação por Dinâmica. O principal alvo do sistema foi suprir, na medida do possível, estas necessidades. Tendo sempre em mente a criação de vários módulos que atuassem como ferramentas à disposição do sistema, foram criados os cinco módulos que compõem o ANIMADO: o EGESP, o FRED, o COLISÃO, o CONTROLE e o INTEREXPSERV. Todos eles totalmente independentes uns dos outros. O CONTROLE, na realidade deve ser considerado um módulo à parte, que não tem razão de existência se não houver os outros quatro módulos.

Em seguida, a próxima preocupação foi descobrir como **modelar**, da melhor maneira possível, o mundo físico no computador. A obtenção de um modo eficiente para a especificação da Animação Modelada por Dinâmica, partindo do conceito de que o movimento de um objeto é determinado por sua própria natureza e por sua interação com o ambiente em que está posicionado, deve levar em consideração o tipo de objeto que está sendo animado. No caso de corpos articulados (humanos, animais e robôs) que possuem a capacidade de realizar movimentos complexos, esta especificação torna-se ainda mais importante. Desta forma, para efeitos de simplificação, nesta primeira versão do ANIMADO, serão considerados somente os **corpos rígidos 3D** como objetos participantes da animação.

Espera-se com o uso do ANIMADO diminuir as limitações encontradas na especi-

ficação cinemática, em especial, a nível de realismo do movimento.

Como já foi dito no item 1.3 deste trabalho, no ANIMA_{DO}, a Animação Modelada por Dinâmica será produzida nos seguintes estágios distintos:

- Derivação das equações de movimento,
- Resolução das equações de movimento,
- Integração das equações de movimento,
- Detecção de Colisões e conseqüente estudo da Dinâmica de Impacto,
- Controle das equações de movimento,
- Reprodução da Cinemática.

Cada uma destas etapas será detalhada no item 3.3 deste trabalho, incluindo uma descrição das ferramentas que foram escolhidas e desenvolvidas para atuar em cada uma das etapas acima. Mas antes disso, será dedicada uma atenção especial a alguns Métodos de Modelagem do Movimento que existem na literatura.

3.2 Modelagem do movimento

Um dos componentes mais importantes no processo de modelagem do movimento é o desenvolvimento das **equações diferenciais** que representam a dinâmica do sistema. Existe uma grande variedade de formulações para o movimento de corpos rígidos, todas baseadas nas leis de Newton. Basicamente, todas estas formulações produzem o mesmo movimento, ou seja, descrevem a relação entre massas, forças e torques e seus efeitos no sistema. A complexidade destas formulações aumenta de acordo com a complexidade do objeto e com a tarefa envolvida na animação (por exemplo, animar um objeto de geometria complexa, incluindo colisões, sujeito a várias restrições no movimento).

No próximo item serão colocadas algumas das formulações mais utilizadas para a modelagem do movimento por Dinâmica. Em seguida, utilizaremos uma destas formulações para a modelagem do movimento dos objetos utilizando o ANIMA_{DO} (movimento planetário e dois tipos de colisões, citados no apêndice *a* desta dissertação).

3.2.1 Formulações existentes

A formulação mais familiar para a modelagem por Dinâmica é a segunda lei de Newton, onde “A soma de todas as forças F atuantes sobre uma partícula é igual à massa m da partícula, multiplicada por sua aceleração a ”. Outras formulações conhecidas são a de Gibbs-Appell [TT85b], Euler, Lagrangiano [CRA89], D’Alembert e a formulação recursiva de Armstrong [AG85a], muitas vezes, mais apropriadas para a descrição de figuras articuladas.

As principais diferenças entre estas formulações estão diretamente relacionadas às variações de robustez, velocidade, facilidade de implementação e flexibilidade da especificação da formulação escolhida. A representação destas formulações pode ser matricial ou recursiva. Segundo [AG85b], a forma matricial para corpos articulados com n GLs¹ pode ser representada por:

$$M \times \ddot{q} = f \quad (3.1)$$

onde

$\ddot{q}$ = vetor de tamanho n descrevendo a aceleração linear ou angular em cada grau de liberdade i ,

f = vetor de tamanho n descrevendo as **forças totais** (forças ou torques) atuantes em cada grau de liberdade,

M = matriz $n \times n$ descrevendo a influência da força total f_i atuante no grau de liberdade i sobre a aceleração $\ddot{q}$ de cada grau de liberdade do corpo.

Para encontrar as acelerações, a matriz M deve (intuitivamente) ser invertida. Contudo, o custo do cálculo da inversa de uma matriz é bastante alto, sendo assim, utilizam-se Métodos Numéricos de Eliminação para a Resolução de Sistemas. As formulações recursivas não são tão caras quanto as matriciais. A Formulação de Armstrong (assume uma estrutura em árvore para um corpo articulado) começando no nível das folhas até a raiz, calculando as forças e torques e retornando às folhas, calculando as acelerações. De um

¹Graus de Liberdade

modo geral, as formulações recursivas são mais gerais. Por exemplo, algumas equações de restrição² podem ser incorporadas na formulação matricial, integrando as equações num só processo. Pode ser que o custo para a resolução desta matriz acabe se tornando alto.

Talvez, a mais intuitiva das formulações dinâmicas seja a de Euler [CRA89], que será abordada neste trabalho. A formulação de Euler é a base da formulação recursiva de Armstrong. Ela é representada por duas equações vetoriais que descrevem o movimento de um corpo rígido 3D:

$$\begin{aligned} f &= ma - mc \otimes \dot{w} + mw \otimes (w \otimes c) \\ \tau &= J\dot{w} + mc \otimes a + w \otimes Jw \end{aligned} \quad (3.2)$$

onde

f = força aplicada no corpo,

τ = torque aplicado no corpo,

m = massa do corpo,

J = matriz 3x3 do tensor de inércia que descreve como a massa do corpo está distribuída,

c = vetor 3D do centro de massa do corpo,

w = vetor 3D da velocidade angular do corpo no sistema de coordenadas local,

a = aceleração linear,

$\dot{w}$ = aceleração angular.

O significado da representação matricial dos produtos vetoriais na equação 3.2, é especificado a seguir. Dados dois vetores a e b , tem-se:

$$a \otimes b = a \otimes b \quad e \quad b \otimes a = b \otimes a \quad (3.3)$$

²Restrição = qualquer controle de movimento de baixo nível, incluindo **restrições no movimento** (não estender além de uma certa junta limite, ou não interseccionar outros corpos durante a colisão) e **controle posicional** (rotacione o segmento de 10 graus durante o próximo intervalo de tempo).

Por conveniência, estes termos serão descritos em relação ao SC^3 local fixo no corpo. Desta forma, o **centro de massa** e o **tensor de inércia** não mudarão seus valores durante o movimento.

Segundo [AG85a], se forem modelados corpos do tipo articulado, as interações existentes entre os seus segmentos também devem ser consideradas. Para corpos articulados com três GLs nas juntas de revolução, utilizam-se as seguintes equações, muito semelhantes às citadas anteriormente:

$$\begin{aligned} f_i &= m_i a_i - m_i c_i \otimes \dot{w}_i + m_i w_i \otimes (w_i \otimes c_i) \\ \tau_i &= J_i \dot{w}_i + m_i c_i \otimes a_i + w_i \otimes J_i w_i \end{aligned} \quad (3.4)$$

onde $i = i_{esimo}$ segmento do objeto.

Usando a formulação matricial, as restrições tornam-se parte integrante da matriz. Se as restrições forem declaradas como valores para as acelerações dos segmentos do corpo, a matriz ficará simplificada, removendo GLs. Esta abordagem foi usada para corpos articulados, juntamente com a formulação dinâmica de D'Alembert [IC87]. Segundo [MW88], a formulação matricial de Euler para dois corpos rígidos 3D articulados é:

$$\begin{pmatrix} m_p & -m_p c_p \otimes & 0 & 0 & -1 \\ m_p c_p \otimes & J_p & 0 & 0 & 0 \\ 0 & 0 & m_s & -m_s c_s \otimes & R_s \\ 0 & 0 & m_s c_s \otimes & J_s & 0 \\ 1 & \otimes I_s & -R_s & 0 & 0 \end{pmatrix} \begin{pmatrix} a_p \\ \dot{w}_p \\ a_s \\ \dot{w}_s \\ f_{restr} \end{pmatrix} = \begin{pmatrix} f_p - m_p w_p \otimes (w_p \otimes c_p) \\ \tau_p - w_p \otimes J_p w_p \\ f_s - m_s w_s \otimes (w_s \otimes c_s) \\ \tau_s - w_s \otimes J_s w_s \\ w_p \otimes w_p \otimes I_s \end{pmatrix} \quad (3.5)$$

Ou seja, um sistema composto por cinco equações vetoriais:

1. a equação da força para o segmento p ,
2. a equação do torque para o segmento p ,
3. a equação da força para o segmento s ,

³Sistema de Coordenadas

4. a equação do torque para o segmento s ,
5. a equação de restrição que mantém a junta entre os dois segmentos.

A força f_{restr} é uma força de restrição que faz com que os dois segmentos permaneçam unidos. I_s é um vetor da origem do SC do objeto p à origem do SC do objeto s (a junta). R_s é uma matriz de rotação que leva o SC do objeto p no SC do objeto s . As outras forças e torques incluem quaisquer forças e torques externos incluindo os aplicados pelo usuário. As acelerações linear e angular dos dois objetos são encontradas em relação às suas coordenadas locais. A força de restrição f_{restr} é expressa no SC do objeto p . A adição de forças de restrição extras (f_{restr}) origina um sistema no qual o número de colunas é igual ao número de linhas. O tamanho da matriz mudará de acordo com a presença ou não de restrições.

Alguns métodos, como o descrito por [WFB87] adicionam **uma função objetivo** que define uma quantidade que deve ser minimizada (o torque resultante ou a energia gasta). Infelizmente, as técnicas de minimização são complicadas, caras e não suficientemente robustas.

Agora, depois de conhecer algumas das formulações existentes no âmbito da Animação Modelada por Dinâmica, será desenvolvida a formulação atualmente utilizada no ANIMADO.

3.2.2 A formulação utilizada no ANIMADO: o Método de Euler

Em geral, o movimento de um corpo rígido pode ser decomposto em um movimento linear de um ponto localizado no centro de massa de um corpo (devido à uma força externa) mais um movimento rotacional sobre o centro de massa do corpo (devido a um torque externo). Assim, o Método de Euler é composto por seis equações: três equações para a translação que relacionam as **acelerações lineares** e **massas** às **forças** e outras três equações para a rotação que relacionam as **acelerações angulares** e a **distribuição de massa** ao **torque**.

A partir da conhecida primeira lei de Newton, tem-se as seguintes equações escalares:

$$\begin{aligned}
f_x &= m a_x \\
f_y &= m a_y \\
f_z &= m a_z
\end{aligned} \tag{3.6}$$

onde

f = força total,

a = aceleração linear do centro de massa relativa ao SC {A}⁴,

m = massa do corpo.

É importante salientar que a **força** e a **aceleração** devem ser descritas em relação **ao mesmo** SC. A força ${}^A f$ relativa ao SC {A} é responsável pela aceleração ${}^B a$ em termos do SC {B}. A matriz direção dos co-senos pode ser usada para encontrar as componentes da força no SC {A} relativas ao SC {B}.

Ou seja,

$${}^A f = {}^A R {}^B f \Rightarrow {}^A R^{-1} {}^A f = {}^A R^{-1} {}^A R {}^B f \Rightarrow {}^B f = {}^A R^{-1} {}^A f \tag{3.7}$$

mas, como são ortonormais:

$${}^A R = {}^B R^{-1} = {}^B R^t, \tag{3.8}$$

então:

$${}^B f = {}^A R^t {}^A f \tag{3.9}$$

Sabendo que ${}^A R$ corresponde à matriz de rotação do SC {A} para o {B} e ${}^A R^t$ corresponde à transposta de ${}^A R$, então, podemos utilizar o produto interno do vetor $\vec{f} = (f_x, f_y, f_z)^t$ com cada eixo (x, y, z) representados pelas colunas da matriz ${}^A R^t$. Ou seja:

⁴É por isso que o centro de massa atua como se o corpo todo estivesse localizado no centro de massa e todas as forças estivessem atuando neste ponto.

$${}^A_B R = \begin{pmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{pmatrix} \quad (3.10)$$

$${}^A_B R^t = \begin{pmatrix} a_{00} & a_{10} & a_{20} \\ a_{01} & a_{11} & a_{21} \\ a_{02} & a_{12} & a_{22} \end{pmatrix} \quad (3.11)$$

Que originam as seguintes equações:

$$\begin{aligned} {}^B f_x &= {}^A f_x a_{00} + {}^A f_x a_{10} + {}^A f_x a_{20} \\ {}^B f_y &= {}^A f_y a_{01} + {}^A f_y a_{11} + {}^A f_y a_{21} \\ {}^B f_z &= {}^A f_z a_{02} + {}^A f_z a_{12} + {}^A f_z a_{22} \end{aligned} \quad (3.12)$$

As equações rotacionais cujos valores estão descritos em relação ao SC {B} cuja origem é o centro de massa do corpo rígido são dadas por:

$$\tau = \frac{dL}{dt} \quad (3.13)$$

$$L = I w \quad (3.14)$$

onde,

L = momento angular do corpo,

τ = torque externo aplicado,

I = momento de inércia,

w = velocidade angular do corpo que se deseja encontrar.

Derivando 3.14 em relação ao SC {B} fixo no corpo, e substituindo-se 3.13 têm-se:

$$\begin{aligned}
w &= I^{-1} L \\
\frac{dw}{dt} &= \frac{d I^{-1}}{dt} L + I^{-1} \frac{dL}{dt} \\
\frac{dw}{dt} &= \frac{d I^{-1}}{dt} L + I^{-1} \tau \\
\frac{dw}{dt} &= w^* I^{-1} L + I^{-1} w^{*t} L + I^{-1} \tau \\
\frac{dw}{dt} &= w^* w + I^{-1} (w^{*t} L + \tau) \\
\frac{dw}{dt} &= I^{-1} (L \otimes w + \tau) \\
I \frac{dw}{dt} &= L \otimes w + \tau \\
\tau &= I \frac{dw}{dt} - L \otimes w \\
\tau &= I \frac{dw}{dt} - I w \otimes w \\
\tau &= I \frac{dw}{dt} + w \otimes (I w)
\end{aligned} \tag{3.15}$$

onde w^* = matriz anti-simétrica a w , ou seja:

Para todo vetor a , w^*a é equivalente a $w \otimes a^5$.

Se forem escolhidos os eixos principais como eixos fixos no corpo e modificar 3.15 no SC {B}, têm-se:

$$\begin{aligned}
\tau_x &= I_{xx} \dot{w}_x + (I_{zz} - I_{yy}) w_y w_z \\
\tau_y &= I_{yy} \dot{w}_y + (I_{xx} - I_{zz}) w_x w_z \\
\tau_z &= I_{zz} \dot{w}_z + (I_{yy} - I_{xx}) w_x w_y
\end{aligned} \tag{3.17}$$

onde

w = velocidade angular do corpo no SC {B} relativo ao SC {A}, mas expressa no SC {B},

$\dot{w}$ = aceleração angular,

τ = torque atuante no corpo.

⁵Exemplificando: Seja o vetor $b = (b_1, b_2, b_3)^t$, define-se:

$$b^* = \begin{pmatrix} 0 & b_3 & -b_2 \\ -b_3 & 0 & b_1 \\ b_2 & -b_1 & 0 \end{pmatrix} \tag{3.16}$$

Observe que os produtos de inércia não aparecem. Estas são as conhecidas **Equações de Euler** [GOL50] e x , y e z são as direções dos eixos principais.

A seguir, será representada uma forma mais geral de representar estas equações. **Todos** os valores são descritos em relação a um único SC (que pode ser o $\{A\}$) e que, no nosso caso refere-se à **posição** e à **orientação** instantânea de um SC $\{B\}$ fixo no corpo.

A forma vetorial das duas equações que descrevem o movimento de um único corpo rígido é:

$$\begin{aligned} f &= ma - mc \otimes \dot{w} + mw \otimes (w \otimes c) \\ \tau &= J\dot{w} + mc \otimes a + w \otimes Jw \end{aligned} \tag{3.18}$$

onde

f = força total aplicada no corpo,

τ = torque total atuando no corpo,

m = massa do corpo,

J = matriz do tensor inercial 3x3 que descreve como a massa está distribuída,

c = vetor 3D do centro de massa,

w = vetor 3D da velocidade angular atual em relação ao SC $\{B\}$,

a = aceleração linear,

$\dot{w}$ = aceleração angular.

Estas duas últimas variáveis serão encontradas via resolução das duas equações referenciadas em 3.2.

Como estes termos foram descritos relativamente a um SC $\{B\}$ fixo no corpo, os valores da **massa**, do **centro de massa** e da **matriz do tensor inercial** não se alteram durante a animação e seus valores podem ser dados inicialmente pelo usuário ou calculados pelo sistema.

Do produto vetorial entre vetores temos que:

$$\begin{aligned}
f_x &= ma_x - m(\dot{w}_z c_y - \dot{w}_y c_z) + m[(w \otimes c)_z w_y - (w \otimes c)_y w_z] \\
f_x &= m[a_x - \dot{w}_z c_y - \dot{w}_y c_z + (c_y w_x - c_x w_y - (c_x w_z - c_z w_x) w_z)] \\
f_x &= m[a_x - \dot{w}_z c_y - \dot{w}_y c_z + c_y w_x w_y - c_x w_z^2 + c_z w_x w_z] \\
f_x &= m[a_x - c_x(w_y^2 + w_z^2) + c_y(w_x w_y - \dot{w}_z) + c_z(w_x w_z + \dot{w}_y)]
\end{aligned} \tag{3.19}$$

$$\begin{aligned}
f_y &= ma_y - m(\dot{w}_x c_z - \dot{w}_z c_x) + m[(w \otimes c)_x w_z - (w \otimes c)_z w_x] \\
f_y &= m[a_y - \dot{w}_x c_z + \dot{w}_z c_x + w_z(c_z w_y - c_y w_z) - w_x(c_y w_x - c_x w_y)] \\
f_y &= m[a_y - \dot{w}_x c_z + \dot{w}_z c_x + w_z c_z w_y - w_z^2 c_y - w_x^2 c_y + w_x c_x w_y] \\
f_y &= m[a_y + c_x(w_x w_y + \dot{w}_z) - c_y(w_x^2 + w_z^2) + c_z(w_y w_z - \dot{w}_x)]
\end{aligned} \tag{3.20}$$

$$\begin{aligned}
f_z &= m[a_z - (\dot{w}_y c_x - \dot{w}_x c_y) + w_x(c_x w_z - c_z w_x) - w_y(c_z w_y - c_y w_z)] \\
f_z &= m[a_z - \dot{w}_y c_x + \dot{w}_x c_y + w_x c_x w_z - w_x^2 c_z - w_y^2 c_z + w_y c_y w_z] \\
f_z &= m[a_z - \dot{w}_y c_x + \dot{w}_x c_y + w_x c_x w_z - w_x^2 c_z - w_y^2 c_z + w_y c_y w_z] \\
f_z &= m[a_z + c_x(w_x w_z - \dot{w}_y) + c_y(w_y w_z + \dot{w}_x) - c_z(w_x^2 + w_y^2)]
\end{aligned} \tag{3.21}$$

Analogamente para o torque:

$$\tau_x = m(a_z c_y - a_y c_z) + I_{xx} \dot{w}_x + (I_{zz} - I_{yy}) w_y w_z + \tag{3.22}$$

$$I_{xy}(w_x w_z - \dot{w}_y) - I_{xz}(w_x w_y + \dot{w}_z) + I_{yz}(w_z^2 - w_y^2)$$

$$\tau_y = m(a_x c_z - a_z c_x) + I_{yy} \dot{w}_y + (I_{xx} - I_{zz}) w_x w_z +$$

$$I_{yz}(w_y w_x - \dot{w}_z) - I_{xy}(w_y w_z + \dot{w}_x) + I_{xz}(w_x^2 - w_z^2)$$

$$\tau_z = m(a_y c_x - a_x c_y) + I_{zz} \dot{w}_z + (I_{yy} - I_{xx}) w_x w_y +$$

$$I_{xz}(w_y w_z - \dot{w}_x) - I_{yz}(w_x w_z + \dot{w}_y) + I_{xy}(w_y^2 - w_x^2)$$

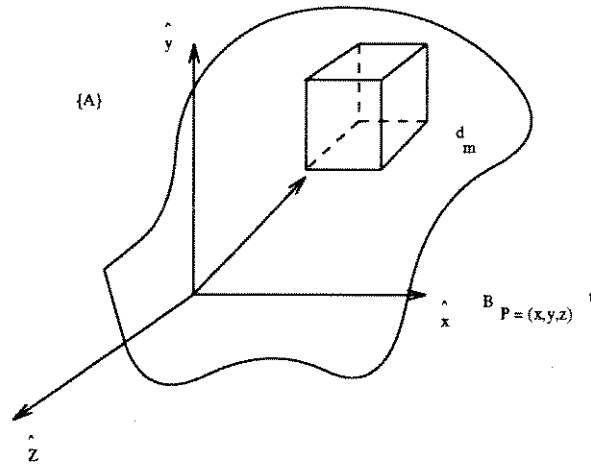


Figura 3.1: O tensor de inércia de um objeto descreve a distribuição de massa do objeto. O vetor ${}^B P$ localiza o elemento massa, d_m [CRA89].

