INTRODUCTION TO COMPUTATIONAL PHYSICS

Marvin L. De Jong
College of the Ozarks

Addison-Wesley Publishing Company
Reading, Massachusetts • Menlo Park, California • New York
Don Mills, Ontario • Wokingham, England • Amsterdam • Bonn
Sydney • Singapore • Tokyo • Madrid • San Juan • Milan • Paris

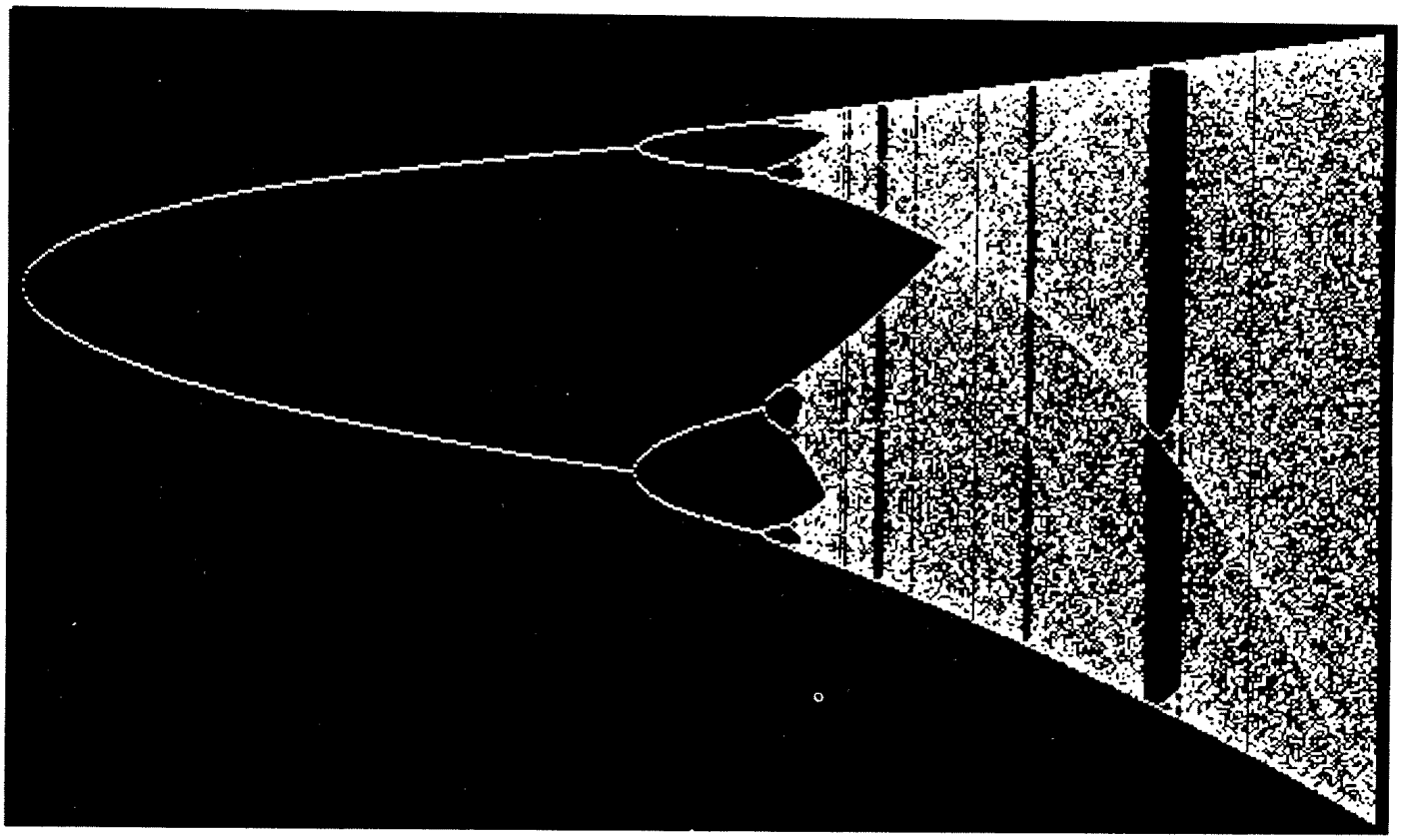
QC52 .D4 1991

Library of Congress Cataloging-in-Publication Data
De Jong, Marvin L.
Computational physics / Marvin De Jong.
p. cm
ISBN 0-201-51387-0
1. Physics—Data processing. 2. Mathematical physics—Data processing. I. Title.
QC52.D4 1991
530'.0285—dc20 90-40467
CIP

Copyright © 1991 by Addison-Wesley Publishing Company, Inc.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher. Printed in the United States of America.

