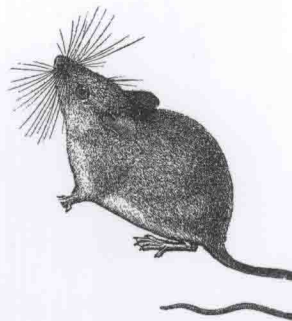


游戏开发中的物理学 (影印版)

第二版



Physics for Game Developers



O'REILLY®

东南大学出版社

David M. Bourg
& Bryan Bywalec 著

第二版

游戏开发中的物理学 (影印版)

Physics for Game Developers

David M. Bourg, Bryan Bywalec 著

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

O'Reilly Media, Inc. 授权东南大学出版社出版

南京 东南大学出版社

图书在版编目(CIP)数据

游戏开发中的物理学：第2版：英文 / (美) 布尔格 (Bourg, D.M.), (美) 拜沃利克 (Bywalec, B.) 著. —影印本. —南京：东南大学出版社，2014.1

书名原文：Physics for Game Developers, 2E
ISBN 978-7-5641-4599-6

I. ①游… II. ①布… ②拜… III. ①物理学—应用—游戏—软件开发—英文 IV. ①TP311.5

中国版本图书馆CIP数据核字(2013)第246076号

江苏省版权局著作权合同登记

图字：10-2013-378号

©2013 by O'Reilly Media, Inc.

Reprint of the English Edition, jointly published by O'Reilly Media, Inc. and Southeast University Press, 2014. Authorized reprint of the original English edition, 2013 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly Media, Inc. 出版 2013。

英文影印版由东南大学出版社出版 2014。此影印版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有，未得书面许可，本书的任何部分和全部不得以任何形式重制。

游戏开发中的物理学 第二版（影印版）

出版发行：东南大学出版社

地 址：南京四牌楼2号 邮编：210096

出 版 人：江建中

网 址：<http://www.seupress.com>

电子邮件：press@seupress.com

印 刷：扬中市印刷有限公司

开 本：787毫米×980毫米 16开本

印 张：36.25

字 数：710千字

版 次：2014年1月第1版

印 次：2014年1月第1次印刷

书 号：ISBN 978-7-5641-4599-6

定 价：89.00元（册）

本社图书若有印装质量问题，请直接与营销部联系。电话（传真）：025-83791830

Preface

Who Is This Book For?

Simply put, this book is targeted at computer game developers who do not have a strong mechanics or physics background, charged with the task of incorporating *real physics* in their games.

As a game developer, and very likely as a gamer yourself, you've seen products being advertised as "ultra-realistic," or as using "real-world physics." At the same time you, or perhaps your company's marketing department, are wondering how you can spice up your own games with such realism. Or perhaps you want to try something completely new that requires you to explore real physics. The only problem is that you threw your college physics text in the lake after final exams and haven't touched the subject since. Maybe you licensed a really cool physics engine, but you have no idea how the underlying principles work and how they will affect what you're trying to model. Or, perhaps you are charged with the task of tuning someone else's physics code but you really don't understand how it works. Well then, this book is for you.

Sure you could scour the Internet, trade journals, and magazines for information and how-to's on adding physics-based realism to your games. You could even fish out that old physics text and start from scratch. However, you're likely to find that either the material is too general to be applied directly, or too advanced requiring you to search for other sources to get up to speed on the basics. This book will pull together the information you need and will serve as the starting point for you, the game developer, in your effort to enrich your game's content with physics-based realism.

This book is not a recipe book that simply gives sample code for a miscellaneous set of problems. The Internet is full of such example programs (some very good ones we might add). Rather than give you a collection of specific solutions to specific problems, our aim is to arm you with a thorough and fundamental understanding of the relevant topics such that you can formulate your own solutions to a variety of problems. We'll do this by explaining, in detail, the principles of physics applicable to game development, and

by providing complimentary hand calculation examples in addition to sample programs.

What We Assume You Know

Although we don't assume that you are a physics expert, we do assume that you have at least a basic college level understanding of classical physics typical of non-physics and non-engineering majors. It is not essential that your physics background is fresh in your mind as the first several chapters of this book review the subjects relevant to game physics.