3.2.2.1 Cálculo dos Momentos de Inércia e Eixos Principais

O tensor de inércia pode ser considerado uma generalização do momento de inércia escalar de um objeto. Deve-se definir um conjunto de valores que fornecem informação sobre a distribuição de massa de um corpo rígido em relação a um SC referencial como ilustra a figura 3.1.

Calculado o valor do centro de massa no SC {B}, da Física [SYM71], para objetos cuja massa está simetricamente distribuída ao redor deste SC {B}, calculam-se **três momentos de inércia**, um sobre cada eixo, onde ${}^B P_c = (x_c, y_c, z_c)^t$ localiza o centro de massa relativo ao SC {B}:

$$\begin{aligned} I_{xx} &= \int (y^2 + z^2) dm \\ I_{yy} &= \int (x^2 + z^2) dm \\ I_{zz} &= \int (x^2 + y^2) dm \end{aligned} \tag{3.23}$$

Caso o objeto não seja simétrico, é necessário calcular os **três produtos de inércia**:

$$\begin{aligned}
I_{xy} &= \int (xy) dm \\
I_{xz} &= \int (xz) dm \\
I_{yz} &= \int (yz) dm
\end{aligned} \tag{3.24}$$

Que serão definidos numa **matriz do tensor inercial**:

$$J = \begin{pmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{pmatrix} \tag{3.25}$$

Este conjunto de seis quantidades independentes dependerá da **posição** e da **orientação** do SC no qual estiverem definidas. Se a escolha da orientação do SC referencial for livre, é possível fazer com que os **produtos** de inércia sejam zero (SC localizado no centro de massa do objeto). Os eixos do SC referencial (quando estão alinhados) são chamados de **Eixos Principais** e os momentos de inércia correspondentes são os **momentos principais** de inércia.

Para calcular estes valores sobre quaisquer eixos ou pontos do espaço, por exemplo, num outro SC cujos eixos maiores sejam paralelos ao SC originário no centro de massa mas transladado de $(\delta x, \delta y, \delta z)$, utiliza-se o **Teorema dos Eixos Paralelos** que relaciona o tensor de inércia num SC com origem no centro de massa do tensor de inércia com respeito a outro SC referencial. Sendo que (I_{xx}, I_{yy}, I_{zz}) estão localizados no centro de massa do corpo e $(I'_{xx}, I'_{yy}, I'_{zz})$ é um SC arbitrariamente transladado:

$$J = \begin{pmatrix} I'_{xx} = I_{xx} + m(\delta y^2 + \delta z^2) \\ I'_{yy} = I_{yy} + m(\delta x^2 + \delta z^2) \\ I'_{zz} = I_{zz} + m(\delta x^2 + \delta y^2) \\ I'_{xy} = I_{xy} + m(\delta x)(\delta y) \\ I'_{xz} = I_{xz} + m(\delta x)(\delta z) \\ I'_{yz} = I_{yz} + m(\delta y)(\delta z) \end{pmatrix} \tag{3.26}$$

onde δx , δy e δz correspondem à translação em relação ao SC referencial e m é a massa do objeto.

As integrações numéricas para o tensor de inércia são realizadas num SC cuja origem coincida com o centro de massa do objeto para facilitar a separação do momento linear e do rotacional. Intuitivamente, os **eixos principais** correspondem aos **eixos de simetria** do objeto, e os momentos de inércia principais correspondem aos momentos de inércia associados aos eixos principais. É por isso que os eixos principais e os momentos de inércia principais devem ser calculados uma única vez.

Caso o novo SC tenha sido também rotacionado em relação ao SC referencial, os novos valores relativos ao novo SC podem ser derivados a partir do centro de massa original em duas etapas:

1. Criar um SC (SC') paralelo ao SC original (SC) mas deslocado tal que sua origem esteja localizada numa posição desejada (na origem do SC (SC'')). Este novo SC (SC') é criado usando 3.26.
2. Os valores para o SC (SC''), rotacionado e transladado, podem ser derivados de SC' como segue. Inicialmente, necessita-se a **matriz direção dos co-senos** que descreve como o novo eixo x está relacionado com o antigo eixo $x(a_{00}, a_{10}, a_{20})$, o novo eixo y com o antigo eixo $y(a_{01}, a_{11}, a_{21})$ e o novo eixo $z(a_{02}, a_{12}, a_{22})$ com o antigo eixo z .

A matriz de rotação D_k que representa a orientação de um SC pode ser considerada como três vetores direção dos co-senos(colunas) definindo os eixos do SC. A coluna 0 representa o novo eixo x , a 1 o novo eixo y e a 2 o novo eixo z .

$$D_k = \begin{pmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{pmatrix} \quad (3.27)$$

Agora, os novos momentos e produtos de inércia são encontrados através da **transformação de similaridade** [GOL50], num SC paralelo àquele centrado no centro de massa de I' :

$$I'' = D_k I' D_k^t \quad (3.28)$$

onde,

D_k = matriz ortogonal de terceira ordem.

Para o propósito da transformação rotacional, a matriz D_k é a matriz **direção dos co-senos**, que expandida origina:

$$\begin{aligned}
 I''_{xx} &= I'_{xx}a_{00}^2 + I'_{yy}a_{01}^2 + I'_{zz}a_{02}^2 - 2I'_{xy}a_{00}a_{01} - 2I'_{xz}a_{00}a_{02} - 2I'_{yz}a_{01}a_{02} \\
 I''_{yy} &= I'_{xx}a_{10}^2 + I'_{yy}a_{11}^2 + I'_{zz}a_{12}^2 - 2I'_{xy}a_{10}a_{11} - 2I'_{xz}a_{10}a_{12} - 2I'_{yz}a_{11}a_{12} \\
 I''_{zz} &= I'_{xx}a_{20}^2 + I'_{yy}a_{21}^2 + I'_{zz}a_{22}^2 - 2I'_{xy}a_{20}a_{21} - 2I'_{xz}a_{20}a_{22} - 2I'_{yz}a_{21}a_{22} \\
 I''_{xy} &= (a_{00}a_{11} + a_{01}a_{10})I'_{xy} + (a_{00}a_{12} + a_{02}a_{10})I'_{xz} + (a_{01}a_{12} + a_{02}a_{11})I'_{yz} \\
 &\quad - (a_{00}a_{10}I'_{xx} + a_{01}a_{11}I'_{yy} + a_{02}a_{12}I'_{zz}) \\
 I''_{xz} &= (a_{00}a_{21} + a_{01}a_{20})I'_{xy} + (a_{00}a_{22} + a_{02}a_{20})I'_{xz} + (a_{01}a_{22} + a_{21}a_{02})I'_{yz} \\
 &\quad - (a_{00}a_{20}I'_{xx} + a_{01}a_{21}I'_{yy} + a_{02}a_{22}I'_{zz}) \\
 I''_{yz} &= (a_{10}a_{21} + a_{11}a_{20})I'_{xy} + (a_{10}a_{22} + a_{20}a_{12})I'_{xz} + (a_{11}a_{22} + a_{12}a_{21})I'_{yz} \\
 &\quad - (a_{10}a_{20}I'_{xx} + a_{11}a_{21}I'_{yy} + a_{12}a_{22}I'_{zz})
 \end{aligned} \tag{3.29}$$

Ou seja,

- Se o SC não estiver localizado no centro de massa do corpo, mas transladado, as equações 3.26 podem ser usadas.
- Se o SC estiver rotacionado, a equação 3.28 ou 3.29 pode ser usada.

Alguns fatos adicionais importantes sobre tensores de inércia são:

1. Se os dois eixos de um SC referencial formarem um plano de simetria para a distribuição de massa do corpo, os produtos de inércia cujo índice tiver uma coordenada normal ao plano de simetria serão zero.
2. Momentos de inércia devem sempre ser positivos. Produtos de inércia podem ter sinal negativo.
3. A soma dos três momentos de inércia são invariantes sob mudança de orientação no SC referencial.

4. Os autovalores de um tensor de inércia são os momentos principais do corpo. Os autovetores associados são os eixos principais.

3.3 Etapas da animação utilizando o ANIMA_{DO}

O ANIMA_{DO} tem como filosofia, a flexibilidade e modularidade de ferramentas que podem ser agrupadas e/ou acrescentadas em bibliotecas. Esta característica possibilita uma boa estruturação do sistema. Ele é composto por cinco módulos independentes, como ilustra a figura 3.2, com funções específicas, todas elas relacionadas ao processo de Animação Modelada por Dinâmica. Nesta figura, pode-se observar como o ANIMA_{DO} está organizado internamente, como os módulos se comunicam entre si (através do envio de mensagens e sinais de controle), e onde se enquadraria um módulo que permite uma visualização realista da animação, por exemplo, o TOOKIMA. Estes módulos serão detalhados a seguir, incluindo a fase de inicialização do sistema que diz respeito ao processo de modelagem da Dinâmica, utilizando a formulação de Euler, como detalhado no item 3.2.2.

3.3.1 A fase de inicialização

As informações necessárias para o funcionamento do ANIMA_{DO}, agora serão explicitadas.

Considerando os corpos rígidos 3D como massas estendidas e assumindo que estas massas não mudam de forma ou massa durante a animação, as informações necessárias para a fase de inicialização do ANIMA_{DO} serão:

1. Informações Invariáveis

- (a) m = massa do objeto
- (b) c = centro de massa do objeto
- (c) I = distribuição de massa ao redor do centro de massa

O centro de massa é um vetor 3D que descreve uma localização no espaço. Pode ser um vetor da origem do espaço mundo (onde os objetos estão posicionados). Entretanto, é melhor assumir um SC local fixo no corpo e descrever o centro de massa em

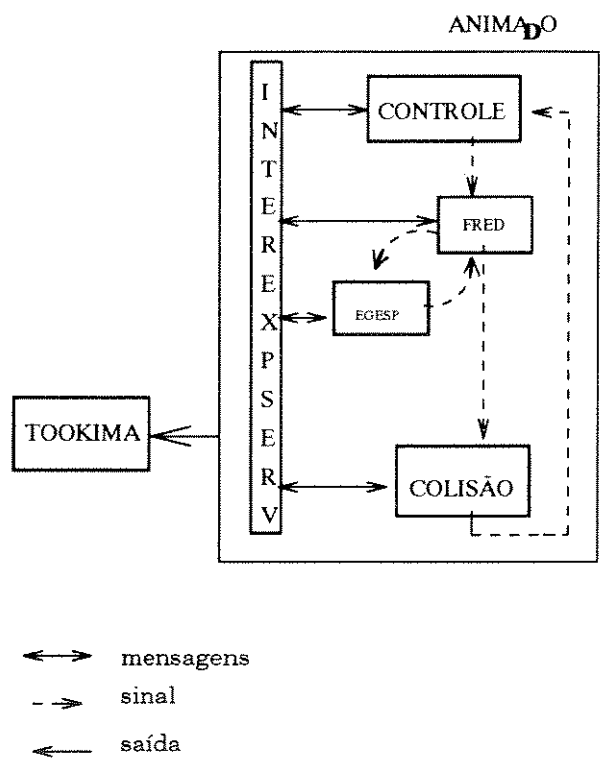


Figura 3.2: Um protótipo de um Sistema de Animação Modelada por Dinâmica.

relação a este SC local para que o valor do centro de massa não se altere quando o objeto se movimentar. Como a localização do SC local em relação ao SC mundo é conhecida, fica fácil encontrar o centro de massa em coordenadas do mundo, caso necessário. Tipicamente, este SC local já é usado para descrever a geometria dos objetos. Assim, os valores da **massa**, do **centro de massa**, e do **tensor inercial** não mudam durante o movimento e podem ser fornecidos pelo usuário ou calculados pelo sistema.

2. Informações Variáveis

Os corpos rígidos têm seis GLs (três de translação, três de rotação). Assumindo um SC local fixo no objeto, os GLs translacionais podem representar um deslocamento relativo a um SC fixo e inercial do espaço mundo, ou ao longo do SC local atual (ou qualquer outro eixo). Similarmente, os GLs de rotação descrevem a orientação do objeto em relação a algum SC referencial.

A ordem das rotações serão fixadas como:

- (a) rotação em x
- (b) rotação em y
- (c) rotação em z ,

conhecidas como **rotações de Euler** [CRA89]. As ordens das rotações de Euler podem variar. No caso, para $x \rightarrow y \rightarrow z$, a rotação em x é relativa ao eixo x original, a rotação em y é sobre o eixo y criado pela rotação em x e a rotação em z é sobre o eixo z criado pelas duas primeiras rotações citadas.

É bastante sensato assumir que o eixo z local representa o eixo longitudinal do corpo, quando houver um eixo longitudinal óbvio.

Outras informações variantes são as forças f e os torques τ que causam o movimento. Se um número de forças estiverem atuando no corpo, o efeito translacional total pode ser encontrado somente somando-as. O centro de massa do corpo terá movimento como se fosse uma partícula massa, influenciada por uma força. O mesmo torque causa um movimento rotacional sobre um eixo particular.

Se todas as forças fossem aplicadas no centro de massa, elas não produziriam nenhum torque, contudo, uma força atuando num ponto do corpo (sem ser o centro de massa)

causará um torque. Para encontrar um torque sobre um SC devido a uma força $f = (f_x, f_y, f_z)$, aplicada a uma distância $d = (x, y, z)$, ambas definidas em relação a este SC, calcule o produto vetorial:

$$\tau = d \otimes f \quad (3.30)$$

Ou seja, dados os vetores:

$$\begin{aligned} \vec{d} &= (x, y, z) \\ \vec{f} &= (f_x, f_y, f_z) \end{aligned} \quad (3.31)$$

define-se o produto vetorial $\vec{p} \otimes \vec{f}$ mediante a expressão:

$$\vec{d} \otimes \vec{f} = (x\vec{i} + y\vec{j} + z\vec{k}) \otimes (f_x\vec{i} + f_y\vec{j} + f_z\vec{k}) \quad (3.32)$$

portanto,

$$\begin{aligned} \vec{d} \otimes \vec{f} &= (yf_z - zf_y)\vec{i} + (zf_x - xf_z)\vec{j} + (xf_y - yf_x)\vec{k} \\ \vec{d} \otimes \vec{f} &= (f_z y - f_y z, f_x z - f_z x, f_y x - f_x y) \end{aligned} \quad (3.33)$$

Ou seja, o produto vetorial que origina o torque $\tau = (\tau_x, \tau_y, \tau_z)$ resulta nas seguintes equações escalares:

$$\begin{aligned} \tau_x &= f_z y - f_y z \\ \tau_y &= f_x z - f_z x \\ \tau_z &= f_y x - f_x y \end{aligned} \quad (3.34)$$

O ANIMADO utiliza a formulação matricial destas equações dada por:

$$\begin{pmatrix} m & -mc \otimes \\ mc \otimes & J \end{pmatrix} \begin{pmatrix} a \\ \dot{w} \end{pmatrix} = \begin{pmatrix} f - mw \otimes (w \otimes c) \\ \tau - w \otimes Jw \end{pmatrix} \quad (3.35)$$

Que expandida resulta:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & c_z & -c_y \\ 0 & 1 & 0 & -c_z & 0 & c_x \\ 0 & 0 & 1 & c_y & -c_x & 0 \\ 0 & -mc_z & mc_y & I_{xx} & -I_{xy} & -I_{xz} \\ mc_z & 0 & -mc_x & -I_{xy} & I_{yy} & -I_{yz} \\ -mc_y & mc_x & 0 & -I_{xz} & -I_{yz} & I_{zz} \end{pmatrix} \begin{pmatrix} a_x \\ a_y \\ a_z \\ \dot{w}_x \\ \dot{w}_y \\ \dot{w}_z \end{pmatrix} = \quad (3.36)$$

$$\begin{pmatrix} \frac{f_x}{m} + c_x(w_y^2 + w_z^2) - c_y(w_x w_y) - c_z(w_x w_z) \\ \frac{f_y}{m} - c_x(w_x w_y) + c_y(w_x^2 + w_z^2) - c_z(w_y w_z) \\ \frac{f_z}{m} - c_x(w_x w_z) - c_y(w_y w_z) + c_z(w_x^2 + w_y^2) \\ \tau_x - (I_{zz} - I_{yy})w_y w_z - I_{xy}(w_x w_z) + I_{xz}(w_x w_y) - I_{yz}(w_z^2 - w_y^2) \\ \tau_y - (I_{xx} - I_{zz})w_x w_z - I_{yz}(w_y w_x) + I_{xy}(w_y w_z) - I_{xz}(w_x^2 - w_z^2) \\ \tau_z - (I_{yy} - I_{xx})w_x w_y - I_{xz}(w_y w_z) + I_{yz}(w_x w_z) - I_{xy}(w_y^2 - w_x^2) \end{pmatrix} \quad (3.37)$$

Ou seja, um sistema de equações composto por seis incógnitas ($a_x, a_y, a_z, \dot{w}_x, \dot{w}_y$ e $\dot{w}_z$) e seis equações. As incógnitas são encontradas resolvendo-se este sistema para serem utilizadas no ANIMA_{DO}.

O tensor de inércia pode ser definido relativamente a qualquer SC. No caso do ANIMA_{DO}, o tensor momento de inércia será definido num SC {B} fixo no objeto. Na maioria das vezes, deseja-se que o movimento do corpo rígido seja descrito em relação ao SC {B} fixo no objeto e o ponto de aplicação da força é expresso em relação a este SC {B}. Contudo, a força externa aplicada é atualmente dada em termos do SC {A} podendo ser convertida para o SC {B} multiplicando-a pela matriz direção dos co-senos que dá a orientação do SC {A} em relação ao SC {B}. Esta matriz é a inversa (neste caso a transposta) da matriz que representa como o SC {B} está definido relativamente ao SC {A}.