1 2 3 4 5 6 7 8 9 10 MA 9594939291



Logistic Map

Preface

A few years ago I purchased James Gleick's *Chaos* (Viking, New York), and quickly became absorbed in his book, which must be one of the best popularizations of science ever written. In particular, I knew that I had to construct and see for myself a solution to the Lorenz equations, and I wanted to write a program to make one of Michael Barnsley's ferns[1]. The solution to the Lorenz equations came relatively quickly. The illustration on the opening page of the Supplement is one example. The fern was more difficult, and there was something deeply mysterious about using random numbers and a few simple transformations to create the fern. I have since made the fern, but it is not clear whether a new secret of nature has been discovered, or if we simply have a mathematical procedure that has no relationship to real ferns. It was a quite exciting time in my life, as I discovered dynamical chaos for myself, along with fractals, Julia sets, and the Mandelbrot set. I must admit to being captivated. It is not by chance that the chapter on chaotic dynamic systems is the longest one in this book.

At approximately this same time, interest in computational physics started to mushroom. There were conferences of national significance on computers in physics instruction and computational physics; a national committee began a study of the calculus-based introductory physics course, and soon called for more twentieth century physics and a new role for the computer; the *MUPPET MANIFESTO: The Implications of the Microcomputer for the Physics Major's Curriculum* outlined the case for computers in the curriculum; new and powerful tools including spreadsheets and systems capable of doing symbolic mathematics appeared on the market; a new journal, *Computers in Physics*, was begun; and computational power continued to decline in price. Now the beginning calculus sequence is also being reformed by our colleagues in mathematics. Among other things, the reformed version of beginning calculus will be more numerical and it will make use of the new computational tools that are now available.

The time seemed ripe to make a significant effort to introduce computational techniques and skills to calculus-based introductory physics students. This book is my contribution to the movement to change both how physics is taught and the content of the introductory course. In this book the computer is used to solve problems, build intuition, and understand physics. The goal of this book is to put computational tools for doing physics in the hands of students early in their college experience. I believe that "tool" is the appropriate metaphor for the computer. I also believe these skills are just as important to students in biology, chemistry, medicine, and engineering as they are to physics majors.

To Instructors

Once the decision was made to write this book, the question of the appropriate environment arose. In what hardware and software setting should the computational work be done? There are many choices of environments: programming languages such as Pascal, FORTRAN, or BASIC; spreadsheets such as Excel™, Jazz™, or Lotus 1-2-3™; packages to do numerical mathematics such as MathCad™, MATLAB™, and TK Solver™, and symbolic mathematical systems such as Mathematica™, Derive™, and Maple™. It seemed less than wise to pick one of these to the exclusion of all of the others, and I decided to express the work of this book in English-language algorithms. And, in fact, any of the preceding environmental options are suitable for solving the problems in this book. It is as simple to implement Euler's method in Maple as it is in Pascal.

It will be up to the instructor to choose the environment appropriate to each setting. It will be up to the student to translate the algorithms of this book into the language of his or her setting. The last statement is, I think, quite important. If the computer is to be a tool rather than a teacher or a tutor, then the student must acquire skills in using the tool. Providing slick animations, simulations, or demonstrations is not the goal of this book. In a Pascal environment the student should write programs that implement the algorithms in this book; if the environment is Excel, then the student should construct spreadsheet solutions to the problems, at least up to the point where students have acquired some fluency in the environment.

I think it is appropriate to make a few comments on the choice of environment. It is possible that my bias as a programmer is showing, but I feel that the best environment is a programming language, Pascal, FORTRAN, BASIC, etc. The choice between a programming language or one of the other environments is a choice between static or dynamic graphics. A program gives the student a dynamic perspective of the solution to a problem that neither the numerical nor the symbolic mathematics environments provide. Their output is static. For example, when projectile motion is modelled with a program, students can watch the "flight" as time marches onward. Other environments will only produce a graph of the solution. It is difficult to see, for another example, how intuition about chaotic dynamics can be built as easily in a static context as in a dynamic one. However, I have used spreadsheets to do many of the problems, and spreadsheets have the pedogical advantage of greater simplicity.