We also assume that you are proficient in trigonometry, vector, and matrix math, although we do include reference material in the appendices. Further, we assume that you have at least a basic college level understanding of calculus, including integration and differentiation of explicit functions. Numerical integration and differentiation is a different story, and we cover these techniques in detail in the later chapters of this book.

Mechanics

Most people that we've talked to when we was developing the concept for this book immediately thought of flight simulators when the phrases "real physics" and "real-time simulation" came up. Certainly cutting edge flight simulations are relevant in this context; however, many different types of games, and specific game elements, stand to benefit from physics-based realism.

Consider this example: You're working on the next blockbuster hunting game complete with first-person 3D, beautiful textures, and an awesome sound track to set the mood, but something is missing. That something is realism. Specifically, you want the game to "feel" more real by challenging the gamer's marksmanship, and you want to do this by adding considerations such as distance to target, wind speed and direction, and muzzle velocity, among others. Moreover, you don't want to fake these elements, but rather, you'd like to realistically model them based on the principles of physics. Gary Powell, with MathEngine Plc, put it like this "The illusion and immersive experience of the virtual world, so carefully built up with high polygon models, detailed textures and advanced lighting, is so often shattered as soon as objects start to move and interact."¹ "It's all about interactivity and immersiveness," says Dr. Steven Collins, CEO of Havok.com.² We think both these guys are right on target. Why invest so much time and

1. At the time of this book's first edition, Gary Powell worked for MathEngine Plc. Their products included Dynamics Toolkit 2 and Collision Toolkit 1, which handled single and multiple body dynamics. Currently the company operates under the name CM Labs.

2. At the time of this book's first edition, Dr. Collins was the CEO of Havok.com. Their technology handled rigid body, soft body, cloth, and fluid and particle dynamics. Intel purchased Havok in 2005.

effort making your game world look as realistic as possible, but not take the extra step to make it behave just as realistically?

Here are a few examples of specific game elements that stand to benefit, in terms of realism, from the use of real physics:

- The trajectory of rockets and missiles including the effects of fuel burn off
- The collision of objects such as billiard balls
- The effects of gravitation between large objects such as planets and battle stations
- The stability of cars racing around tight curves
- The dynamics of boats and other waterborne vehicles
- The flight path of a baseball after being struck by a bat
- The flight of a playing card being tossed into a hat

This is by no means an exhaustive list, but just a few examples to get you in the right frame of mind, so to speak. Pretty much anything in your games that bounces around, flies, rolls, slides, or isn't sitting dead still can be realistically modeled to create compelling, believable content for your games.

So how can this realism be achieved? By using physics, of course, which brings us back to the title of this section, the subject of *mechanics*. Physics is a vast field of science that covers many different, but related subjects. The subject most applicable to realistic game content is the subject of mechanics, which is really what's meant by "real physics."

By definition, mechanics is the study of bodies at rest and in motion, and of the effect of forces on them. The subject of mechanics is subdivided into *statics*, which specifically focuses on bodies at rest, and *dynamics*, which focuses on bodies in motion. One of the oldest and most studied subjects of physics, the formal origins of mechanics can be traced back more than 2000 years to Aristotle. An even earlier treatment of the subject was formalized in *Problems of Mechanics*, but the origins of this work are unknown. Although some of these early works attributed some physical phenomena to magical elements, the contributions of such great minds as Galileo, Kepler, Euler, Lagrange, d'Alembert, Newton, and Einstein, to name a few, have helped develop our understanding of this subject to such a degree that we have been able to achieve the remarkable state of technological advancement that we see today.

Because you want your game content to be alive and active, we'll primarily look at bodies in motion and will thus delve into the details of the subject of dynamics. Within the subject of dynamics there are even more specific subjects to investigate, namely, *kinematics*, which focuses on the motion of bodies without regard to the forces that act on the body, and *kinetics*, which considers both the motion of bodies and the forces that act on or otherwise affect bodies in motion. We'll take a very close look at these two subjects throughout this book.