Caso existam várias forças e torques atuando sobre um corpo, estes valores podem ser facilmente encontrados (em relação SC {B}) somando-se as forças no SC {B}. Estas forças são usadas para encontrar os torques causados por estas forças usando a equação 3.34 e somando estes torques aos torques puros para encontrar o torque total (τ). Depois disso, a força total efetiva é **aplicada na origem do SC {B}**. O SC não precisa estar localizado no centro de massa do objeto para que este procedimento seja válido.

Estes valores podem ser fornecidos pelo usuário ou calculados pelo sistema durante a fase de inicialização da animação. No caso do **ANIMADO**, f e τ são dois vetores dados pelo usuário. O torque incluirá quaisquer torques causados por forças. As acelerações (linear a e angular $\dot{w}$), velocidades (linear v e angular w) e posições ocupadas pelo objeto serão encontradas através dos módulos detalhados a seguir.

3.3.2 Resolução das equações de movimento: o EGESP

Utilizando a formulação matricial 3.36 e 3.37, o objetivo da resolução deste sistema de equações é encontrar as acelerações (linear e angular), dadas as forças e os torques (assumindo que a posição e velocidade iniciais são conhecidas). Portanto, no máximo, o sistema possuirá seis equações e seis incógnitas ($a_x, a_y, a_z, \dot{w}_x, \dot{w}_y, \dot{w}_z$). Para resolver o sistema de equações, encontrando as acelerações, será utilizado o EGESP⁶, com pivoteamento parcial (parâmetro T), usando estrutura de dados para matrizes esparsas. Este método será detalhado a seguir.

3.3.2.1 O EGESP

Talvez, a forma mais natural de resolução de um sistema $Ax = b$ seja obter a inversa da matriz A , escrevendo $x = A^{-1}b$. Segundo [RL88] é totalmente desaconselhável o cálculo computacional de A^{-1} já que a matriz obtida pode diferir muito da verdadeira A^{-1} , ainda que a precisão utilizada seja grande.

Dentre os **métodos diretos** para a resolução de sistemas lineares do tipo $Ax = b$, os mais conhecidos são o Método de Gauss e a Decomposição LU [RL88]. Estes métodos são chamados de métodos diretos porque trabalham diretamente na matriz A evitando o cálculo direto da inversa de A , obtendo a solução do sistema através de operações elementares, num número finito de passos. É importante analisar que estratégias podem ser acopladas ao método para contornar os problemas freqüentes de erros de arredondamento a que os métodos numéricos estão sujeitos. No **ANIMADO**, foi escolhido o Método de Eliminação de Gauss com Pivoteamento Parcial (Parâmetro T).

Freqüentemente, a aplicação da formulação de Euler para uma animação com-

⁶método numérico de Eliminação de Gauss para resolução de sistemas lineares ESParsos

posta de muito objetos origina matrizes muito esparsas que podem ser resolvidas rapidamente via Métodos de Resolução de Matrizes esparsas, como por exemplo, o EGESP.

O EGESP, prevendo o alto custo para a resolução da matriz das equações de movimento possui uma boa estrutura de dados para armazenamento por linhas, suficientemente eficiente para trabalhar com matrizes esparsas. Geralmente, esta estrutura faz com que o custo da formulação matricial se equipare ao custo da formulação recursiva citada no item 3.2.1.

A idéia básica advinda de Gauss é transformar a matriz A , através de operações elementares, numa matriz triangular superior. Considere o sistema linear $Ax = b$, onde A é uma matriz $n \times n$, triangular superior, com elementos da diagonal diferentes de zero. Então:

$$\begin{pmatrix} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + a_{34}x_4 \\ . \\ . \\ . \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + a_{n4}x_4 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ . \\ . \\ . \\ b_n \end{pmatrix} \quad (3.38)$$

A notação usada para a implementação do método foi a seguinte:

a_{ij}^k = elemento da linha i e coluna j no início da etapa k ,

b_i^k = elemento da linha i do vetor b no início da etapa k .

Na etapa k , o objetivo é eliminar x_k nas equações $(k+1), \dots, (n)$. Assim, depois de efetuar algumas operações, tem-se:

$$\left(\begin{array}{ccccccccc} a_{11}^1 x_1 & + & a_{12}^1 x_2 & + & a_{13}^1 x_3 & + & a_{14}^1 x_4 & = & b_1^1 \\ 0 & + & a_{22}^2 x_2 & + & a_{23}^2 x_3 & + & a_{24}^2 x_4 & = & b_2^2 \\ 0 & + & 0 & + & a_{33}^3 x_3 & + & a_{34}^3 x_4 & = & b_3^3 \\ \cdot & & \cdot & & \cdot & & \cdot & & \cdot \\ \cdot & & \cdot & & \cdot & & \cdot & & \cdot \\ \cdot & & \cdot & & \cdot & & \cdot & & \cdot \\ 0 & + & 0 & + & 0 & + & a_{nn}^n x_n & = & b_n^n \end{array} \right) \quad (3.39)$$

Da última equação, tem-se:

$$x_n = \frac{b_n^n}{a_{nn}^n} \quad (3.40)$$

Assim, sucessivamente são obtidos os valores $x_{n-1}, x_{n-2}, \dots, x_1$, que é equivalente ao sistema original.

O processo de triangularização custa uma ordem de $\frac{n^3}{3}$ operações. Teoricamente, seria irrelevante o esquema de armazenamento da matriz A na resolução do sistema linear $Ax = b$ pelo Método de Eliminação de Gauss. Poderia ser utilizado o armazenamento por linhas ou por colunas, já que entre o pivoteamento e o escalonamento haveria uma espécie de “compensação” nas buscas efetuadas. Contudo, o armazenamento por colunas tem sua implementação dificultada, principalmente no escalonamento, quando as buscas por linhas se fazem necessárias para alteração, inserção ou eliminação de elementos. Em decorrência deste fato, no EGESP foi utilizado o esquema de **armazenamento por linhas**. Neste esquema, são necessários três vetores:

- valor A , que contém os coeficientes a_{ij} de A
- ICA , que contém os índices da coluna das linhas de A
- $ILA(i)$, que aponta para a posição do primeiro elemento não-nulo da linha i

Este esquema de armazenamento gasta menos memória que o armazenamento aleatório e é mais rápido que o armazenamento em valor com associação de um número que representa os índices i, j da matriz A .

Devido aos erros de arredondamento que tais sistemas estão sujeitos, optou-se por uma **estratégia de pivoteamento**⁷ que além de diminuir estes erros, não deixa que seja escolhido um pivô nulo, nem um pivô próximo de zero, durante todo o processo.

O algoritmo para o Método de Eliminação de Gauss requer o cálculo dos multiplicadores m_{ik} :

$$m_{ik} = \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}} \quad (3.41)$$

onde $i = k + 1, \dots, n$ em cada estágio k do processo.

No caso do ANIMADO, havia outro fator importante a ser considerado, a esparsidade da matriz A . Para que tal esparsidade fosse explorada, minimizando o preenchimento da matriz, dentre várias estratégias de pivoteamento existentes, optou-se pelo **pivoteamento parcial com parâmetro T** , que pode ser calculado pelo sistema ou fornecido pelo usuário.

Na estratégia de pivoteamento parcial com parâmetro T , em cada etapa k , o(s) elemento(s) pivô(s) são escolhidos da seguinte forma:

$$|pivo| \geq T \times \max_{(k \leq i \leq n)} \{|a_{ik}|\} \quad (3.42)$$

onde T pertence ao intervalo $(0,1]$

Isto significa que em cada etapa, deve-se escolher para pivô o elemento de maior módulo entre os coeficientes $a_{ik}^{(k-1)}$, $i = k, k+1, \dots, n$. Observe que pode haver a ocorrência de mais de uma candidata à linha pivotal, pois o parâmetro nos dá um intervalo que os possíveis elementos pivôs devem respeitar. Portanto, se forem utilizadas **matrizes esparsas**, dentre as possíveis linhas pivotais, é escolhida aquela que tem **o menor número de elementos diferentes de zero**, ocasionando um menor preenchimento da matriz. A escolha do maior elemento entre os candidatos a pivô faz com que os multiplicadores, em módulo, estejam entre 0 e 1, evitando a ampliação dos erros de arredondamento.

É importante ressaltar que a utilização de uma estratégia de pivoteamento é

⁷Onde adota-se um processo de escolha da linha e/ou coluna pivotal

extremamente importante porque evita que os multiplicadores sejam grandes e, com isto, **ampliem** os erros de arredondamento.

3.3.3 Integração das equações de movimento: o FRED

De posse dos valores obtidos para as acelerações do(s) objeto(s) pelo EGESP, a próxima etapa do processo será encontrar as velocidades (linear e angular) utilizando-se o FRED⁸ como Método de Integração Numérica para Resolução de **edo** com m variáveis independentes. O FRED encontrará os valores destas velocidades e será novamente realimentado com as velocidades para encontrar os valores das posições ocupadas pelo(s) objeto(s) durante o decorrer da animação.

3.3.3.1 O FRED

O FRED foi criado baseado no Método de Runge-Kutta Variante de quarta ordem. De posse dos valores das acelerações, o FRED encontrará os valores das velocidades (linear e angular) e novamente será utilizado para encontrar os valores das posições ocupadas pelo objeto no decorrer da animação. Um método direto para a integração numérica de **edo** (com valor inicial para a solução), utiliza a série de Taylor truncada em termos da r_{th} derivada da função f :

$$y_{r+h} = y_n + h y'_n + \frac{h^2}{2} y''_n + \dots + \frac{h^r y_n^{(r)}}{(r)!} \quad (3.43)$$

A truncagem do erro é $\frac{h^{r+1} z^{r+1}(\zeta)}{(r+1)!}$.

A principal desvantagem desta abordagem é que ela requer o cálculo das derivadas de $f(x, y)$ o que torna os métodos de série de Taylor computacionalmente inaceitáveis. Isto pode tornar-se complicado e demorado se f não for uma função relativamente simples. O Método de Runge-Kutta pertence ao grupo dos métodos com valor inicial que têm a mesma ordem da truncagem do erro da série de Taylor mas que envolve somente o cálculo dos valores de $f(x, y)$ e não das derivadas da função. Pode-se dizer que o método de Runge-Kutta de ordem p se caracteriza por três propriedades básicas:

⁸Ferramenta para Resolução de Equações Diferenciais

- é de passo um,
- concorda com a série de Taylor até os termos de ordem h^p ,
- não exige o cálculo de qualquer derivada de $f(x, y)$; paga, por isso, o preço de calcular $f(x, y)$ em vários pontos.

Um outro método que também poderia ser utilizado para a integração das equações é o Método de Euler que envolve somente o cálculo de $f(x, y)$ e é um Método de Runge-Kutta de primeira ordem ($p=1$). Ele é representado pela seguinte lei de formação:

$$y_{n+1} = y_n + hf(x_n, y_n), n = 0, 1, 2, \dots \quad (3.44)$$

Entretanto, este método é extremamente instável, freqüentemente levando a resultados pouco realistas, pois está muito sujeito a erros de precisão numérica [RL88].

Generalizando o Método de Runge-Kutta de quarta ordem mais usado, segue-se o procedimento abaixo:

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (3.45)$$

onde,

$$\begin{aligned} k_1 &= hf(x_n, y_n) \\ k_2 &= hf(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}) \\ k_3 &= hf(x_n + \frac{h}{2}, y_n + \frac{k_2}{2}) \\ k_4 &= hf(x_n + h, y_n + k_3) \end{aligned} \quad (3.46)$$

Neste método, a análise do erro não é trivial e por ser de quarta ordem, necessita de quatro cálculos de f . Novamente a atenção é chamada para o fato dos métodos de Runge-Kutta, apesar de serem auto-iniciáveis (pois são de passo um) e não trabalharem com derivadas de $f(x, y)$, apresentam a desvantagem de não terem uma alternativa simples para o erro o que inclusive, poderia ajudar na escolha do passo h .

3.3.4 Detecção de Colisão e a Dinâmica de Impacto: o COLISÃO

Durante uma Animação por Computador, existe a possibilidade de dois ou mais objetos se chocarem. No caso da Animação Modelada por Dinâmica, a interpenetração de dois objetos ou mais objetos não é um efeito desejável porque não representa uma realidade física, causando um efeito visual pouco agradável.

É comum a existência de sistemas de animação que não contém módulos para a Detecção de Colisão, forçando o animador a forjar visualmente um ponto aproximado de intersecção entre os objetos, simulando uma resposta aproximada de impacto. Isto consome muito tempo de CPU, é uma tarefa não-amigável e pouco inteligente. Sendo assim, serão discutidas algumas técnicas existentes para Detecção e Resposta à Colisão, bem como a estratégia usada para tal abordagem no ANIMADPO.

3.3.4.1 Detecção de colisão

A Detecção da Colisão está diretamente relacionada ao momento em que se deu o choque entre os objetos. É fácil perceber que é um processo caro de ser resolvido, especialmente nos casos em que muitos objetos estão envolvidos na animação ou quando possuem formas geométricas mais complexas. Existem inúmeros algoritmos para a Detecção de Colisão e, em geral, eles são repletos de casos específicos.

[MW88] trabalharam com um algoritmo para Detecção de Colisão onde os objetos envolvidos são poliedros convexos. Este método é baseado no algoritmo de *Clipping* de Cyrus-Beck [ROG85] que também pode ser aplicado a objetos articulados. A ordem deste algoritmo é $O(n^2m^2)$ para n poliedros e m vértices por poliedro.

Segundo [HAH88b], existem basicamente dois tipos de penetração entre objetos, ilustradas nas figuras 3.3 e 3.4. Na figura 3.3, o ponto de colisão é dado pelos vértices **internos** das arestas que interseccionam polígonos. Assumindo que o objeto 1 está parado em relação ao SC {A}, pode-se definir um vetor a partir do ponto de colisão em direção à velocidade relativa dos objetos no ponto de colisão. Se for assumido que a velocidade dos dois objetos (no ponto de colisão) permanece constante durante este intervalo, o vetor representará a trajetória que o objeto 2 fará quando penetrar o objeto 1. A intersecção

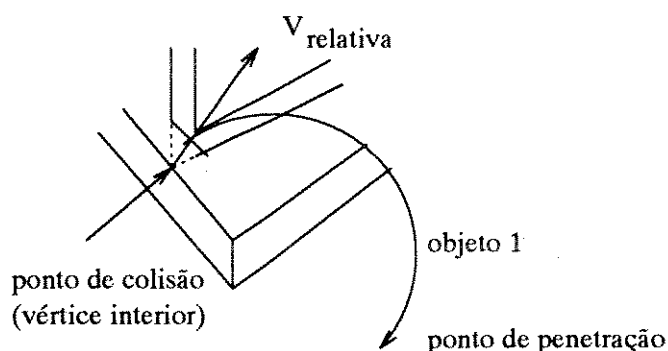


Figura 3.3: Detecção do ponto de colisão [HAH88b].

entre este vetor e os polígonos do objeto 1 representa o ponto de penetração atual no qual os dois objetos colidiram. Com o objetivo de determinar as posições e orientações atuais dos objetos no instante t , os dois objetos devem ser trazidos para o instante de colisão e, em seguida, deve-se usar a dinâmica de colisão para gerar as novas velocidades e depois, as posições e orientações no instante t . Em algumas implementações, assumindo-se um intervalo de tempo bem pequeno, o objeto 2 será trazido de volta ao tempo t tal que os dois objetos se toquem, mas não se interpenetrem.

A figura 3.4 representa uma colisão de aresta, onde uma aresta penetra mais do que um polígono. O ponto de penetração é calculado pela intersecção do polígono com a aresta, durante um certo intervalo de tempo, com as arestas dos polígonos penetrantes. O ponto de colisão é calculado encontrando-se a intersecção entre a aresta penetrante e o vetor originário a partir do ponto de penetração em direção contrária à velocidade relativa.

Este algoritmo assume que o intervalo de tempo e/ou a velocidade dos polígonos dos objetos são suficientemente pequenas para que a distância coberta durante este intervalo seja muito menor do que as dimensões dos polígonos. Se este não for o caso, então, o volume das *bounding boxes* e os polígonos devem ser considerados no algoritmo de detecção de colisão para não perder a colisão. Existem outras diversas formulações, mais adiante, será descrita a formulação implementada no ANIMADO.

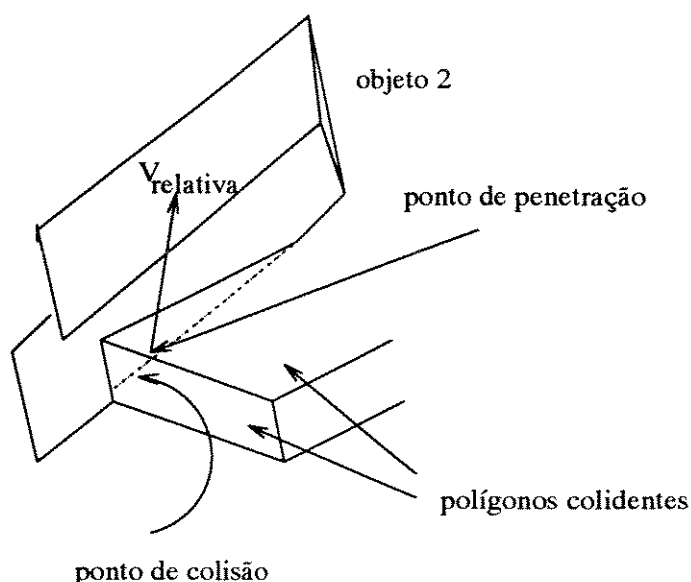


Figura 3.4: Detecção da aresta de colisão [HAH88b].

3.3.4.2 Resposta à colisão

No momento do choque, as forças internas envolvidas são muito intensas e, portanto, as eventuais forças externas (como por exemplo, a gravidade) são desprezíveis durante um breve intervalo de tempo. O sistema é considerado um sistema isolado. Estas forças internas interagem entre si obedecendo a Lei da Ação e Reação que diz: “Os impulsos das forças internas se neutralizam dois a dois e, como no sistema isolado a resultante externa é nula, o impulso total no sistema isolado é nulo” [SYM71]. Desta forma, a variação da quantidade de movimento do sistema permanece constante.

O primeiro pesquisador a propor uma análise do processo de impacto entre dois corpos rígidos foi [TROU05] que incluiu o efeito de Coulomb no modelo de fricção e a presença de materiais elásticos. Até hoje, sua proposta foi pouquíssimo modificada e ainda é a mais utilizada. Esta análise utiliza Técnicas de Solução Numérica **antes** e **depois** da colisão.

Pode-se caracterizar a dissipação de energia de uma colisão especificando um número chamado e (coeficiente de restituição) que representa a razão das velocidades relativas antes e depois da colisão.

O valor do coeficiente de restituição depende dos materiais que compõem os

corpos colidentes e como os coeficientes de atrito, não é realmente calculado, mas sim, determinado empiricamente para cada par de corpos colidentes. O valor de ϵ varia de (0,1). Para choques **completamente elásticos** onde nenhuma energia é perdida, ϵ é igual a 1, para choques **perfeitamente inelásticos**, onde toda a energia é perdida, ϵ é igual a zero, e para choques **parcialmente elásticos**, onde alguma energia é perdida ϵ está entre 0 e 1. Em geral, as colisões não são perfeitamente elásticas, nem completamente inelásticas, mas podem enquadrar-se nestes dois extremos, ou seja, há alguma energia cinética perdida durante o choque.