Another consideration about the choice of environment is cost. The least expensive choice is probably a programming language, both in terms of the package and in terms of memory requirements. Turbo Pascal™, for example, is much cheaper than Excel. Mathematica not only is expensive in dollars, it requires approximately four megabytes of memory. Most programming languages will require less than one-tenth of that amount of memory. BASIC of one sort or another is frequently bundled with the machine.

Students and instructors will bring a large variety of environmental skills to this book, from almost total illiteracy to proficiency in several environments. To meet the needs of those students and instructors with little experience, I have provided at the end of each chapter bare bones BASIC programs that implement most of the algorithms. These programs may also serve as models or pseudocode for other environments. I have chosen BASIC because it is the closest thing to a universal language in the world of computers, and it is simple to understand because it was written for beginners. The dialect of BASIC that I have chosen is True BASIC, a version of BASIC that is more portable than other versions, because of its graphics capability, and because it is structured. (Be aware that simply loading and running the programs in this book will not achieve the goals previously outlined.) With the exception of some drawings, all of the figures in this book were made with True BASIC and a "paint" program, which did the inversions and attached the legends. If your language has the equivalent of a

```
PLOT X, Y
```

instruction, then all of the graphics in this book are possible.

If an instructor incorporates computational physics into a course, what must be left out? The current curriculum in calculus-based introductory physics is packed. I have near me a new first edition of a physics textbook with more than 1000 pages, an accompanying study guide 755 pages thick with 1530 practice exercises, 208 example problems, and 901 practice test questions. Students have math anxiety, professors have textbook anxiety. It is now clearly impossible to cover all of the material, and decisions must be made about course content. Conscientious people that we are, we do not want to "cheat" or handicap our students. Furthermore, innovation feels risky for those of us who have the same relationship to Halliday-and-Resnick as an evangalist has to the Bible. But our students are less fragile than we think, and the company is impressive in the "less-is-more" camp. So let's use the bite-the-

bullet metaphor, and take the responsibility and freedom, as professionals, to decide for ourselves which topics to include and which to omit. At this point there appears to be no such thing as conventional wisdom. New models for teaching calculus-based introductory physics are a much-sought-after commodity. In my own reflections I wonder if content is less important than the ability to think conceptually and bring mathematical skills—analytical and numerical—to bear on physics problems.

If you choose to teach numerical problem-solving skills to your students, how should you use this book? Above all, this book is intended to be a self-paced and hands-on experience with the book and the computer side-by-side. The book is supplemental, not comprehensive. There is, furthermore, not even a hint that you should cover all of it. Realistically, you should also expect some cost in time while the students learn or adapt to the environment you have chosen. Chapters 1 and 2 are probably essential, especially if you plan to do Chapter 12 on chaotic dynamics, which I urge you to try. (In fact, Chapter 12 can be profitably placed after Chapter 2.) To help relate the book to the current curriculum, you might call the first five chapters "mechanics," the next five "E & M," and the last three "modern physics."

You might try a laboratory approach, sometimes doing experiments, other times working with the computer. Assign one or more sections, and have students hand in their programs, the output of those programs and a written response to the questions. The questions should be answered in sentences or paragraphs as opposed to one-word answers. Look for students who really like working with computers and let them use this book in independent study. Or, try replacing one lecture or recitation a week with time spent by students working on the computer doing computational physics. We all acquire guilt feelings when we do not lecture, but I think that deep down we know that students learn when they are *engaged*, and some of our lectures are less than engaging. I hope that with the new course content that will come from current studies of the introductory physics curriculum, and with results of current research in physics education, a new paradigm in learning physics will emerge. My bet is that the computer-as-tool metaphor will be an important component.