Digital Physics

This book's first edition focused exclusively on mechanics. More than a decade after its release we've broadened our definition of game physics to include *digital physics* not in the cosmological sense but in the context of the physics associated with such devices as smart phones and their unique user interaction experience. As more platforms such as the Wii, PlayStation, X Box and smart phones come out and are expanded developers will have to keep up with and understand the new input and sensors technologies that accompany these platforms in order to keep producing fresh gaming experiences. But you shouldn't look at this as a burden, and instead look at it as an opportunity to enhance the user's interactive experience with your games.

Arrangement of This Book

Physics-based realism is not new to gaming, and in fact many games on the shelves these days advertise their physics engines. Also, many 3D modeling and animation tools have physics engines built in to help realistically animate specific types of motion. Naturally, there are magazine articles that appear every now and then that discuss various aspects of physics-based game content. In parallel, but at a different level, research in the area of real-time rigid body³ simulation has been active for many years, and the technical journals are full of papers that deal with various aspects of this subject. You'll find papers on subjects ranging from the simulation of multiple, connected rigid bodies to the simulation of cloth. However, while these are fascinating subjects and valuable resources, as we hinted earlier, many of them are of limited immediate use to the game developer as they first require a solid understanding of the subject of mechanics requiring you to learn the basics from other sources. Further, many of them focus primarily on the mathematics involved in solving the equations of motion and don't address the practical treatment of the forces acting on the body or system being simulated.

We asked John Nagle, with Animats, what is, in his opinion, the most difficult part of developing a physics-based simulation for games and his response was developing numerically stable, robust code.⁴ Gary Powell echoed this when he told me that minimizing the amount of parameter tuning to produce stable, realistic behavior was one of the most difficult challenges. We agree; speed and robustness in dealing with the mathematics of bodies in motion are crucial elements of a simulator. And on top of that, so are completeness and accuracy in representing the interacting forces that initiate and

3. A rigid body is formally defined as a body, composed of a system of particles, whose particles remain at fixed distances from each other with no relative translation or rotation among particles. Although the subject of mechanics deals with flexible bodies and even fluids such as water, we'll focus our attention on bodies that are rigid.

4. At the time of this book's first edition, John Nagle was the developer of Falling Bodies, a dynamics plug-in for Softimage|3D.

perpetuate the simulation in the first place. As you'll see later in this book, forces govern the behavior of objects in your simulation and you need to model them accurately if your objects are to behave realistically.

This prerequisite understanding of mechanics and the real world nature of forces that may act on a particular body or system have governed the organization of this book. Generally, this book is organized in four parts with each building on the material covered in previous parts:

Part I, Fundamentals

A mechanics refresher, comprising Chapters 1 through 6.

Chapter 1, Basic Concepts

This warm up chapter covers the most basic of principles that are used and referred to throughout this book. The specific topics addressed include mass and center of mass, Newton's Laws, inertia, units and measures, and vectors.

Chapter 2, Kinematics

This chapter covers such topics as linear and angular velocity, acceleration, momentum, and the general motion of particles and rigid bodies in two and three dimensions.

Chapter 3, Force

The principles of force and torque are covered in this chapter, which serves as a bridge from the subject of kinematics to that of kinetics. General categories of forces are discussed including drag forces, force fields, and pressure.

Chapter 4, Kinetics

This chapter combines elements of Chapters 2 and 3 to address the subject of kinetics and explains the difference between kinematics and kinetics. Further discussion treats the kinetics of particles and rigid bodies in two and three dimensions.

Chapter 5, Collisions

In this chapter we'll cover particle and rigid body collision response, that is, what happens after two objects run in to each other.

Chapter 6, Projectiles

This chapter will focus on the physics of simple projectiles laying the ground work for further specific modeling treatment in later chapters.

Part II, Rigid-Body Dynamics

An introduction to real time simulations, comprising Chapters 7 through 14.

Chapter 7, Real-Time Simulations

This chapter will introduce real-time simulations and detail the core of such simulations—the numerical integrator. Various methods will be presented and coverage will include stability and tuning.

Chapter 8, Particles

Before diving into rigid body simulations, this chapter will show how to implement a particle simulation, which will be extended in the next chapter to include rigid bodies.

Chapter 9, 2D Rigid-Body Simulator

This chapter will extend the particle simulator from the previous chapter showing how to implement rigid bodies, which primarily consists of adding rotation and dealing with the inertia tensor.