Considerando as massas dos dois objetos envolvidos na colisão (m_1 e m_2), as velocidades lineares iniciais (v_1 e v_2), as velocidades angulares iniciais (w_1 e w_2), os momentos de inércia (I_1 e I_2) e o coeficiente de restituição (ϵ) como dados, as velocidades finais de cada objeto serão encontradas usando a Lei de Conservação do Movimento. Resolve-se um sistema de equações achando o valor das velocidades finais em função das velocidades iniciais, das massas e do ϵ .

O tipo de algoritmo empregado para a modelagem da resposta à colisão dependerá do tipo de objetos envolvidos no choque. Entre os mais utilizados estão os algoritmos baseados na inserção de “molas” (força elástica) e os algoritmos baseados diretamente na resolução analítica do problema. Soluções analíticas são tipicamente mais rápidas em colisões bruscas porque a solução deverá ser encontrada uma única vez. Entretanto, para colisões amenas, a inserção de “molas” é mais aconselhável. Neste caso, a gravidade pode causar alguma interpenetração entre os dois objetos, e a solução analítica deverá ser aplicada diversas vezes. Uma simples mola que neutralize a gravidade é mais rápida e mais estável, sendo mais apropriada neste caso. Estas duas abordagens serão detalhadas a seguir.

3.3.4.3 Resposta à colisão utilizando força elástica

Este é o método mais intuitivo para a modelagem da colisão [MW88]. Na realidade, corresponde a um simples “truque” que consegue simular visualmente a elasticidade entre os corpos envolvidos no choque, com bom nível de realismo. A idéia é inserir uma pequena “mola” no ponto de colisão. Este tipo de modelagem é muito freqüente em colisões onde um dos objetos envolvidos é do tipo “chão”, ou seja, um objeto de massa infinita que permanecerá sem movimento independentemente do choque. Aliado à constante de

elasticidade também é usada uma constante de amortecimento do choque.

Dadas as posições atuais e os graus de liberdade dos objetos, as forças de interação são assim calculadas:

$$\begin{aligned} F_{elastica} &= k_{elastica} \rho \\ F_{amortecimento} &= k_{amortecimento} v \end{aligned} \quad (3.47)$$

onde,

$k_{elastica}$ e $k_{amortecimento}$ são constantes do modelo,

ρ corresponde à distância do ponto de colisão ao centro de massa do objeto,

e v é a velocidade do objeto.

3.3.4.4 Resposta à colisão utilizando solução analítica

Neste caso, o tipo de colisão está diretamente relacionado à forma do objeto envolvido no instante do choque. A colisão deve ser detectada e as forças de reação devem ser encontradas antes da Dinâmica atuar no objeto, respondendo à esta colisão. Para corpos simples, pode-se calcular os efeitos da colisão usando soluções analíticas.

A solução analítica para o problema depende da conservação do momento durante a colisão, e tem como produto final a velocidade linear e angular finais de cada objeto. Nesta seção será apresentada uma solução analítica geral para a colisão de dois corpos rígidos sugerida por [MW88].

Seguindo a figura 3.5, a cada objeto é associada uma velocidade linear (v_i), uma velocidade angular (w_i), uma massa (m_i), um centro de massa (c_i) e uma matriz do tensor inercial (I_i) relativa ao centro de massa do corpo. Todos estes valores (para ambos os objetos) devem ser expressos em relação ao mesmo SC $\{A\}$. Tão logo for detectada uma colisão, cada objeto terá um vetor associado ρ_i , que aponta do seu centro de massa ao ponto de colisão. Além destes valores, será necessário três vetores unitários ortogonais (i, j, k) que definirão o “SC de colisão”, k será perpendicular ao plano de colisão, i e j pertencerão a este plano.

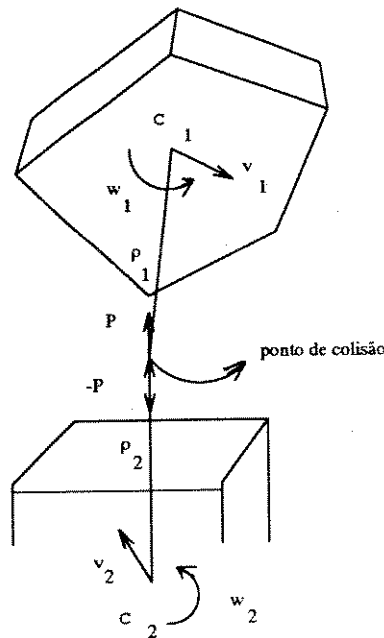


Figura 3.5: Um modelo de colisão entre dois corpos rígidos [MW88].

A definição do plano de colisão é arbitrária. No caso do ANIMADO define-se o seguinte: Caso um vértice do objeto colidir com a face do outro objeto, então, a face definirá o plano de colisão. Se uma aresta de um objeto colidir com uma aresta de outro objeto, estas duas arestas definem o plano de colisão. Se os dois vértices colidirem, k apontará na direção da reta que os une.

Assumindo a existência de um único ponto de colisão, a solução para a dinâmica da colisão dependerá da resolução de quinze equações lineares, com quinze incógnitas: nova velocidade linear para cada objeto (v'_1, v'_2), nova velocidade angular para cada objeto (w'_1, w'_2) e um vetor impulso P . Por convenção, o impulso terá direção dada do objeto 2 para o objeto 1.

As equações lineares que representam a mudança do momento angular e linear de cada objeto, devido ao impulso P , são as seguintes:

$$\begin{aligned}
m_1 v_1' &= m_1 v_1 + P \\
m_2 v_2' &= m_2 v_2 - P \\
I_1 w_1' &= I_1 w_1 + \rho_1 \otimes P \\
I_2 w_2' &= I_2 w_2 - \rho_2 \otimes P
\end{aligned} \tag{3.48}$$

Desenvolvendo estas equações, teremos doze equações lineares:

$$\begin{aligned}
P_x &= m_1(v_{1x}' - v_{1x}) \\
P_y &= m_1(v_{1y}' - v_{1y}) \\
P_z &= m_1(v_{1z}' - v_{1z}) \\
P_x &= m_2(v_{2x} - v_{2x}') \\
P_y &= m_2(v_{2y} - v_{2y}') \\
P_z &= m_2(v_{2z} - v_{2z}') \\
I_1(w_{1x} - w_{1x}') &= -P_z y_1 - (-P_y) z_1 \\
I_1(w_{1y} - w_{1y}') &= -P_x z_1 - (-P_z) x_1 \\
I_1(w_{1z} - w_{1z}') &= -P_y x_1 - (-P_x) y_1 \\
I_2(w_{2x} - w_{2x}') &= P_x y_2 - (-P_y) z_2 \\
I_2(w_{2y} - w_{2y}') &= P_y z_2 - (-P_z) x_2 \\
I_2(w_{2z} - w_{2z}') &= P_z x_2 - (-P_x) y_2
\end{aligned} \tag{3.49}$$

Onde as incógnitas são:

$$w_{1x}', w_{1y}', w_{1z}', w_{2x}', w_{2y}', w_{2z}', v_{1x}', v_{1y}', v_{1z}', v_{2x}', v_{2y}', v_{2z}', P_x, P_y \text{ e } P_z.$$

Outras três equações referentes às condições da colisão também serão consideradas. Para um choque inelástico ($\epsilon = 0$), onde pode ocorrer deslize entre as superfícies (o impulso deverá ser perpendicular ao plano de colisão). Neste caso, considera-se que o produto interno de P com i e j será zero e a diferença de velocidade do ponto de colisão (quando vista de um dos dois objetos) será zero na direção k .

A partir da conhecida **Regra de Newton Generalizada**, que é um resultado empírico para a situação do choque, conhecido como coeficiente de elasticidade ϵ , tem-se:

$$-\epsilon = \frac{(v'_1 + w'_1 \otimes \rho_1)N - (v'_2 - w'_2 \otimes \rho_2)N}{v_1 + w_1 \otimes \rho_1)N - (v_2 + w_2 \otimes \rho_2)N} \quad (3.50)$$

Então:

$$\begin{aligned} P_i &= 0 \\ P_j &= 0 \\ (v'_2 + w'_2 \otimes \rho_2 - v'_1 - w'_1 \otimes \rho_1)k &= 0 \end{aligned} \quad (3.51)$$

Para uma colisão elástica ($\epsilon = 1$), escolhe-se o menor coeficiente de elasticidade entre as duas superfícies envolvidas. Este coeficiente escolhido será considerado como o coeficiente de elasticidade da colisão e será utilizado para encontrar o novo impulso (P_{real}) da colisão, através da seguinte equação:

$$P_{real} = (1 + \epsilon_{real})P \quad (3.52)$$

onde $\epsilon_{real} = \min(\epsilon_1, \epsilon_2)$.

Este novo impulso P_{real} , é substituído nas equações já mencionadas para encontrar os valores de v'_i e w'_i . Os vetores v'_i são diretos. Os w'_i são encontrados mediante a inversão de I_i .

Para superfícies **sem deslizamento**, se existe atrito infinito entre os objetos e a **colisão for inelástica** ($\epsilon = 0$), então:

$$v'_2 + w'_2 \otimes \rho_2 - v'_1 - w'_1 \otimes \rho_1 = 0 \quad (3.53)$$

A modelagem realista do atrito pode tornar-se extremamente complexa. Segundo [MN78] a abordagem explicitada a seguir é suficiente para produzir resultados realistas razoáveis. Nesta abordagem, considera-se o coeficiente de fricção μ como uma proporção máxima entre a força tangencial ao plano de colisão e a força normal ao plano de colisão. Embora μ seja uma propriedade peculiar das superfícies, nesta abordagem, escolhe-se o maior μ entre os objetos colidentes.

Para ($\epsilon = 0$) e μ finito, resolve-se a colisão em duas etapas:

1. Primeiramente resolve-se a colisão como se ela fosse totalmente inelástica (atrito infinito) e encontra-se o impulso resultante P .
2. Se a seguinte condição for satisfeita:

$$\begin{aligned}\mu P k &\geq |P - k(P k)| \\ \mu &\geq \frac{|P - k(P k)|}{P k}\end{aligned}\quad (3.54)$$

Então os objetos devem ficar parados, caso contrário, os objetos devem deslizar. Novas condições de colisão são definidas e o sistema de equações é novamente resolvido. Estas novas condições originarão uma força tangencial de restrição devido ao fato de que somente uma quantidade limitada de fricção pode atuar contra o movimento de deslize. Duas constantes, α e β são calculadas, tal que o impulso de colisão satisfaça:

$$\begin{aligned}\mu P k &= |P - k(P k)| \\ \mu &= \frac{|P - k(P k)|}{P k}\end{aligned}\quad (3.55)$$

Ou seja, tal que a razão entre as componentes tangencial e normal do impulso seja exatamente μ e a direção da componente tangencial de P permaneça inalterada. Este procedimento encontra o valor da força tangencial máxima permitida pela força perpendicular (necessária para a resposta ao choque) e o coeficiente de atrito. Estas condições de colisão são:

$$\begin{aligned}P_i &= \alpha P_k \\ P_j &= \beta P_k \\ (v'_2 + w'_2 \otimes \rho_2 - v'_1 - w'_1 \otimes \rho_1) k &= 0\end{aligned}\quad (3.56)$$

onde,

$$\begin{aligned}\alpha &= \mu (P_t i) \\ \beta &= \mu (P_t j) \\ P_t &= \frac{P - P_n}{|P - P_n|} \\ P_n &= k(P k)\end{aligned}\quad (3.57)$$

P_n = componente normal de P , perpendicular ao plano de colisão,

P_t = vetor direção unitário de P , paralelo ao plano de colisão.

Resumindo, o algoritmo de resposta à colisão de dois corpos rígidos baseia-se no seguinte procedimento:

- *Transforme $(v_1, v_2, w_1, w_2, I_1, I_2, \rho_1, \rho_2)$ do SC $\{B\}$ para o SC $\{A\}$*
- *Defina (i, j, k) , ou seja, o plano de colisão*
- *Escolha condições de Colisão*
- *Estabeleça e Resolva o Sistema de Equações*
- *Transforme (v'_1, v'_2, w'_1, w'_2) do SC $\{A\}$ para o SC $\{B\}$*

Pode ser que este método pareça trabalhoso se comparado com o método que utiliza a inserção de uma força elástica. Além disso, ele requer um maior número de linhas de código de programação. Contudo, este método é costumeiramente aplicado uma única vez em cada colisão, enquanto que o algoritmo que utiliza **força elástica** geralmente deve ser aplicado muitas vezes, em intervalos bem pequenos de tempo. Para concluir, o método analítico é mais barato computacionalmente principalmente se o custo da colisão não depender da violência do choque (por exemplo, em colisões muito suaves). Sem dúvida, uma propriedade muito desejada nas colisões.

Como já mencionado, o objetivo do COLISÃO no ANIMADO, é produzir um movimento razoavelmente realista (evitando que os corpos envolvidos no processo se interpenetrem) e economizando tempo do animador caso o seu interesse seja o verificar rapidamente a possibilidade ou não, da ocorrência de um choque (utilizando *bounding boxes*). Para que tal objetivo fosse concretizado, este módulo de colisão foi dividido em dois problemas distintos e independentes:

- Deteção da colisão (problema cinemático)
- Resposta à colisão (problema dinâmico)

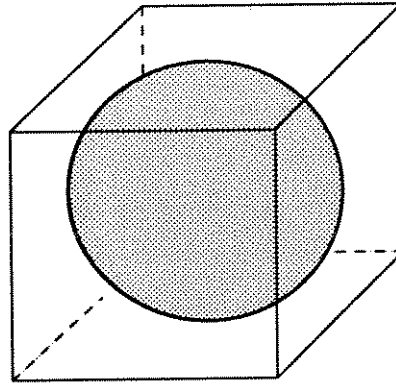


Figura 3.6: Um exemplo de *bounding box*.

Como já foi dito, existem diversas formulações para a detecção de colisão. Com o objetivo de utilizar os trabalhos gerados pelo próprio grupo ProSim, o trabalho [MAD93] foi aperfeiçoado nos casos de intersecção avaliados e otimizado com a criação de *bounding box*, podendo então, ser utilizado para a Detecção de Colisão no ANIMADO.

Basicamente, o algoritmo verifica a existência de intersecção entre dois objetos, otimizada utilizando-se *bounding boxes* como mostra a figura 3.6. Com isto, o número de testes de polígonos de intersecção é minimizado, encontrando-se uma “não-colisão” entre os objetos com muita rapidez. Este fato é simples de ser observado: quando um ponto é testado contra um poliedro, ele deverá ser primeiramente testado contra a *bounding box* do poliedro. Na maioria dos casos, elimina-se a necessidade de comparar este ponto com todas as faces do poliedro, agilizando o algoritmo.

Contudo, como já foi dito, os algoritmos para a Detecção de Colisão são complexos, dependentes da geometria do objeto e na maioria das vezes, não são totalmente abrangentes. O mesmo problema foi observado no algoritmo utilizado no ANIMADO que está limitado ao estudo do seguinte caso: no nível mais alto da hierarquia, cada aresta do objeto é testada contra cada face do outro objeto. Se a aresta interceptar a face em algum ponto, acrescenta-se um vértice na estrutura de dados do objeto. [MAD93] criou um Sistema de Modelagem Geométrica de Sólidos onde podem ser realizadas as operações de soma, subtração e intersecção entre objetos. As rotinas que analisam a existência de intersecção entre dois sólidos são restritas e os casos degenerados, não são tratados: aresta pertencente à face, ponto pertencente à face, objetos coincidentes, intersecção de vértices, etc. Em

decorrência da vinculação do módulo de detecção de colisão do ANIMADO a [MAD93], o COLISÃO torna-se dependente da complexidade geométrica do objeto. Em se tratando dos casos degenerados, sua dependência será a nível do posicionamento dos objetos.

Atualmente, [MAD93] está sendo reavaliado [FER93] para uma classe mais abrangente de objetos para ser adaptado e integrado ao Sistema de Modelagem Phoenix [LT91].

No ANIMADO, o próprio sistema responderá à colisão, determinando a nova velocidade linear e angular dos objetos colidentes. Em qualquer tipo de colisão, suposta ser um sistema isolado, há a conservação do momento linear e angular total do sistema. O coeficiente de elasticidade dos objetos envolvidos também deve ser levado em consideração e ele determina quanto de energia cinética é gasta neste processo.

No ANIMADO, a resposta à colisão foi modelada utilizando-se os dois métodos já citados nos itens 3.3.5.3 e 3.3.5.4. O primeiro baseia-se na introdução de “molas” temporárias no ponto de colisão, o segundo, nada mais é do que a **solução analítica do sistema linear** que representa a colisão. O método baseado na existência de uma força elástica (“molas”) é mais geral e possui o mesmo desempenho independentemente do tipo de objeto envolvido na colisão (objetos flexíveis, rígidos ou articulados). O método baseado na solução analítica do sistema linear é bem mais rápido e aplica-se somente a objetos rígidos e articulados. Sendo assim, como esta versão do ANIMADO tratará somente objetos rígidos, os dois métodos poderão ser utilizados. No caso da colisão modelada como um sistema de equações, o EGESP pode ser utilizado como ferramenta para resolução do sistema.

No apêndice c desta dissertação serão descritas algumas situações onde ocorrem interações entre objetos com o intuito de modelar **restrições no movimento** (não-interpenetrar) e resolver a Dinâmica de Impacto do movimento utilizando-se o COLISÃO. No ANIMADO, foram implementados três tipos de colisão (queda-livre e lançamento oblíquo de uma esfera que colide com um objeto do tipo “chão” usando **força elástica**, e a colisão de uma esfera e um cubo baseada na solução analítica do sistema de equações modelado), também citadas no apêndice a desta dissertação.

3.3.5 Controle das equações de movimento e dos módulos: o CONTROLE

O CONTROLE se responsabilizará pelo envio de mensagens e sinais entre os módulos que compõem o sistema. Caso ocorram colisões, ele determinará qual será a próxima etapa a ser executada pelo sistema. Depois da análise de colisão, se a velocidade relativa dos objetos no ponto de colisão (na direção da normal ao ponto de contato local) for **menor** do que uma certa taxa, pode-se considerar que ocorreu contato contínuo entre os objetos. Se os objetos estão aplicando uma força mutuamente (se um dos objetos for o chão), a dinâmica ainda pode ser simulada usando as equações de impacto citadas no item 3.3.8.3 deste trabalho.

Embora as forças e torques não apareçam explicitamente, suas contribuições são vistas como um ganho no momento do objeto durante um intervalo de tempo que corresponde aos impulsos aplicados nos objetos no momento do impacto. O controle de corpos rígidos pode ser visto como uma combinação de torques e forças aplicados ao sistema. Os torques causam um movimento rotacional ao redor do SC {B} e necessitam de um vetor $3D$. As forças causam um movimento translacional em relação ao SC {B} e também necessitam de um vetor $3D$ que descreve onde a força está sendo aplicada. Geralmente, a localização destes vetores será especificada em termos do SC {B}.

A força total será encontrada somando-se os vetores força **independentemente do ponto de aplicação**. O torque total é encontrado através da soma dos valores obtidos pela equação 3.34 (calculando os torques causados por estas forças) e por torques puros. Estes seis valores serão utilizados nas seis equações de movimento. O controle também pode ser fornecido como equações de restrição extras que limitam os GLs envolvidos na animação.