I want to close this section with a few comments about the content of this book. An important theme, implicit in the calculus-based introductory course and implicit in this book, is the solution of differential equations. There are good reasons for this. Newton's Second Law is a differential equation, which, in our introductory physics courses, we cleverly disguise in the form **F** = m**a**. Kirchhoff's Loop Rule is frequently a differential equation. So are wave equations, and so is Schrodinger's equation. Heat flow problems also yield differential equations. The differential equation is ubiquitous. However, we do not make a major point of this in the text. Because the problems are solved numerically, there is no need to overwhelm students with the fact that we are solving second-order differential equations. For example, to solve motion problems, the following three simple relationships are required:

$$v_{n+1} \cong v_n + ha_n,$$

where

$$a_n = \frac{F_n}{m},$$

and

$$x_{n+1} \cong x_n + hv_n.$$

On the other hand, I feel that it is extremely important for students to understand, mathematically and/or intuitively, where these equations originate. Newton's Second Law should be obvious. The other two are obtained from the definition of the derivative, which they find in their calculus textbook. I use acceleration, in a one-dimensional context, to illustrate just what I mean. Let the $\equiv$ symbol mean "is defined as." Then, in one dimension,

$$a(t) \equiv \frac{dv(t)}{dt} \equiv \lim_{h \to 0} \frac{v(t+h) - v(t)}{h}.$$

For a sufficiently-small time interval h, this turns into the approximation ($\cong$),

$$a(t) \cong \frac{v(t+h) - v(t)}{h},$$

which may be rearranged to read

$$v(t+h) \cong v(t) + ha(t).$$

If this approximation is expressed in subscript notation, we obtain

$$v_{n+1} \cong v_n + ha_n.$$

A similar derivative argument relates position and velocity. Thus

$$x_{n+1} \cong x_n + hv_n.$$

It is difficult to overemphasize how important the concepts just described are. They are *key concepts* for most of the book, and they empower us to solve quite complex differential equations. For many students the approximations will also be intuitive; the velocity at time $t + h$ is the velocity at time t plus the acceleration times the time.

I have chosen to use the "$v(t + h)$" notation and the difference quotient,

$$\frac{v(t+h) - v(t)}{h},$$

precisely because that is how students will see the derivative defined in their calculus textbook. Specifically they will see the derivative defined this way:

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h},$$

or

$$\frac{df(x)}{dx} = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}.$$

The notation I have used is a modern one; the Δ notation,

$$\frac{dy}{dx} = \lim_{\Delta x \to 0} \frac{\Delta y}{\Delta x},$$

used so frequently in physics textbooks, is seldom found in calculus. Furthermore, I think it is easy to understand that, for example, $v(t + h)$ is the velocity at time $t + h$, while $v(t)$ is the velocity at time t, hence $[v(t + h) - v(t)]$ is the *change* in the velocity in time h. This makes $[v(t + h) - v(t)]/h$ the average acceleration.

In concluding this section, I wish you good luck. We are breaking new ground—teachers, publishers, authors, and students, all searching and groping for the best ways to use the computer in physics. This book makes available to the student an option that is economically feasible, within reach intellectually, and, I sincerely hope, useful, joyful, creative, and engaging.

To the Student

Your section is shorter than that for Instructors. If you have read the preceding paragraphs you will already know that the goal of this book is to help you learn how to use the computer to do physics. I have no hints or tricks that will make this easy. Learning is not easy; it is frequently frustrating.

A year or so ago, a student in my calculus-based introductory physics class came to me at the end of the semester and said, "I learned how to do physics problems, but I don't feel that I understand physics." He was a bright student, and he did know how to do physics problems. In fact, he was very good at it. Like so many of his predecessors, including myself, have done, he had committed to memory a repertoire of physics problems and the techniques to solve them. He could proceed *by analogy* on his homework and tests. Indeed, the homework helped him create his repertoire.

I take some responsibility for his problem, perhaps most of it. I should have asked many more conceptual questions, probing the class members for their understanding of physics. On exams, I should not have provided such large rewards for doing the same kind of problems that the students had done for their homework. In the classroom and on exams, the students should have been required to think.