Chapter 10, Implementing Collision Response

Collision detection and response will be combined to implement real-time collision capabilities in the 2D simulator.

Chapter 11, Rotation in 3D Rigid-Body Simulators

This chapter will address how to handle rigid body rotation in 3D including how to deal with the inertia tensor. Then we'll show the reader how to extend the 2D simulator to 3D.

Chapter 12, 3D Rigid-Body Simulator

Multiple unconnected bodies will be incorporated in the simulator in this chapter. Introduction of multiple bodies requires resolution of multiple rigid body collisions, which can be very tricky. Issues of stability and realism will be covered.

Chapter 13, Connecting Objects

Taking things a step further, this chapter will show how to join rigid bodies forming connected bodies, which may be used to simulate human bodies, complex vehicles that may blow apart, among many other game objects. Various connector types will be considered.

Chapter 14, Physics Engines

In this chapter, specific aspects of automobile performance are addressed, including aerodynamic drag, rolling resistance, skidding distance, and roadway banking.

Part III, Physical Modeling

A look at some real world problems, comprising Chapters 15 through 19.

Chapter 15, Aircraft

This chapter focuses on the elements of flight including propulsor forces, drag, geometry, mass, and most importantly lift.

Chapter 16, Ships and Boats

The fundamental elements of floating vehicles are discussed in this chapter, including floatation, stability, volume, drag, and speed.

Chapter 17, Cars and Hovercraft

In this chapter, specific aspects of automobile performance are addressed, including aerodynamic drag, rolling resistance, skidding distance, and roadway banking. Additionally hovercraft shares some of the same characteristics of both cars and boats.

This chapter will consider those characteristics that distinguish the hovercraft as a unique vehicle. Topics covered include hovering flight, aerostatic lift, and directional control

Chapter 18, Guns and Explosions

This chapter will focus on the physics of guns including power, recoil, and projectile flight. Since we generally want things to explode when hit with a large projectile, this chapter will also address the physics of and modeling explosions.

Chapter 19, Sports

This chapter will focus on the physics of ball sports such as baseball, golf, and tennis. Coverage will go beyond projectile physics and include such topics as including pitching, bat swing, bat-ball impact, golf club swing and club ball impact, plus tennis racket swinging and racket/ball impacts.

Part IV, Digital Physics

Chapters in this part of the book will explain the physics behind accelerometers, touch screens, GPS and other gizmos showing the reader how to leverage these elements in their games, comprising Chapters 20 through 26.

Chapter 20, Touch Screens

Touch screens facilitate virtual tactile interfaces with mobile device games, such as those made for the iPhone. This chapter will explain the physics of touch screen and how the reader can leverage this interface in their games particularly with respect to virtual physical interaction with game elements through gesturing.

Chapter 21, Accelerometers

Accelerometers are now widely used in mobile devices and game controllers allowing virtual physical interaction between players and game objects. This chapter will explain how accelerometers work, what data they provide and how that data can be manipulated with respect to virtual physical interaction with game elements. Topics covered will include, but not be limited to integration of acceleration data to derive velocities and displacements and rotations.

Chapter 22, Gaming from One Place to Another

Mobile devices commonly have GPS capabilities and this chapter will explain the physics of the GPS system including relativistic effects. Further, GPS data will be explained and this chapter showing the reader how to manipulate that data for virtual interaction with game elements. For example, we'll show the reader how to differentiate GPS data to derive speed and acceleration among other manipulations.

Chapter 23, Pressure Sensors and Load Cells

Pressure sensing devices are used in games as a means of allowing players to interact with game elements, for example, the Wii balance board uses pressure sensors allowing players to interact with the Wii Fit game. This chapter will explain the physics behind such pressure sensors, what data they generate, and how to manipulate that data for game interaction.

Chapter 24, 3D Display

The new PlayStation Move and Microsoft's Kinect use optical tracking systems to detect the position of players' game controllers or gestures. This chapter will explain the physics behind optical tracking and how to leverage this technology in games.