Uma variedade de forças e torques contribuem para o movimento, algumas das quais podem ser calculadas automaticamente, outras, fornecidas pelo usuário. As forças e torques que podem ser computados automaticamente no ANIMADO incluem àquelas devido à gravidade, a colisões entre partes de corpos e contato com objetos externos, tais como o chão (que tem massa infinita) ou outros objetos do meio ambiente cuja posição e comportamento são conhecidos pelo sistema.

As forças e torques fornecidos como dados de entrada para o sistema incluem forças e torques internos que serão chamados de **forças e torques de controle**. Outras

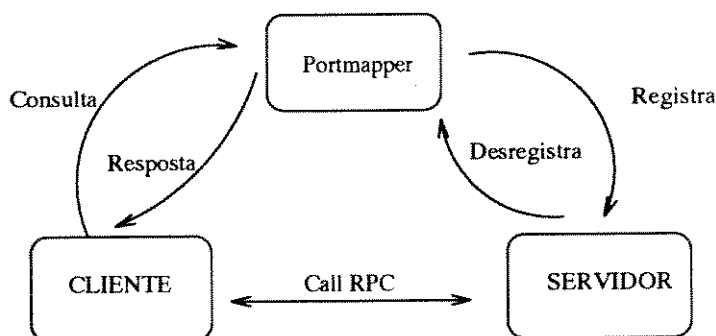


Figura 3.7: Esquema geral de comunicação entre portmapper, cliente e servidor [SUN88].

forças e torques são calculadas (somando-se suas contribuições em cada corpo) compondo uma força total resultante.

3.3.6 Comunicação entre os módulos: o INTEREXPSERV

Tendo como plataforma de trabalho o ambiente Unix, possuidor de recursos para a criação e manipulação de *sockets*⁹, o INTEREXPSERV¹⁰ foi criado para capacitar a comunicação entre os diversos módulos do ANIMAD_O, como mostra a figura 3.2. O INTEREXPSERV executa as tarefas de comunicação entre todas as partes envolvidas no sistema (faz o esquema do servidor), interpreta expressões e armazena dados.

Os módulos que compõe o ANIMAD_O correspondem a processos que se comunicam através de **chamadas de procedimentos remotos**¹¹. A comunicação entre os processos é baseada no modelo cliente/servidor, ou seja, na comunicação entre processos que necessitam de algum serviço e outros que realizam estes serviços. As requisições de serviços por parte dos clientes e o cadastramento destes serviços por parte dos servidores são administradas por um processo chamado *portmapper*¹² [NE92], como ilustra a figura 3.7.

⁹*socket* é o bloco básico de comunicação em rede através de mecanismos de transporte baseados em fluxos[SUN88].

¹⁰INTERpretador de EXPressões e SERVidor

¹¹RPC = Remote Procedure Call

¹²Um mapeador de portas

3.3.6.1 A comunicação entre processos

A comunicação entre processos [SUN88] é realizada através de acessos orientados à uma conexão que é estabelecida, quando então, se evidenciam as características da troca de informação que serão usadas durante a fase de transmissão de dados. Este tipo de acesso é apropriado também para serviços em que uma transmissão depende de transmissões previamente realizadas. Também tem boa aplicação na transmissão de dados de um arquivo, pois as várias mensagens que transportam partes do arquivo possuem uma dependência entre si e pode-se estabelecer um contexto a fim de definir algumas características, isto é: nome do arquivo, interpretação dos conteúdos das mensagens, etc.

Os domínios de comunicação servem para interligar propriedades comuns entre os processos de comunicação correspondendo ao esquema usado para nomear um *socket*. Geralmente, a troca de dados somente ocorre com *sockets* no mesmo domínio.

Os acessos orientados à conexão são os mais confiáveis. Além disto, nos acessos orientados à conexão, as mensagens são recebidas na mesma seqüência de envio. Outra peculiaridade é o acesso orientado à conexão, tanto antes como após à transmissão. Presume-se que os processos só se comuniquem entre *sockets* do mesmo tipo, embora isso não possa ser totalmente garantido. O estabelecimento de uma conexão *stream* é realizada no ambiente Unix como mostra a figura 3.8. Este tipo de *socket* prevê o fluxo de dados bidimensional, confiável, sequenciado e não-duplicado [NE92].

3.3.6.2 O modelo cliente/servidor

O paradigma mais comumente usado na construção de aplicações distribuídas é o modelo cliente/servidor. Neste esquema, o cliente requisita serviços para um processo servidor, fato que implica numa assimetria no estabelecimento de comunicação entre o cliente e o servidor. Tanto o cliente, quanto o servidor, requerem um conjunto bem conhecido de convenções antes do serviço ser dado (e aceito). Este conjunto de convenções compreende um protocolo que deve ser implementado em ambos os pontos extremos de uma conexão. Dependendo da situação, o protocolo pode ser simétrico ou assimétrico. No protocolo simétrico, qualquer uma das partes envolvidas pode ser o mestre ou o escravo na execução das tarefas. No protocolo assimétrico, um dos lados é fixado como o mestre e o outro, como

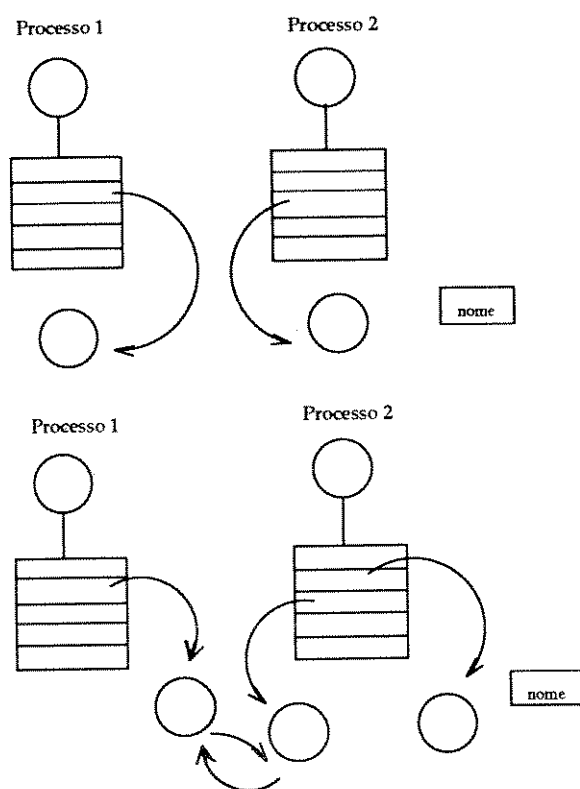


Figura 3.8: Estabelecimento de uma conexão do tipo *stream* [SUN88].

escravo. Independentemente desta simetria ou não, quando houver um serviço de acesso, haverá um **processo cliente** e um **processo servidor**.

Um processo servidor permanece dormiente até que uma conexão seja requisitada por um cliente, via um endereço do servidor. Então, o processo servidor “acorda” e atende o cliente. É interessante ressaltar que um servidor específico atende somente um número de porta restrito que se refere ao processo requisitado pelo cliente.

O servidor é uma entidade passiva, que atende a chamadas dos clientes, enquanto o processo cliente é uma entidade ativa, que inicia uma conexão quando invocado. O primeiro passo para que o processo de comunicação aconteça é localizar a definição do serviço para um login remoto. Em seguida, é identificado o hospedeiro destino. Então, estabelece-se uma conexão para o servidor no hospedeiro requisitado e começa o protocolo do login remoto. O endereço do *buffer* é limpo, depois preenchido com o endereço do hospedeiro forasteiro e o número da porta no qual o processo login reside no hospedeiro forasteiro. Um *socket* é criado e uma conexão estabelecida.

3.3.6.3 Implementação do INTEREXPSERV

O interpretador de expressões utilizado no ANIMADO foi implementado baseado em *sockets*. No caso do INTEREXPSERV ao invés de utilizar um nome para definir o *socket*, foi utilizado o endereço inthetnet e a porta de comunicação que torna o nome do *socket* único. Cada *socket* ativo tem um **tipo** e um ou mais processos de comunicação associados que existem dentro dos **domínios de comunicação**. Os domínios de comunicação são introduzidos para interligar propriedades comuns entre os processos de comunicação através dos *sockets*. Estas propriedades correspondem ao esquema usado para nomeá-los. No INTEREXPSERV optou-se pelo uso do *socket* do tipo *stream* ilustrado na figura 3.8.

No ANIMADO, espera-se que um dos módulos inicie o processo. O INTEREXPSERV será o processo servidor que esperará que qualquer um dos outros módulos clientes requisite um serviço para ele. Quando requisitado, ele dará atenção especial a um cliente de cada vez. O CONTROLE será o cliente responsável pela iniciação da comunicação, sua tarefa será a de definir o tempo inicial e final da animação e cuidar da discretização do tempo na integração, quando necessário. Também será responsável pelas mudanças relativas às novas

forças que atuarão nos objetos que estão sendo animados quando houver uma colisão. A comunicação dar-se-á seguindo o esquema ilustrado na figura 3.9.

3.3.7 Introdução de restrições físicas no modelo

Caso exista o interesse em se modelar movimentos que sejam limitados por certos fatores, como por exemplo, movimentos sujeitos à resistência do ar, gravidade, forças de atrito atuando nos objetos, etc., o animador poderá realizar este intuito dependendo da sua habilidade em modelar o mundo físico existente.

Um exemplo bastante simples de restrição no movimento, é a atuação da força da gravidade, que nada mais é do que um tipo de força de atrito. Dada a aceleração gravitacional ($\approx 9.81 \text{ m/s}^2$ na superfície terrestre), a força da gravidade atuante no centro de massa de um corpo rígido c é:

$$\begin{aligned} f_{gr} &= (0, -9.81, 0)m \\ \Rightarrow f_{grx} &= 0 \\ f_{gry} &= -9.81m \\ f_{grz} &= 0 \end{aligned} \tag{3.58}$$

O torque devido à esta força, atuando num SC {B} fixo no corpo rígido é:

$$\tau_{gv} = c \otimes f_{gr} \tag{3.59}$$

que expandido origina:

$$\begin{aligned} \tau_{grx} &= f_{grz} c_y - f_{gry} c_z \\ \tau_{gry} &= f_{grx} c_z - f_{grz} c_x \\ \tau_{grz} &= f_{gry} c_x - f_{grx} c_y \end{aligned} \tag{3.60}$$

de 3.58 tem-se:

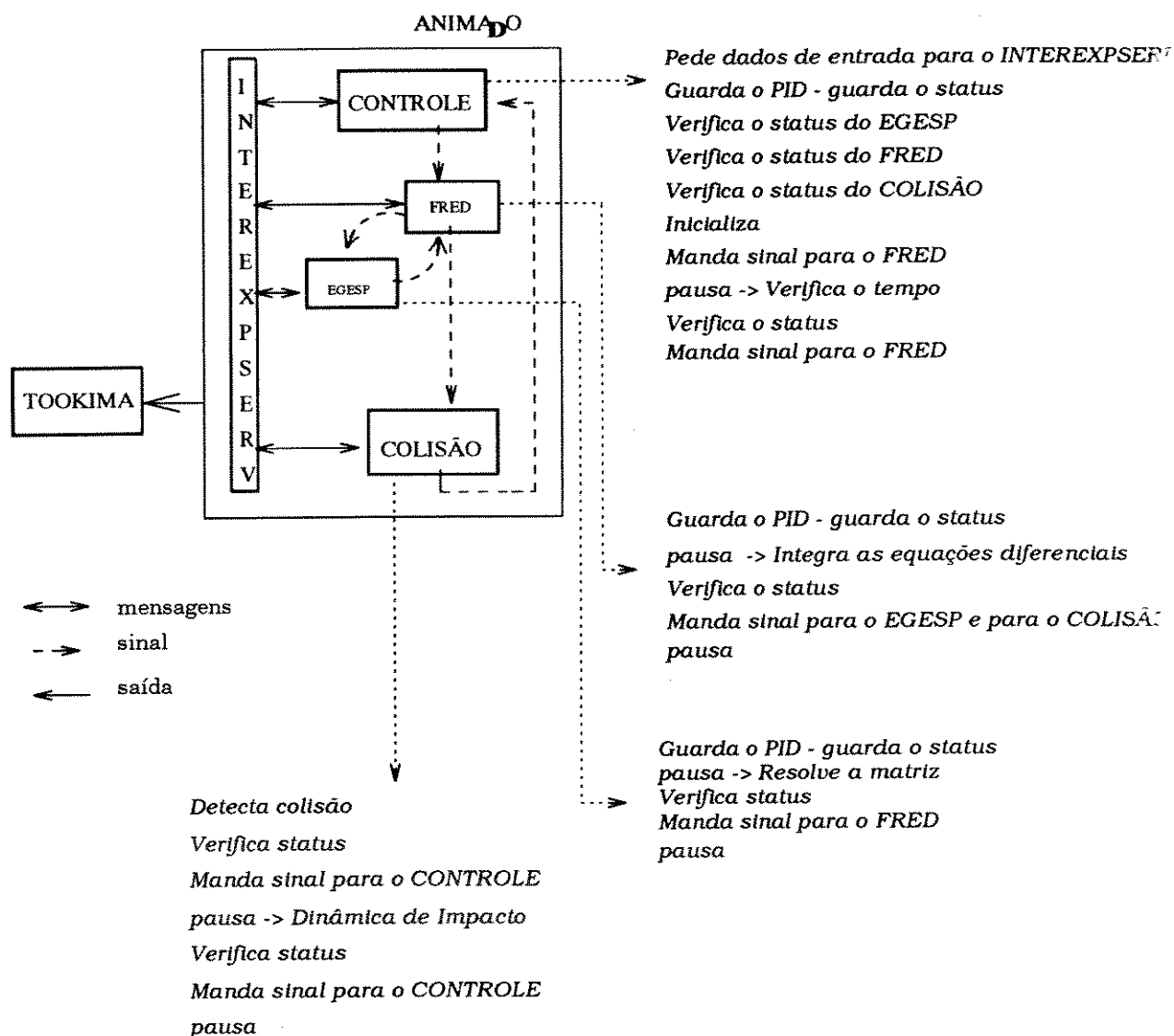


Figura 3.9: A comunicação no ANIMADO.

$$\begin{aligned}
\Rightarrow \tau_{grx} &= +9.81 \, m \, c_x \\
\tau_{gry} &= 0 \\
\tau_{grz} &= -9.81 \, m \, c_x
\end{aligned} \tag{3.61}$$

3.3.8 Reprodução da Cinemática

As posições nos respectivos instantes de tempo correspondentes serão dados de saída do ANIMADO, que poderá utilizar o TOOKIMA como ferramenta somente para a inicialização de parâmetros ambientais (luz, observador, câmera, modelo de tonalização, cor dos objetos, etc) e para o posicionamento do(s) objeto(s) de acordo com os dados fornecidos pelo ANIMADO, ou seja, os objetos sofrerão a influência de forças e torques. Os torques τ produzirão um movimento de rotação sobre a origem do SC{B} do objeto e as forças f produzirão um movimento de translação relativa ao SC{B}.

3.3.9 O funcionamento do ANIMADO

No ANIMADO, as etapas para a produção da Animação Modelada por Dinâmica interagem seguindo o esquema referenciado pela figura 3.10.

Na fase de inicialização do ANIMADO, cada objeto poderá apresentar características físicas, tais como: coeficiente de fricção, coeficiente de restituição, massa, etc. Em seguida, são calculadas outras propriedades físicas: centro de massa, tensor momento de inércia, eixos principais, etc.

O estado dinâmico de cada objeto descreve a velocidade linear, a velocidade angular, a posição e a orientação atual do objeto. O estado dinâmico corrente é usado para a obtenção do próximo estado dinâmico num intervalo de tempo próximo ($t + dt$). A atualização é feita em três níveis:

1. As acelerações angular e linear dos objetos são encontradas usando um Método para Resolução de Sistemas Lineares Esparsos (no caso, o EGESP) de acordo com a Formulação de Euler. O estado dinâmico corrente é utilizado para a obtenção destas acelerações.

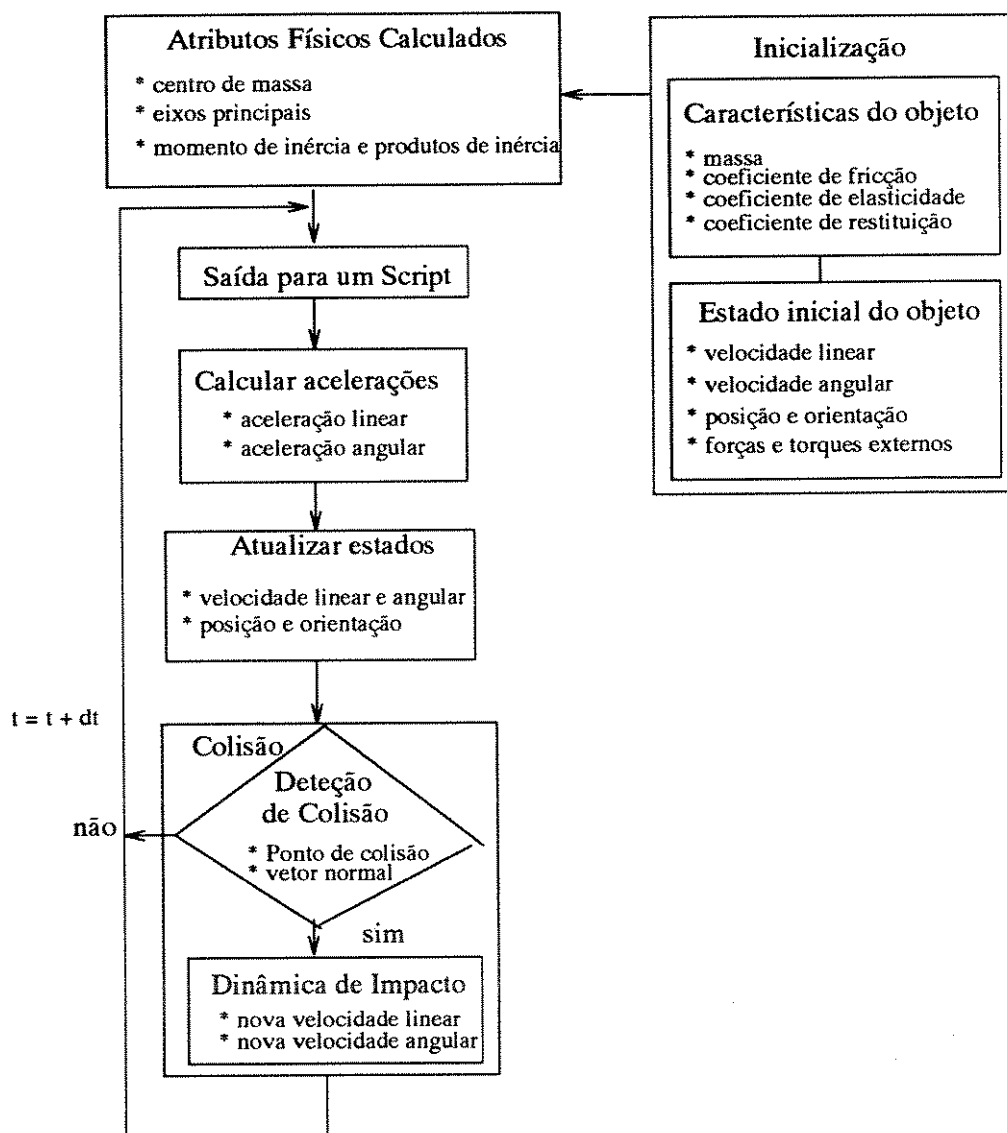


Figura 3.10: Esquema de funcionamento do ANIMAD0.

2. Os objetos são movidos usando o estado dinâmico corrente. Isto envolve a Integração de **edo** (no caso, o **FRED**) para a obtenção das posições e orientações dos objetos no próximo intervalo de tempo. Se o movimento do objeto for por meio de um arquivo de descrição (*script*), então, o estado pode ser lido ali.
3. Finalmente, com o auxílio do **COLISÃO** checa-se a existência de intersecção entre os objetos. Se o contato ocorreu, o novo estado dinâmico dos objetos que foram afetados pela colisão é calculado usando Dinâmica de Impacto.

Capítulo 4

Conclusão

Sabe-se que a Animação Modelada por Dinâmica consome muito tempo de CPU e que necessita de ferramentas capazes de suportar cálculos intensivos, com boa precisão numérica. A Animação Modelada por Dinâmica também requer um conhecimento das leis físicas que regem o nosso universo para efeitos da modelagem da animação. Caso contrário, o processo físico não será modelado com fidelidade, prejudicando o realismo da animação.

Com relação ao custo computacional envolvido nesta abordagem, devido aos cálculos intensivos que o sistema realiza (que acarretam problemas numéricos na abordagem por Dinâmica), a Animação Modelada por Dinâmica acaba se tornando mais cara do que a Modelada por Cinemática.

No estudo de colisões, dentro da abordagem da Animação Modelada por Dinâmica, detectar a ocorrência de uma colisão pode se tornar uma tarefa extremamente complicada, dependendo da complexidade dos objetos envolvidos no choque. Responder à colisão também é uma tarefa complexa onde cada modelo de colisão terá um novo conjunto de características bastante peculiares onde serão adicionadas novas restrições.

Com a finalidade de estudar os pontos acima mencionados, foi implementado o **ANIMADO**, um primeiro protótipo de um Sistema de Animação Modelada por Dinâmica, criado no Departamento de Engenharia de Computação e Automação Industrial, cujo objetivo é introduzir a Simulação Dinâmica dentro de um ambiente de Animação por Computador. Composto por cinco módulos totalmente independentes o EGESP, o FRED, o COLISÃO,

o **CONTROLE** e o **INTEREXPSERV** que trabalham unidos para gerar a animação desejada. Atualmente, estes módulos estão atuando numa mesma estação gráfica consumindo muita CPU. Entretanto, o paralelismo destas ferramentas permite que cada uma delas possa ser distribuída via rede, por exemplo, aproveitando a ociosidade de alguma máquina, duplicando ferramentas, ou seja, executando suas respectivas funções em cinco máquinas diferentes, como uma possível possibilidade de ganhar tempo de processamento.

O **ANIMADO** tem grande flexibilidade e precisão numérica, é de fácil utilização pois as rotinas menos “amigáveis” são invisíveis ao usuário. De acordo com sua estruturação, permite a existência de uma interface bastante simples de ser implementada e manuseada, pois não requer conhecimento prévio de programação.

No âmbito de Detecção de Colisões o **ANIMADO** dispõe de uma ferramenta baseada numa extensão de [MAD93] permitindo o tratamento dos casos de intersecção entre objetos onde cada aresta é testada contra cada face do outro objeto. Se a aresta interceptar a face em algum ponto, acrescenta-se um vértice na estrutura de dados do objeto.

Duas alternativas são oferecidas ao usuário para o tratamento de colisões: o modelo de resposta à colisão utilizando força elástica e o modelo de resposta à colisão via resolução de sistemas de equações, descritos no item 3.3.5.3 e 3.3.5.4 respectivamente. Dentro do Módulo de Dinâmica de Impacto associada à Colisão, o método mais óbvio, geral e fácil de programar é o que utiliza a inserção de uma força elástica. Contudo, não é muito indicado em colisões bruscas, sendo aconselhável neste caso, usar uma solução analítica. É interessante ter os dois métodos à disposição num sistema de Animação por Dinâmica para que se possa avaliar qual método é mais apropriado para o tipo de choque em questão.

Com relação ao **FRED** que contém o Método Numérico de Runge-Kutta de quarta ordem para resolução de edo com m variáveis independentes, ele é mais complexo de ser programado e demora um pouco mais por unidade de tempo, mas unidades de tempo ainda maiores são requisitadas por outros métodos e os resultados não são tão precisos. O uso de cálculos adaptativos também auxilia, por exemplo, usar unidades de tempo grandes quando o corpo está caindo em queda-livre e pequenos intervalos de tempo quando o corpo colide com o chão. Uma idéia que pode ser futuramente utilizada no **ANIMADO** é usar o Método de Runge-Kutta de quarta ordem, usando cálculos adaptativos. Se o desvio for maior do que uma certa taxa, os cálculos devem ser refeitos com uma unidade de tempo um

pouco menor. Quanto à precisão numérica, o EGESP também origina resultados numéricos bastante precisos.

Muito estudo ainda está por ser feito na abordagem por Dinâmica, uma área ainda recente de pesquisa no país. O uso de formulações dinâmicas recursivas e de Métodos Numéricos sofisticados contribuem para um melhor desempenho deste tipo de sistemas de animação.

Quanto à formulação de Euler, utilizada nesta dissertação, ela compõe uma biblioteca de formulações para a Dinâmica. A experiência ganha com a criação do ANIMADO, mostrou que a escolha da formulação dinâmica a ser utilizada na animação é somente o primeiro passo na construção de um sistema de Animação por Dinâmica. A idéia é estender esta biblioteca com outras formulações. Seria interessante acrescentar a formulação recursiva de Armstrong e comparar o desempenho das duas formulações, já que o Método de Euler é um tanto caro (embora possa ser compensado pelo EGESP e pelo FRED).

Animações mais complexas (movimentos humanos, faciais e comportamentais) exigirão pesquisas mais complexas na área, não só a nível de animação, mas a nível de modelagem geométrica e visualização.

Um dos tópicos que ainda necessitam ser explorados no ambiente de Animação do ProSim é a criação de um pré-visualizador que dê uma boa idéia de como será a animação. Quanto a alguns algoritmos de visualização desenvolvidos no DCA, é preciso que sejam reavaliados, pois não foram idealizados para serem utilizados em animações, ocorrendo com frequência, problemas de falta de *aliasing* entre *frames*.

O ANIMADO tem como plataforma de trabalho estações SUN utilizando o sistema operacional Unix. Descrito em linguagem C e C++, com aproximadamente 3200 linhas de código, sua ideologia é direcionada à Programação Orientada a Objetos com utilização de *sockets* para a comunicação entre os diferentes módulos que compõem o sistema, baseado no modelo cliente/servidor.

Em resumo, a contribuição deste trabalho pode ser definida em cinco frentes:

1. Estudo e apresentação dos diversos tipos de Animação Modelada por Dinâmica,
2. Especificação de uma arquitetura flexível para o ANIMADO que comporta a comu-

nicação entre os diferentes módulos que o compõem,

3. Implementação de ferramentas numéricas adequadas para a abordagem por Dinâmica,
4. Implementação das técnicas referenciadas no decorrer da dissertação, evidenciando as vantagens e desvantagens de cada abordagem,
5. Produção de animações ilustradoras de algumas das principais técnicas de animação existentes.

Como pode ser constatado, o controle da Animação Modelada por Dinâmica tem suas vantagens e suas desvantagens. Há casos em que a modelagem por Cinemática é mais vantajosa, por isso, cabe ao animador discernir sobre a utilização de um ou outro modelo.

Segundo [TT91], os modelos de animação baseados em Dinâmica, possuem um sério problema: todos os objetos se comportam da mesma maneira. Isto é fácil de ser verificado. Os seres, em geral, e em especial os humanos, não atuam somente de acordo com leis físicas. Eles possuem **individualidade** e esta individualidade só pode ser analisada por intermédio de algum tipo de controle que estude o seu **modelo comportamental** ao lado das leis físicas que simulam o movimento humano.

O uso da Dinâmica em Computação de Imagens tem alcançado bons resultados em animação e em modelagem. Por intermédio de suas equações dinâmicas, ela não só gera movimento de corpos flexíveis ou articulados, como também, modela com um pouco mais de complexidade, restrições, colisões e controle de movimento através de pontos conhecidos. Certamente, métodos mais sofisticados e eficientes precisam ser desenvolvidos. Além das questões de baixo nível de controle da Dinâmica, é preciso considerar as questões de alto nível de controle e coordenação da animação. Para isso, outros campos de estudo devem ser explorados: Robótica, Engenharia Mecânica, Biomecânica e Teoria de Controle.

Em resumo, métodos baseados em Dinâmica e métodos baseados em Cinemática devem ser considerados como duas maneiras diferentes, mas complementares, de melhorar o controle do movimento num ambiente de Animação por Computador.

4.1 Trabalhos Futuros

Uma das principais metas para os próximos anos é o desenvolvimento de um sistema de animação integrado para a animação de cenas 3D envolvendo seres humanos conscientes da existência do ambiente em que estão. Tal sistema abrangeria uma interdisciplinaridade de abordagens e integraria aspectos e Métodos de Animação, Mecânica, Robótica, Fisiologia e Inteligência Artificial. Alguns dos objetivos deste sistema seriam:

1. Produzir, automaticamente, seres humanos sintetizados com comportamento natural,
2. Melhorar a complexidade e o realismo do movimento: o realismo do movimento necessita ser melhorado não somente sob o ponto de vista das juntas (movimentos articulados), como no caso dos robôs, mas também, na deformação dos corpos, mãos e faces durante a animação,
3. Reduzir a complexidade da descrição do movimento,
4. Ampliação dos casos de detecção de colisões, abordando o problema a nível da complexidade geométrica do objeto.

O controle do movimento na animação por computador é uma área recente e rica de pesquisa. Os métodos de controle de movimento mais utilizados, tais como *keyframing* e arquivos de descrição (*scripts*), necessitam de muito projeto para produzirem animações aceitáveis. Basta observar o tempo gasto para a produção das animações citadas no apêndice a, que além de consumirem muito tempo de trabalho, possuem uma descrição do movimento muito pouco amigável e efeitos não muito realistas. Métodos mais automáticos de animação permitirão a produção de animações sofisticadas com menos esforço do usuário. Para movimentos articulados de passeio (andar) [ZEL82], a Dinâmica certamente dará maior realismo ao movimento embora seja um difícil problema de modelagem matemática, ainda não abordado pelo ANIMADO, porém, em fase de desenvolvimento por [CAM92]. Outros problemas, tais como, pegar objetos, necessitam do uso da Cinemática Inversa, outra linha de trabalho na área de animação.

Enfim, os métodos de animação vêm buscando cada vez mais um estudo mais aprofundado em outras áreas de pesquisa, tais como, Análise Dinâmica, Planejamento de

Trajetórias (com a intenção de evitar colisões), Controle estímulo/resposta e Algoritmos de Aprendizado, tendo como objetivo final a obtenção de um maior **realismo** na animação dos objetos sintetizados por computador.

Apêndice a

Animações Produzidas

Este apêndice contém a descrição de várias seqüências de animação que foram modeladas usando arquivos de descrição de imagens, do **TOOKIMA** e do **ANIMADO**. A maioria delas está disponível em vídeo.

Estas animações foram realizadas durante o período do mestrado, e contribuíram para um importante aprendizado tanto a nível prático quanto teórico na área de Animação por Computador. Dentre as animações produzidas, em seqüência, serão apresentadas desde um exemplo de Animação *keyframe* até exemplos de Animação Modelada por Dinâmica, utilizando o **ANIMADO**.

1. “QUATRO ESTAÇÕES” - Primeiro trabalho de animação do grupo ProSlm [ST91]. [ROD92b] realizou as interpolações de cores das imagens. Essencialmente, uma técnica *keyframe* onde os quadros-chave das cores das imagens foram utilizados como pontos de controle de *Splines Lineares*, *Cúbicas* e *Hermite*, interpolando os quadros intermediários. Os autores [ST91] geraram 1440 *frames*, perfazendo um total de 1 minuto de animação (9 meses de trabalho).
2. “MANÉ MOSQUITO E CORUJITO” - Segundo trabalho de animação [ROD92a], utilizando o **TOOKIMA**. O objetivo desta animação foi testar e corrigir uma parte deste sistema de animação. Essencialmente, um exemplo de Animação Modelada por Cinemática. O aprendizado ganho com este trabalho (desde a sua idealização, até a produção do arquivo de descrição) foi de muita valia, realmente, uma experiência

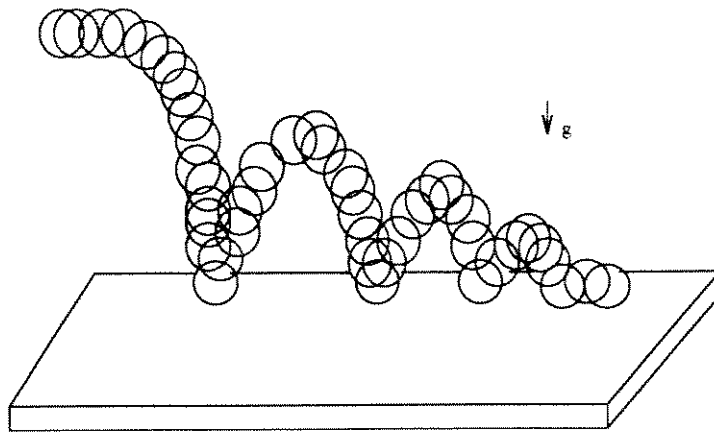


Figura a.1: Lançamento de uma esfera, colidindo geometricamente com o chão.

rica na área. Esta animação é composta por 1368 *frames*, perfazendo um total de 57 segundos de animação (6 meses de trabalho).

3. “QUEDA-LIVRE” e “LANÇAMENTO OBLÍQUO” - Terceiro trabalho de animação [ROD93], ainda utilizando o **TOOKIMA**. Aqui, o principal objetivo foi modelar situações físicas bastante simples, utilizando uma ferramenta de Animação por Cinemática. Para tal, foram escolhidas duas situações clássicas das Física (queda-livre e lançamento oblíquo) sujeitas a restrições geométricas (presença de um objeto que limita o movimento, no caso, o chão) a.1. Nestas animações são simuladas colisões (bolinha com o chão), e o chão atua com uma força de atrito na bolinha. Cada uma delas é composta por 120 *frames* (totalizando 240 *frames*), que correspondem a 8 segundos de animação (1 mês de trabalho).
4. “TEOREMA DE ATOMIZAÇÃO DO CÍRCULO” - Quarto trabalho de animação [RST93] cujo propósito foi ilustrar um teorema matemático, a pedido do IMECC¹ da UNICAMP com o intuito de ser utilizado na área educacional. Utilizando o **TOOKIMA** foram compostos 881 *frames*, totalizando 40 segundos de animação (2 meses de trabalho).
5. “MOVIMENTO PLANETÁRIO” - Quinto trabalho de animação com o objetivo de exemplificar o funcionamento do **ANIMADO** utilizando a aplicação de uma força gravitacional constante, agindo em nosso sistema solar a.2. Composta por 600 *frames*,

¹Instituto de Matemática, Estatística e Ciência da Computação

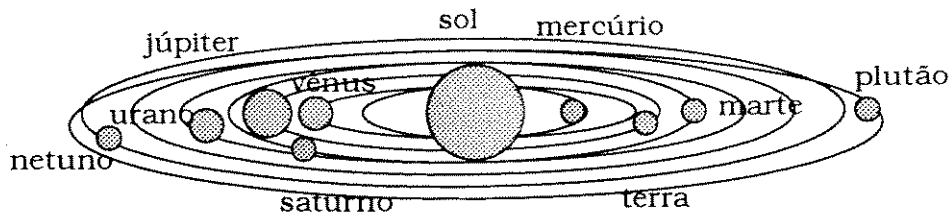


Figura a.2: Simulação do Sistema Solar.

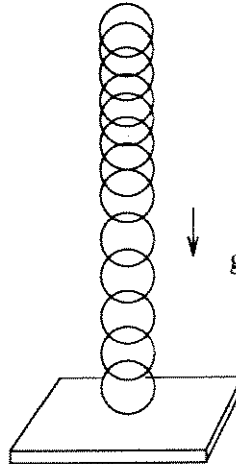


Figura a.3: Colisão de uma esfera com o chão.

totalizando 600 segundos de animação (2 dias de trabalho).

6. “QUEDA-LIVRE COM COLISÃO” - Sexto trabalho de animação, cujo objetivo foi ilustrar o ANIMA_{DO}, onde uma esfera colide com um objeto do tipo “chão” (massa infinita), dentro de uma abordagem modelada por Dinâmica, utilizando “molas” como modelo de colisão a.3. Composta por 150 *frames*, totalizando 5 segundos de animação (1 dia de trabalho).
7. “COLISÃO ENTRE UMA ESFERA E UM CUBO” - Sétimo trabalho na área. Novamente o objetivo foi testar o ANIMA_{DO} de tal forma que, os dois objetos citados (cubo e esfera) possuam movimento próprio e características físicas específicas durante o movimento. Colidirão num ponto específico e responderão à Dinâmica de

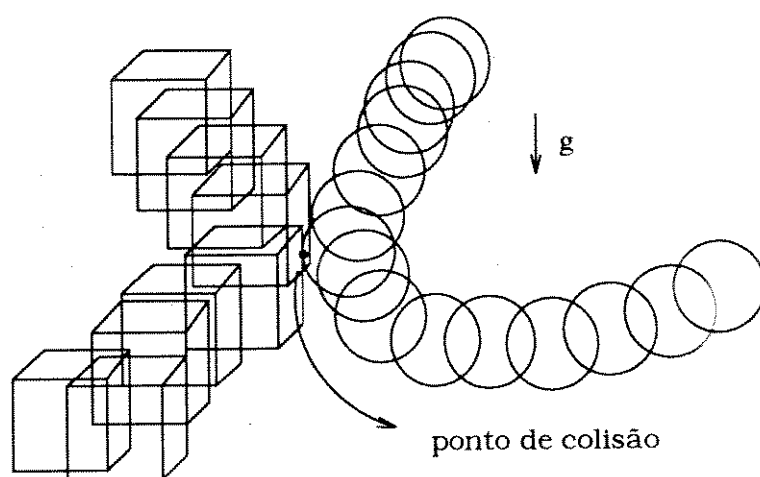


Figura a.4: Colisão de uma esfera e um cubo.

Impacto associada a esta colisão de acordo com o modelo de resolução de sistemas lineares a.4. Composta por 150 *frames*, totalizando 5 segundos de animação (1 dia de trabalho).

Apêndice b

Descrição funcional do ANIMADO

Este apêndice contém a descrição funcional do ANIMADO, que foi idealizado para plataforma Unix, usando um sistema de janelas, como por exemplo, o OpenWindows para melhor visualizar o desempenho de seus módulos.

O comando que ativa os módulos que compõem o ANIMADO chama-se **processos**, que abrirá os cinco módulos principais do ANIMADO (EGESP, FRED, COLISÃO, CONTROLE e o INTEREXPSERV), como mostra b.1.

Existe um comando de ajuda que pode ser utilizado na janela correspondente ao INTEREXPSERV. Nele estão contidas as opções de uso do comando **inter**.