Chapter 25, Optical Tracking

As televisions and handheld game consoles race to implement 3D displays, several different technologies are being developed. By understanding the physics of the glasses dependent stereoscopic displays, the new "glasses free" autostereoscopic displays, and looking forward to holography and volumetric displays, developers will be better positioned to leverage these effects in their games.

Chapter 26, Sound

Sound is a particularly important part of a game's immersive experience; however, to date no book on game physics addresses the physics of sound. This chapter will focus on sound physics including such topics of sound speed and the Doppler Effect. Discussions will also include why sound physics is often ignored in games, for example, when simulating explosions in outer space.

Appendix A, Vector Operations

This appendix shows you how to implement a C++ class that captures all of the vector operations that you'll need to when writing 2D or 3D simulations.

Appendix B, Matrix Operations

This appendix implements a class that captures all of the operations you need to handle 3x3 matrices.

Appendix C, Quaternion Operations

This appendix implements a class that captures all of the operations you need to handle quaternions when writing 3D rigid body simulations.

Part I, Fundamentals focuses on fundamental topics in Newtonian mechanics such as kinematics and kinetics. Kinematics deals with the motion of objects. We'll cover both linear and angular velocity and acceleration. Kinetics deals with forces and resulting motion. Part I serves as a primer for Part II, Rigid-Body Dynamics that covers rigid body dynamics. Readers already versed in classical mechanics can skip Part I, Fundamentals without loss of continuity.

Part II, Rigid-Body Dynamics focuses on rigid body dynamics and development of both single and multi-body simulations. This part covers numerical integration, real-time simulation of particles and rigid bodies, and connected rigid bodies. Generally, this part covers what most game programmers consider elements of a physics engine.

Part III, Physical Modeling focuses on physical modeling. The aim of this part is to provide valuable physical insight for the reader so they can make better judgments on what to include in their models and what they can safely leave out without sacrificing physical realism. We cannot and do not attempt to cover all the possible things you

might want to simulate. Instead we cover several typical things you may try to simulate in a game such as aircraft, boats, sports balls, among others with the purpose of giving you some insight into the physical nature of those things and some of the choices you must make when developing suitable models.

Part IV, Digital Physics covers digital physics in a broad sense. This is an exciting topic as it relates to the technologies associated with mobile platforms, such as smart phones like the iPhone, and ground breaking game systems such as the Nintendo Wii. Chapters in this part of the book will explain the physics behind accelerometers, touch screens, GPS and other gizmos showing the reader how to leverage these elements in their games. We recognize that these topics are not what most game programmers typically think about when they think of game physics; however, the technologies covered play an increasingly important role in modern mobile games and we feel it important to explain the underlying physics behind them with the hope that you'll be better able to leverage these technologies in your games.

In addition to resources pertaining to real-time simulations, the Bibliography at the end of this book will provide sources of information on mechanics, mathematics, and other specific technical subjects, such as books on aerodynamics.

Conventions Used in This Book

The following typographical conventions are used in this book:

Constant width

Used to indicate command-line computer output, code examples, Registry keys, and keyboard accelerators (see “Keyboard Accelerators” later in this book).

Constant width italic

Used to indicate variables in code examples.

Italic

Introduces new terms and to indicate URLs, variables, filenames and directories, commands, and file extensions.

Bold

Indicates vector variables.



This icon signifies a tip, suggestion, or general note.



This icon indicates a warning or caution.

We use boldface type to indicate a vector quantity, such as force, **F**. When referring to the magnitude only of a vector quantity, we use standard type. For example, the magnitude of the vector force, **F**, is *F* with components along the coordinate axes, *F_x*, *F_y*, and *F_z*. In the code samples throughout the book, we use the * (asterisk) to indicate vector dot product, or scalar product, operations depending on the context, and we use the ^ (caret) to indicate vector cross product.

Using Code Examples

This book is here to help you get your job done. In general, if this book includes code examples, you may use the code in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of examples from O'Reilly books does require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation does require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: "*Physics for Game Developers, 2nd Edition* by David M. Bourg and Bryan Bywalec (O'Reilly). Copyright 2013 David M. Bourg and Bryan Bywalec, 978-1-449-39251-2."

If you feel your use of code examples falls outside fair use or the permission given above, feel free to contact us at permissions@oreilly.com.