Uso: **inter [-opção] [portas]**

opções :

- **-h** : mostra a tela de ajuda
- **-I** : lê dados do arquivo data.xxx
- **-i file** : lê dados de um arquivo de inicialização
- **-l** : lista todas as tabelas
- **-n** : não libera a janela para uso
- **-p** : coloca pausas na execução
- **-v** : opção para depuração

CONTROL	EGESP	INTEREXPSERV
> control 1040	> egesp 1043	> inter -i colisao -n 4 > reading data from file colisao ... > process # 414 > socket port 0: 1040 > socket port 1: 1043 > socket port 2: 1050 > socket port 3: 1070
COLISÃO		
> colisao 1070 > colisao sphere.brp cube.brp		
FRED		
> fred 1050		

Figura b.1: Comunicação entre os módulos do ANIMADO.

Caso as **portas** não forem especificadas, o **ANIMADO** criará somente uma porta.

Exemplos:

1. **inter 3 -I -n**, o **ANIMADO** lê o arquivo **data.xxx**, não libera a janela para uso, cria três portas de comunicação.
2. **inter -l -v**, lista as tabelas, entra no modo de depuração.
3. **inter -i planetas.dat**, lê o arquivo **planetas.dat**, criando uma porta de comunicação.

Depois de escolhida a opção desejada, o ambiente de execução do **ANIMADO** estará pronto para começar a execução da animação, bastando ativar cada um dos módulos, via uma porta de comunicação, dada pelo **INTEREXPSERV**. Ativa-se o **CONTROLE** através do comando **control port**, o **EGESP**, através do comando **egesp port**, o **FRED**, **fred port**, e o **COLISÃO** através do comando **colisao port**. O nome para o arquivo de saída de dados, é dado no arquivo de entrada da animação. O processo é desencadeado e, através das janelas, pode-se visualizar a comunicação entre todos os módulos.

Apêndice c

Exemplo de uso (arquivos de descrição ou script)

Este apêndice contém os arquivos criados com o objetivo de produzir as animações realizadas com o auxílio do ANIMADO, citadas no apêndice a: Movimento Planetário, Queda-livre com Colisão e Colisão entre uma Esfera e um Cubo.

```
#####  
#  
#   PLANETAS.DAT  
#  
#####
```

```
##### ATORES
```

```
INCLUDE mercurio.dat  
INCLUDE venus.dat  
INCLUDE terra.dat  
INCLUDE marte.dat  
INCLUDE jupiter.dat
```

```
INCLUDE saturno.dat
INCLUDE urano.dat
INCLUDE netuno.dat
INCLUDE plutao.dat
```

```
PI 3.14159265358979
```

```
# Unidades de medidas consideradas
```

```
# Forca: (G.e14 N)
```

```
# massa: (e24 kg)
```

```
# dsol, planeta: (e9 m)
```

```
# velocidade: (e3 m/s)
```

```
# epsilon - coeficiente de restituicao
```

```
# epsilon real - coeficiente de restituicao real
```

```
# k - constante elastica
```

```
# mi - coeficiente de atrito
```

```
# ro - distancia do ponto ao centro de massa
```

```
# d_lim - distancia limite para ativar forca de amortecimento
```

```
# pc - ponto de colisao
```

```
# i, j, k - vetor que define o plano de colisao
```

```
# F_elastica - Forca elastica
```

```
# F_amort - Forca de amortecimento
```

```
# P - impulso
```

```
# P_real - impulso real
```

```
# delta - deslocamento
```

```
# dim - dimensoes do objeto
```

```
# theta - angulo de rotacao
```

```
time_start 0.0
```

```
time_end 25
```

```
time_step 1
```

```
#####  
#  
#   mercurio  
#  
#####
```

```
##### CARACTERISTICAS
```

```
ACTOR   mercurio
```

```
FILE    mercurio.out
```

```
m      0.239  
cx      0.0  
cy      0.0  
cz      0.0  
Ixx     1.0  
Iyy     1.0  
Izz     1.0  
Ixy     0.0  
Ixz     0.0  
Iyz     0.0
```

```
##### CONDICÕES INICIAIS
```

```
R      57.8565
```

```
sx      R  
sy      0.0  
sz      0.0
```

fix 0.0
fiy 0.0
fiz 0.0

vx 0
vy 50
vz 0.0

wx 0.0
wy 0.0
wz 0.0

FORCAS/ TORQUES ATUANDO

Tx 0.0
Ty 0.0
Tz 1.0

F 1422.94
Fx - (F/R) * sx
Fy - (F/R) * sy
Fz 0

venus

#####

CARACTERISTICAS

ACTOR venus

FILE venus.out

m 4,96

cx 0.0

cy 0.0

cz 0.0

Ixx 1.0

Iyy 1.0

Izz 1.0

Ixy 0.0

Ixz 0.0

Iyz 0.0

***** CONDIC0ES INICIAIS

R 108.0885

sx R

sy 0.0

sz 0.0

fix 0.0

fiy 0.0

fiz 0.0

vx 0

vy 37

vz 0.0

wx 0.0
wy 0.0
wz 0.0

***** FORCAS/ TORQUES ATUANDO

Tx 0.0
Ty 0.0
Tz 1.0

F 8459.4
Fx - (F/R) * sx
Fy - (F/R) * sy
Fz 0

terra

***** CARACTERISTICAS

ACTOR terra

FILE terra.out

m 5.976
cx 0.0
cy 0.0

cz 0.0
Ixx 1.0
Iyy 1.0
Izz 1.0
Ixy 0.0
Ixz 0.0
Iyz 0.0

CONDICÕES INICIAIS

R 149.5

sx R
sy 0.0
sz 0.0

fix 0.0
fiy 0.0
fiz 0.0

vx 0
vy 31
vz 0.0

wx 0.0
wy 0.0
wz 0.0

FORÇAS/ TORQUES ATUANDO

Tx 0.0
Ty 0.0

Tz 1.0

F 5327.8

Fx - (F/R) * sx

Fy - (F/R) * sy

Fz 0

#####

#

marte

#

#####

CARACTERISTICAS

ACTOR marte

FILE marte.out

m 0.657

cx 0.0

cy 0.0

cz 0.0

Ixx 1.0

Iyy 1.0

Izz 1.0

Ixy 0.0

Ixz 0.0

Iyz 0.0

CONDICÖES INICIAIS

R 227.838

sx R

sy 0.0

sz 0.0

fix 0.0

fiy 0.0

fiz 0.0

vx 0

vy 26

vz 0.0

wx 0.0

wy 0.0

wz 0.0

FORÇAS/ TORQUES ATUANDO

Tx 0.0

Ty 0.0

Tz 1.0

F 252.19

Fx - (F/R) * sx

Fy - (F/R) * sy

Fz 0

```
#####  
#  
#   jupiter  
#  
#####
```

CARACTERISTICAS

ACTOR jupiter

FILE jupiter.out

m 1900.37
cx 0.0
cy 0.0
cz 0.0
Ixx 1.0
Iyy 1.0
Izz 1.0
Ixy 0.0
Ixz 0.0
Iyz 0.0

CONDICÕES INICIAIS

R 777.998

sx R
sy 0.0

sz 0.0

fix 0.0

fiy 0.0

fiz 0.0

vx 0

vy 14

vz 0.0

wx 0.0

wy 0.0

wz 0.0

***** FORCAS/ TORQUES ATUANDO

Tx 0.0

Ty 0.0

Tz 1.0

F 62560

Fx - (F/R) * sx

Fy - (F/R) * sy

Fz 0

#

saturno

#

***** CARACTERISTICAS

ACTOR saturno

FILE saturno.out

m 567.72

cx 0.0

cy 0.0

cz 0.0

Ixx 1.0

Iyy 1.0

Izz 1.0

Ixy 0.0

Ixz 0.0

Iyz 0.0

***** CONDICÕES INICIAIS

R 1420.25

sx R

sy 0.0

sz 0.0

fix 0.0

fiy 0.0

fiz 0.0

vx 0

vy 10

vz 0.0

wx 0.0

wy 0.0

wz 0.0

***** FORCAS/ TORQUES ATUANDO

Tx 0.0

Ty 0.0

Tz 1.0

F 5608.2

Fx - (F/R) * sx

Fy - (F/R) * sy

Fz 0

#

urano

#

***** CARACTERISTICAS

ACTOR urano

FILE urano.out

m 89.64

cx 0.0

cy 0.0
cz 0.0
Ixx 1.0
Iyy 1.0
Izz 1.0
Ixy 0.0
Ixz 0.0
Iyz 0.0

CONDICÕES INICIAIS

R 2870.4

sx R
sy 0.0
sz 0.0

fix 0.0
fiy 0.0
fiz 0.0

vx 0
vy 7
vz 0.0

wx 0.0
wy 0.0
wz 0.0

FORÇAS/ TORQUES ATUANDO

Tx 0.0

Ty 0.0

Tz 1.0

F 216.78

Fx - (F/R) * sx

Fy - (F/R) * sy

Fz 0

#####

#

netuno

#

#####

CARACTERISTICAS

ACTOR netuno

FILE netuno.out

m 101.59

cx 0.0

cy 0.0

cz 0.0

Ixx 1.0

Iyy 1.0

Izz 1.0

Ixy 0.0

Ixz 0.0

Iyz 0.0

CONDICÕES INICIAIS

R 4499.95

sx R
sy 0.0
sz 0.0

fix 0.0
fiy 0.0
fiz 0.0

vx 0
vy 6
vz 0.0

wx 0.0
wy 0.0
wz 0.0

FORÇAS/ TORQUES ATUANDO

Tx 0.0
Ty 0.0
Tz 1.0

F 99.966
Fx - (F/R) * sx
Fy - (F/R) * sy
Fz 0

```
#####  
#  
#   plutao  
#  
#####
```

CARACTERISTICAS

ACTOR plutao

FILE plutao.out

m 4.78
cx 0.0
cy 0.0
cz 0.0
Ixx 1.0
Iyy 1.0
Izz 1.0
Ixy 0.0
Ixz 0.0
Iyz 0.0

CONDIC0ES INICIAIS

R 5905.25

sx R
sy 0.0
sz 0.0

fix 0.0
fiy 0.0
fiz 0.0

vx 0
vy 5
vz 0.0

wx 0.0
wy 0.0
wz 0.0

***** FORCAS/ TORQUES ATUANDO

Tx 0.0
Ty 0.0
Tz 1.0

F 2.7
Fx - (F/R) * sx
Fy - (F/R) * sy
Fz 0

```
#####
```

```
#
```

```
# COLISA01.DAT
```

```
#
```

```
#####
```

```
##### ATORES
```

```
INCLUDE esferinha.dat
```

```
INCLUDE chao.dat
```

```
# unidades de medida consideradas
```

```
# Forcas: (N)
```

```
# massas: (kg)
```

```
# velocidades: (m/s)
```

```
# epsilon - coeficiente de restituicao
```

```
# epsilon real
```

```
# kelastica - constante elastica
```

```
# kamortecimento - constante amortecimento
```

```
# mi - coeficiente de atrito
```

```
# ro - distancia do pc ao centro de massa de objeto
```

```
# d_lim - distancia limite para ativar forza de amortecimento
```

```
# pc - ponto de colisao
```

```
# i, j, k - vetores da colisao
```

```
# F_elastica - Forza elastica
```

```
# F_amort - Forza de amortecimento
```

```
# P - Impulso
```

```
# P_real - Impulso real
```

```
# delta - deslocamento
# dim - dimensoes do objeto
# theta - rotacao
```

```
g          9.8
```

```
time_start 0.0
```

```
time_end   8.0
```

```
time_step  0.3
```

```
#####
```

```
#
```

```
#   esferinha
```

```
#
```

```
#####
```

```
***** CARACTERISTICAS
```

```
ACTOR   esferinha
```

```
FILE    esferinha.out
```

```
m          0.5
```

```
Ixx        4.0
```

```
Iyy        4.0
```

```
Izz        4.0
```

```
Ixy        0.0
```

```
Ixz        0.0
```

```
Iyz        0.0
```

Kelastica 600
kamortecimento 0.1
dlim 0.5

***** CONDICÕES INICIAIS

sx 0.0
sy 3.0
sz 0.0

fix 0.0
fiy 0.0
fiz 0.0

vx 0.0
vy 0.0
vz 0.0

wx 0.0
wy 0.0
wz 0.0

***** FORÇAS/ TORQUES ATUANDO

Impulsos gerados pela resolução do sistema de equações

#

chao

#

#####

CARACTERISTICAS

ACTOR chao

FILE chao.out

m	1000.0
Ixx	1.0
Iyy	1.0
Izz	1.0
Ixy	0.0
Ixz	0.0
Iyz	0.0

CONDICÕES INICIAIS

sx	0.0
sy	0.0
sz	0.0

fix	0.0
fiy	0.0
fiz	0.0

vx	0.0
vy	0.0
vz	0.0

wx 0.0

wy 0.0

wz 0.0

***** FORCAS/ TORQUES ATUANDO

Impulsos gerados pela resolucao do sistema de equacoes

```
*****
```

```
#
```

```
# COLISA02.DAT
```

```
#
```

```
*****
```

```
***** ATORES
```

```
INCLUDE esfera.dat
```

```
INCLUDE cubo.dat
```

```
# unidades de medida consideradas
```

```
# Forcas: (N)
```

```
# massas: (kg)
```

```
# velocidades: (m/s)
```

```
# epsilon - coeficiente de restituicao
```

```
# epsilon real
```

```
# kelastica - constante elastica
```

```
# kamortecimento - constante de amortecimento
```

```
# mi - coeficiente de atrito
```

```
# ro - distancia do pc ao centro de massa do objeto
```

```
# dlim - distancia limite para F_amort
```

```
# pc - ponto de colisao
```

```
# i, j, k - vetores da colisao
```

```
# F_elastica - Forca elastica
```

```
# F_amort - Forca de amortecimento
```

```
# P - Impulso
```

```
# P_real - Impulso real
```

```
# delta - deslocamento
```

```
# dim - dimensoes do objeto
# theta - rotacao
```

```
g          9.8
```

```
time_start 0.0
```

```
time_end   8.0
```

```
time_step  0.3
```

```
#####
```

```
#
```

```
# esfera
```

```
#
```

```
#####
```

```
##### CARACTERISTICAS
```

```
ACTOR esfera
```

```
FILE esfera.out
```

```
m          2.0
```

```
Ixx         4.0
```

```
Iyy         4.0
```

```
Izz         4.0
```

```
Ixy         0.0
```

```
Ixz         0.0
```

```
Iyz         0.0
```

```
Kelastica   0.5
```

CONDICÕES INICIAIS

sx 0.0
sy 3.0
sz 0.0

fix 0.0
fiy 0.0
fiz 0.0

vx 0.0
vy -0.5
vz 0.0

wx 5.0
wy 5.0
wz 5.0

FORÇAS/ TORQUES ATUANDO

Impulsos gerados pela resolução do sistema de equações

cubo

#####

***** CARACTERISTICAS

ACTOR cubo

FILE cubo.out

m	0.2
Ixx	1.0
Iyy	1.0
Izz	1.0
Ixy	0.0
Ixz	0.0
Iyz	0.0
Kelastica	1.0

***** CONDICAOES INICIAIS

sx	0.0
sy	0.0
sz	0.0

fix	0.0
fiy	0.0
fiz	0.0

vx	0.0
vy	0.0
vz	7.0

```
wx 10.0  
wy 0.0  
wz 0.0
```

```
***** FORCAS/ TORQUES ATUANDO
```

```
# Impulsos gerados pela resolucao do sistema de equacoes
```

Este último arquivo gerou alguns quadros de animação, ilustrados na figura c.1 cuja visualização deve ser feita da esquerda para a direita, de cima para baixo.

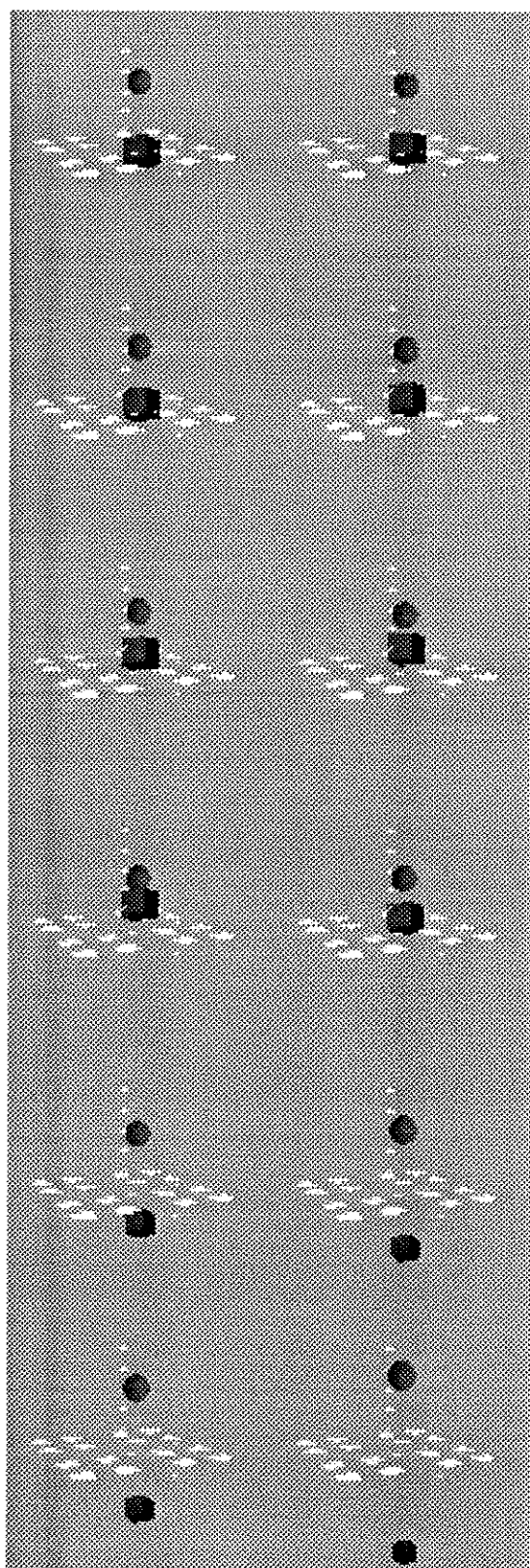


Figura c.1: Um exemplo de colisão utilizando solução analítica.

Apêndice d

Trabalhos existentes no Brasil

- [AP92]
Marcus Guilherme AMORIM e Homero Luiz Píccolo. KIKO: Sistema de Animação para representação de Movimento Humano. SIBGRAPI'92 - UnB, pp243-250, 1992.
- [ASS90]
José Dias Vasconcellos de ASSIS. Animação 3D por Computador - passo a passo. Anais da Jornada EPUSP/IEEE em Computação Visual - São Paulo, pp225-236, 1990.
- [DA92]
Clodoveu Augusto DAVIS e Arnaldo de Albuquerque ARAUJO. PIXELWARE: Um sistema de Processamento Digital de Imagens. SIBGRAPI'92 - UFMG, pp275-284, 1992.
- [FD92]
Cláudia A. S. Velloso FERREIRA e Luiz Alberto Vieira DIAS. Visualização e Simulação de um problema físico baseadas em modelagem física com o uso da Animação por Computador. SIBGRAPI'92 - INPE - SJCps, pp65-68, 1992.
- [HM92]
Marcelo da Silva HOUNSELL e Léo Pini Magalhães. TOOKIMA: Animação Cinemática. SIBGRAPI'92 - UNICAMP, pp268-274, 1992.
- [LBF92]
Robson Rodrigues LEMOS e Rui Manuel R. BASTOS e Carla M. D. S. FREITAS.

Animação de corpos rígidos e tratamento de colisões utilizando análise dinâmica. SIBGRAPI'92 - CPGCC - UFRGS, pp79-82, 1992.

– [LEM93]

Robson R. LEMOS. Tese de mestrado: Animação e Tratamento de Colisões de Corpos Rígidos utilizando Análise Dinâmica. CPGCC - UFRGS, 1993.

– [LET92]

Fabiana R. LETA. Tese de mestrado: O Método dos elementos componentes para a Animação de Figuras Articuladas. Engenharia Mecânica, PUC - Rio, 1992.

– [MSO92]

Soraia Raupp MUSSE, Ana Elisa Ferreira SCHMIDT e Sílvia Delgado OLABARRIAGA. PREVIEW: Ferramenta para Animação e Integração de Sistemas. SIBGRAPI'92 - CPGCC - UFRGS, pp251-258, 1992.

– [NED91]

Luciana Porcher NEDEL. Pinscreen: modelagem e animação por computador. Porto Alegre, PGCC - UFRGS, 1991.

– [NED93]

Luciana Porcher NEDEL. Tese de mestrado: Simulação de objetos deformáveis baseados na Análise Dinâmica. CPGCC - UFRGS, 1993.

– [OLA88]

S. D. OLABARRIAGA e A. LASCHUK. SPA - Um sistema extensível para animação modelada. Proceedings: SIBGRAPI, Petrópolis, 1988.

– [OLI91]

Manuel Menezes OLIVEIRA e Anatólio LASCHUK. Interpol3d-Sistema de Animação por Interpolação de formas de objetos descritos por Superfícies Bezier. Proceedings - SIBGRAPI'91 - CPGCC - UFRGS, pp101-113, 1991.

– [PTC92]

Homero Luiz Pícolo e Kentaro TAKAHASHI e André de Sá CARNEIRO. ACHILLES: Sistema de Visualização do Caminhar de Deficientes Físicos. SIBGRAPI'92 - UnB, pp17-20, 1992.

– [RB92]

Marcelo da Cunha RAMOS e José Antonio dos Santos BORGES. ANIMAL: Um

Sistema para Animações Tridimensionais. SIBGRAPI'92 - UFRJ - RJ, pp75-78, 1992.

– [RIG92]

S. RIGO. Animação por computador - um estudo de técnicas. Porto Alegre, PGCC - UFRGS, 1992.

– [SCH92]

A.E.F. SCHMIDT e S.R. MUSSE. PREVIEW - Sistema de Animação. CPGCC - UFRGS, 1992.

– [SIL92]

R.L. SILVA e S.D. OLABARRIAGA. ANIMAKER - Sistema de Animação Modelada. Proceedings - SIBGRAPI'92. Águas de Lindóia, pp259-267, 1992.

– [VEL89]

Luiz VELHO. SCRIPTS: A Computer Animation System. SIBGRAPI'89. GLOBO Computação Gráfica - RJ, pp509-518, 1989.

– [VEL90]

Luiz VELHO. Modelagem Procedural e Animação. Anais da Jornada EPUSP/IEEE em Computação Visual - IMPA - RJ, pp439-448, 1990.

BIBLIOGRAFIA

- [ADH⁺89] BRINO ARNALDI, Georges DUMONT, Gerard HEGRON, Daniel THALMANN, and Nadia M. THALMANN. Animation control with dynamics. In *State of the Art in Computer Animation*, pages 113–123. Springer-Verlag, 1989.
- [AG85a] W. W. ARMSTRONG and M. W. GREEN. The dynamics of articulated rigid bodies for purposes of animation. *Proc. Graphics Interface'85*, May 1985.
- [AG85b] WILLIAM W. ARMSTRONG and Mark W. GREEN. The dynamics of articulated rigid bodies for purposes of animation. *Proc. Graphics Interface 85*, pages 407–413, 1985.
- [AGL87] WILLIAM W. ARMSTRONG, Mark W. GREEN, and LAKE. Near real-time control of human figure models. *IEEE Computer Graphics and Applications*, Jan 1987.
- [B⁺87] D. E. BREEN et al. The clockworks: An object-oriented computer animation system. *Eurograph'87*, pages 275–282, 1987.
- [BAD80] N. BADLER. Special problems in human movement simulation. *SIGGRAPH'80*, 1980.
- [BAD87] NORMAN I. BADLER. Articulated figure animation. *IEEE Computer Graphics and Applications*, 7(6):10–11, Jun 1987.
- [BAD89] NORMAN I. BADLER. Artificial intelligence, natural language and simulation for human animation. In *State of the Art in Computer Animation*, pages 19–31. Springer-Verlag, 1989.
- [BAE90] SUSAN VAN BAERLE. Character animation: Combining computer graphics and traditional animation. *Proc. SIGGRAPH'90 - course notes*, (8):52–63, 1990.

- [BAR86] A. H. BARR. Dynamic constraints - State of the Art in Image Synthesis. *SIGGRAPH'86 - Course notes*, 1986.
- [BB87] Ronen BARZEL and Alan H. BARR. Modeling with dynamic constraints. *ACM SIGGRAPH'87 - course notes*, 1987.
- [BB88] Ronen BARZEL and Alan H. BARR. A modeling system based on dynamic constraints. *Proc. SIGGRAPH'88*, 22(4):179-188, Aug 1988.
- [BL86] P. BERGERON and P. LACHAPPELLE. Control of facial expressions and body movements in computer generated animated short "Tony de Pezzie". *SIGGRAPH'86*, 1986.
- [BN68] Fred BRAUER and A. John NOHEL. *Problems and Solutions in Ordinary Differential Equations*. W. A. Benjamin, 1968.
- [BOK80] N. I. BADLER, J. O'ROURKE, and B. KAUFMAN. Special problems in human movement simulation. *Computer Graphics (Proc. SIGGRAPH'80)*, pages 189-203, 1980.
- [BRA86] M. BRAUN. *Differential Equations and their Applications*. Springer-Verlag, third edition, 1986.
- [BRE84] L. BREWSTER. Interactive surgical planning. *IEEE Computer Graphics and Applications*, Mar 1984.
- [BRE89] David E. BREEN. Choreographing goal-oriented motion using cost functions. In *State of the Art in Computer Animation*, pages 141-151. Springer-Verlag, 1989.
- [BRU90] Armin BRUDERLIN. Towards realistic human figure animation. *Proc. SIGGRAPH'90*, (8):11-20, 1990.
- [BS79] Norman I. BADLER and S. SMOLIAR. Digital representation of Human Movement. *Computing Survey*, 11:19-38, Mar 1979.
- [BW89] David E. BREEN and J. WOZNY. Message-based choreography for computer animation. In *State of the Art in Computer Animation*, pages 69-82. Springer-Verlag, 1989.
- [CAL90] T. W. CALVERT. The challenge of human figure animation. *Proc. SIGGRAPH'90 - course notes*, (8):3-10, 1990.

- [CAM92] José Tarcísio Franco de CAMARGO. Técnicas de controle local de objetos rígidos articulados. Relatório de estudos especiais, DCA - FEE - UNICAMP, Jun 1992.
- [CCP82] T. W. CALVERT, J. CHAPMAN, and A. PATLA. Aspects of the kinematic simulation of human movement. *IEEE CG&A*, pages 41–52, Nov 1982.
- [CHA91] Eduardo O. C. CHAVES. *Multimídia - Conceituação Aplicações e Tecnologia*. People Computação, Campinas - SP, Set 1991.
- [CRA89] John J. CRAIG. *Introduction to Robotics, Mechanics and Control*. Addison-Wesley, 1989.
- [DIZ91] Jorge DIZ. Relatório prosim/Câmera sintética (em elaboração). Relatório Interno 031, DCA - FEE - UNICAMP, 1991.
- [DOO82] M. DOOLEY. Anthropometric modeling programs - a survey. *IEEE Computer Graphics and Applications*, Nov 1982.
- [FAR90] Gerald FARIN. *Curves and Surfaces for Computer Aided Geometric Design - A Practical Guide*. Academic Press, 1990.
- [FB74] Laurence W. FREDRICK and Robert H. BAKER. *An Introduction to Astronomy*. D. Van Nostrand Company, 1974.
- [FER93] Rubens Ramos Jr. FERNANDES. Relatório interno sobre o GEOMOD (em elaboração, DCA - FEE - UNICAMP, 1993.
- [FET82] W. FETTER. A progression of human figures simulated by computer graphics. *IEEE Computer Graphics and Applications*, Nov 1982.
- [FFC82] A. FOURNIER, D. FUSSEL, and L. CARPENTER. Computer rendering of stochastic models. *Comm. ACM*, pages 371–384, Jun 1982.
- [FFH90] Van Dan FOLEY, FEINER, and HUGHES. *Computer Graphics Principles and Practise*, chapter 21, pages 1057–1081. second edition, 1990.
- [GM85] Michael GIRARD and A. A. MACIEJEWSKI. Computacional modeling for the computer animation of legged figures. *Computer Graphics 19*, pages 263–270, Jul 1985.

- [GOL50] Werner GOLDSTEIN. *Classical Mechanics*. Addison - Wesley Publ. Company, 1950.
- [GQB89] Marc R. GROSSO, D. QUACH, and Norman I. BADLER. Anthropometry for computer animated human figures. In *State of the Art in Computer Animation*, pages 83–96. Springer-Verlag, 1989.
- [GUE89] Brian GUESTER. A system for simulating human facial expression. In *State of the Art in Computer Animation*, pages 191–202. Springer-Verlag, 1989.
- [HAB77] Richard HABERMAN. *Mathematical Models*. Prentice - Hall, 1977.
- [HAH87] James K. HAHN. Rigid body dynamics simulations - film and video show. *ACM SIGGRAPH'87*, 1987.
- [HAH88a] James K. HAHN. Introduction to issues in motion control. *SIGGRAPH'88 - course notes*, (3):101–130, 1988.
- [HAH88b] James K. HAHN. Realistic animation of rigid bodies. *Proc. SIGGRAPH'88 - course notes*, pages 113–120, 1988.
- [HAU87] Dave HAUMANN. Modeling flexible objects. *SIGGRAPH'87 Tutorial Notes: Topics in Physically Based Modeling*, Jul 1987.
- [HER82] D. Evans HERBISON. Real-time animation of human figure drawings with hidden lines omitted. *IEEE Computer Graphics and Applications*, Nov 1982.
- [HOR75] Robert. W. HORBECK. *Numerical Methods*. Quantum, 1975.
- [HOU90] Marcelo da Silva HOUNSELL. Relatório prosim/scanline. Relatório Interno 05, DCA - FEE - UNICAMP, 1990.
- [HOU92] Marcelo da Silva HOUNSELL. TOOKIMA: Uma ferramenta para animação modelada por computador. Master's thesis, DCA - FEE - UNICAMP, Abr 1992.
- [HW88] Pearce A. HILL and B. WYVILL. Animating speech: an automated approach using speech synthetised by rules. *The Visual Computer*, 3(5), 1988.
- [IC87] Paul M. ISAACS and Michael F. COHEN. Controlling dynamic simulation with kinematic constraints, behavior functions and inverse dynamics. *Proc. SIGGRAPH'87*, 21(4):215–224, Jul 1987.

- [KAW82] Y. KAWAGUCHI. A morphological study of the form of nature. *Computer Graphics (Proc. SIGGRAPH'82)*, pages 223–232, Jul 1982.
- [KOC90] Doris KOCHANÉK. Character animation by computer. *SIGGRAPH'90 - course notes*, pages 95–116, Aug 1990.
- [LAS87] John LASSETER. Principles of traditional animation applied to 3D Computer Animation. *Computer Graphics*, 21(4):35–44, Jul 1987.
- [LP87] J. P. LEWIS and F. I. PARKE. Automated lip-synch and speech synthesis for character animation. *Proc. Graphics Interface'87*, pages 143–147, 1987.
- [LT91] F. LOSERIES and Wu Shin TING. User manual for geometrical modeler PHOENIX. Technical report, 1991.
- [MAC88] Arlindo MACHADO. *A Arte do Vídeo*. Editora brasiliense, 1988.
- [MAD93] Heraldo França MADEIRA. Geomod: Modelador geométrico (em andamento). Master's thesis, DCA - FEE - UNICAMP, 1993.
- [MG79] John P. McKELVEY and Howard GROUCH. *FÍSICA I*. Harbra, 1979.
- [MN78] W. E. McLEAN and E. W. NELSON. *Engineering Mechanics: Statics and Dynamics*. McGraw-Hill, 1978.
- [MW88] Matthew MOORE and Jane WILHELMS. Collision Detection and Response for Computer Animation. *Proc. SIGGRAPH'88*, (4):289–298, 1988.
- [NE92] Farid NOURANI and Marcos J. C. EUZÉBIO. Implementação de um portmapper. Relatório interno, DCA - FEE - UNICAMP, 1992.
- [OVE90] C.W.A.M. OVERVELD. A technique for motion specification in computer animation. *The visual computer*, pages 106–116, 1990.
- [PFZ90] Michiel van PANNE, Eugene FIUME, and Vranesic ZVONKO. Reusable motion synthesis using state-space controllers. *Proc. SIGGRAPH'90*, 24(4):225–234, Aug 1990.
- [PT88] Xavier PUEYO and Daniela TOST. A survey of computer animation. *Computer e Graphics Forum*, (7):281–300, 1988.
- [RB85] W. T. REEVES and R. BLAU. Approximate and probabilistic algorithms for shading and rendering structured particles systems. *Computer Graphics (Proc. SIGGRAPH'85)*, pages 313–322, 1985.

- [REE83] W. T. REEVES. Particle systems - a technique for modeling a class of fuzzy objects. *Computer Graphics (Proc. SIGGRAPH'83)*, pages 359–376, 1983.
- [RL88] Márcia A. Gomes RUGGIERO and Vera Lúcia da Rocha LOPES. *Cálculo Numérico - Aspectos Teóricos e Computacionais*. McGraw-Hill, 1988.
- [ROD91] Maria Andréia Formico RODRIGUES. Um script de animação utilizando o TOOKIMA. In *Anais do I Simpósio de Computação de Imagens do LCA*. FEE - UNICAMP, Jul 1991.
- [ROD92a] Maria Andréia Formico RODRIGUES. Animação: Mané mosquito e Corujito. Relatório Interno 02, DCA - FEE - UNICAMP, Fev 1992.
- [ROD92b] Maria Andréia Formico RODRIGUES. Interpolação de pontos no R^n : Spline linear, Cúbica e Hermite. Relatório Interno 13, DCA - FEE - UNICAMP, Abr 1992.
- [ROD93] Maria Andréia Formico RODRIGUES. Animação: Queda-livre e lançamento oblíquo (em andamento). Relatório interno, DCA - FEE - UNICAMP, 1993.
- [ROG85] David F. ROGERS. *Procedural Elements for Computer Graphics*. McGraw-Hill Book Company, 1985.
- [ROU05] E. J. ROUTH. *Dynamics of a System of Rigid Bodies*. Macmillan and Company, 1905.
- [RR68] Paul D. RITGER and Nicholas J. ROSE. *Differential Equations with Applications*. McGraw-Hill Book Company, 1968.
- [RST93] Maria Andréia Formico RODRIGUES, Luciana Martha SILVEIRA, and Shigueo TORIGOI. Animação: Teorema de Atomização do Círculo (em andamento). Relatório interno, DCA - FEE - UNICAMP, 1993.
- [SEL89] Scott SELBIE. An introduction to the use of dynamic simulation for the animation of human movement. In *State of the Art in Computer Animation*, pages 33–45. Springer-Verlag, 1989.
- [SMI84] A. R. SMITH. Plants, fractals, and formal languages. *Computer Graphics (Proc. SIGGRAPH'84)*, pages 1–10, Jul 1984.
- [ST91] Luciana Martha SILVEIRA and Shigueo TORIGOI. Relatório Animação: Quatro Estações. Relatório Interno 29, DCA - FEE - UNICAMP, 1991.

- [STA93] Tânia Mara STAHLKE. GEOMETRIA FRACTAL: Um estudo da teoria com proposição de taxonomia baseada no processo de geração (em andamento). Master's thesis, DCA - FEE - UNICAMP, 1993.
- [SUN88] SUN. *A socket-based interprocess communications tutorial*. SUN Microsystems, 1988.
- [SYM71] Keith R. SYMON. *Mechanics*. Addison - Wesley Publishing Company, 1971.
- [TAK89] Tadao TAKAHASHI. O paradigma de objetos: Introdução e tendências, Jul 1989. Apostila do IX Congresso da SBC.
- [TJ84] THOMAS and JOHSTON. *Disney Animation The Illusion of Life*. Abberviele press, New York, 1984.
- [TPBF87] Demitri TERZPOULOUS, None PLATT, Alan BARR, and Kurt FLEISCHER. Elastically deformable models. *Computer Graphics*, 21(4), Jul 1987.
- [TT85a] Nadia M. THALMANN and Daniel THALMANN. *Computer Animation - Theory and Practise*. Springer - Verlag, Tokyo - New York, 1985.
- [TT85b] Nadia M. THALMANN and Daniel THALMANN. The state of art. *Proc. of Graphics Interface'85*, pages 209-229, 1985.
- [TT86] Nadia M. THALMANN and Daniel THALMANN. Artificial intelligence in 3d computer animation. *Computer Graphics Forum* 5, pages 341-348, 1986.
- [TT89a] Nadia M. THALMANN and Daniel THALMANN. Motion control: from keyframe to task-level animation. In *State of the Art in Computer Animation*, pages 3-17. Springer-Verlag, 1989.
- [TT89b] Nadia M. THALMANN and Daniel THALMANN. The problematics of facial animation. In *State of the Art in Computer Animation*, pages 47-55. Springer-Verlag, 1989.
- [TT91] Nadia M. THALMANN and Daniel THALMANN. Complex models for animating synthetic actors. *IEEE Computer Graphics and Applications*, pages 32-44, Sep 1991.
- [WAT87] K. WATERS. A muscle model for animating three dimensional facial expression. *Proc. SIGGRAPH'87*, 21(4):17-24, 1987.

- [WB85] Jane WILHELMS and B. BARSKY. Using dynamic analysis to animate articulated bodies such as humans and robots. *Graphics Interface 85*, pages 209–229, 1985.
- [WEI86] J. WEIL. The synthesis of cloth objects. *Computer Graphics (Proc. SIGGRAPH'86)*, pages 49–54, 1986.
- [WFB87] Andrew WITKIN, Kurt FLEISCHER, and Alan BARR. Energy constraints on parametric models. *Computer Graphics 21*, pages 225–229, Jul 1987.
- [WIL82] K. WILMERT. Visualizing human body motion simulations. *IEEE Computer Graphics and Applications*, Nov 1982.
- [WIL86] Jane WILHELMS. Virya: A motion control editor for kinematic and dynamic animation. *Proceedings of Graphics Interface 86*, pages 141–146, May 1986.
- [WIL87a] Jane WILHELMS. Toward automatic motion control. *IEEE Computer Graphics and Applications*, pages 11–22, Apr 1987.
- [WIL87b] Jane WILHELMS. Using dynamic analysis for realistic animation of articulated bodies. *IEEE Computer Graphics and Applications*, 7(6):12–27, Jun 1987.
- [WIL90] Jane WILHELMS. Dynamics for computer graphics: A tutorial. *SIGGRAPH'90 - course notes*, (8):85–115, Aug 1990.
- [WK88] Andrew WITKIN and Michael KASS. Spacetime constraints. *Proc. SIGGRAPH'88*, 22(4):159–168, Aug 1988.
- [ZEL82] David ZELTZER. Motor control techniques for figure animation. *IEEE Computer Graphics and Applications*, 2(9):53–59, Nov 1982.