Safari® Books Online



Safari Books Online (www.safaribooksonline.com) is an on-demand digital library that delivers expert content in both book and video form from the world's leading authors in technology and business.

Technology professionals, software developers, web designers, and business and creative professionals use Safari Books Online as their primary resource for research, problem solving, learning, and certification training.

Safari Books Online offers a range of product mixes and pricing programs for organizations, government agencies, and individuals. Subscribers have access to thousands of books, training videos, and prepublication manuscripts in one fully searchable database from publishers like O'Reilly Media, Prentice Hall Professional, Addison-Wesley Professional, Microsoft Press, Sams, Que, Peachpit Press, Focal Press, Cisco Press, John Wiley & Sons, Syngress, Morgan Kaufmann, IBM Redbooks, Packt, Adobe Press, FT Press, Apress, Manning, New Riders, McGraw-Hill, Jones & Bartlett, Course Technol-

ogy, and dozens more. For more information about Safari Books Online, please visit us online.

How to Contact Us

Please address comments and questions concerning this book to the publisher:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938 (in the United States or Canada)
707-829-0515 (international or local)
707-829-0104 (fax)

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at <http://oreil.ly/Physics-GameDev2>.

To comment or ask technical questions about this book, send email to bookquestions@oreilly.com.

For more information about our books, courses, conferences, and news, see our website at <http://www.oreilly.com>.

Find us on Facebook: <http://facebook.com/oreilly>

Follow us on Twitter: <http://twitter.com/oreillymedia>

Watch us on YouTube: <http://www.youtube.com/oreillymedia>

Acknowledgments

We want to thank Andy Oram, the editor of this edition of the book, for his skillful review of our writing and his insightful comments and suggestions, not to mention his patience. We also want to express my appreciation to O'Reilly for agreeing to take on this project giving us the opportunity to expand on the original edition. Furthermore, special thanks go to all of the production and technical staff at O'Reilly.

We'd also like to thank the technical reviewers, Christian Stober and Paul Zirkle, whose valuable insight added much to this edition.

Individually, David would like to thank his loving wife and best friend, Helena, for her endless support and encouragement, and his wonderful daughter, Natalia, for making every day special.

Bryan would like to thank his co-author David for the opportunity to help with the second edition and would also like to thank his parents, Barry and Sharon, for raising him to be curious about the world. Lastly, he would like to thank his fiancée, Anne Hasuly, for her support without which many chapters would still be half-finished.

Table of Contents

Preface.....	xi
--------------	----

Part I. Fundamentals

1. Basic Concepts.....	3
Newton's Laws of Motion	3
Units and Measures	4
Coordinate System	6
Vectors	7
Derivatives and Integrals	8
Mass, Center of Mass, and Moment of Inertia	9
Newton's Second Law of Motion	20
Inertia Tensor	24
Relativistic Time	29
2. Kinematics.....	35
Velocity and Acceleration	36
Constant Acceleration	39
Nonconstant Acceleration	41
2D Particle Kinematics	42
3D Particle Kinematics	45
X Components	46
Y Components	47
Z Components	48
The Vectors	48
Hitting the Target	49
Kinematic Particle Explosion	54
Rigid-Body Kinematics	61
Local Coordinate Axes	62

Angular Velocity and Acceleration	62
3. Force.....	71
Forces	71
Force Fields	72
Friction	73
Fluid Dynamic Drag	75
Pressure	76
Buoyancy	77
Springs and Dampers	79
Force and Torque	80
Summary	83
4. Kinetics.....	85
Particle Kinetics in 2D	87
Particle Kinetics in 3D	91
X Components	94
Y Components	95
Z Components	95
Cannon Revised	95
Rigid-Body Kinetics	99
5. Collisions.....	103
Impulse-Momentum Principle	104
Impact	105
Linear and Angular Impulse	112
Friction	115
6. Projectiles.....	119
Simple Trajectories	120
Drag	124
Magnus Effect	132
Variable Mass	138

Part II. Rigid-Body Dynamics

7. Real-Time Simulations.....	143
Integrating the Equations of Motion	144
Euler's Method	146
Better Methods	153