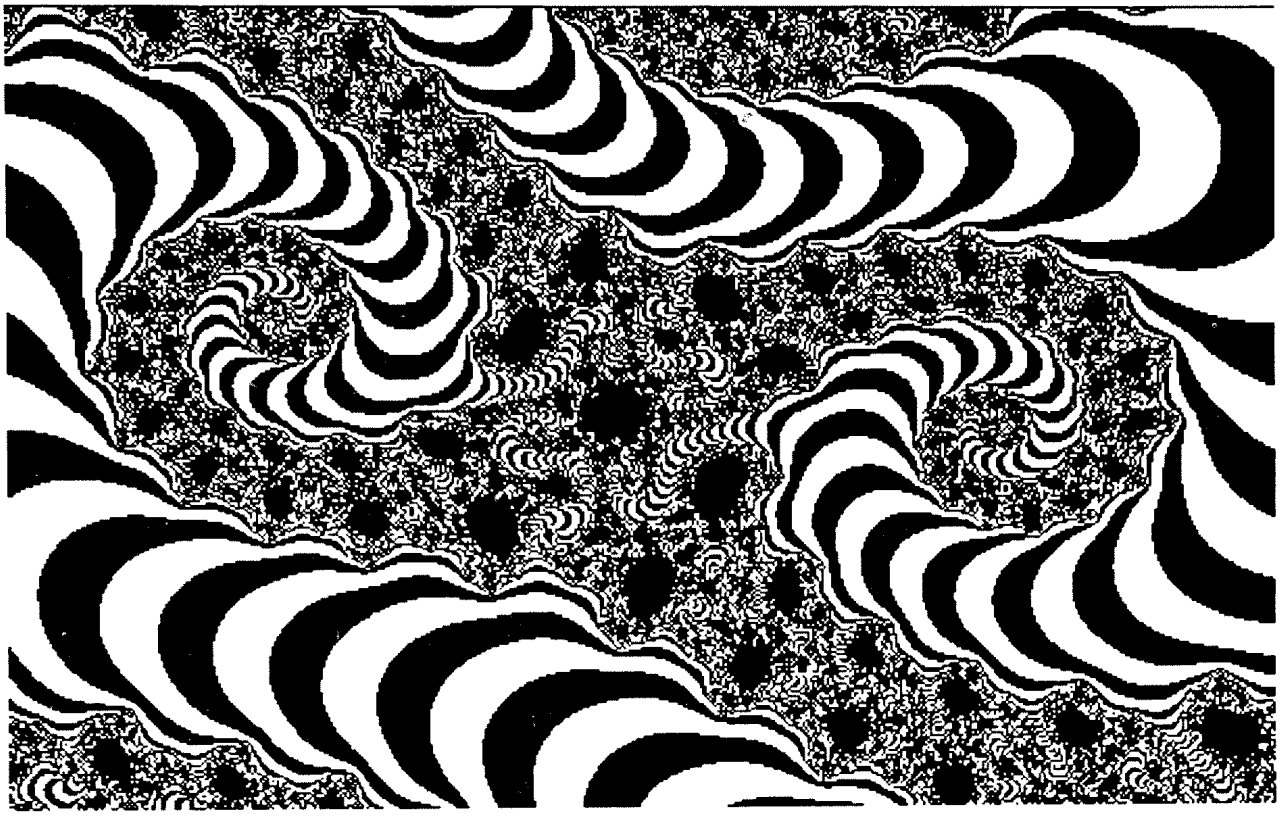
He also bears some responsibility for what happened (or didn't happen). He had a chance to think about the problems, to reflect on their answers, and to conceptualize the physics that was involved. Instead he chose to get finished with his homework and do something else. For the most part, no one can make you think physics. It's a choice you make.

Their are also choices to be made with respect to the work in this book. You can write/run many of the programs more or less automatically, and that, like problem solving, is a skill worth acquiring. But if you want to understand physics, you must make a conscious attempt to give written conceptual explanations and to understand for yourself the output of the programs. For example, give a conceptual explanation of why air resistance makes the time required for a baseball to fall from a height h longer than the time required to get there in the

first place? I have tried to ask many questions like that, but only you can give thoughtful answers.

Understanding physics is a shared responsibility and a shared joy. What a professor or a textbook may do is engage you. I hope that I have assumed my share of the responsibility by engaging you and giving you an opportunity to think physics. The rest of the responsibility is yours.

[1] M. Barnsley, *Fractals Everywhere* (Boston: Academic Press, 1988), 103.



Fractal Geometry

Contents

Chapter 1 The Motion of a Falling Object 1

1.1 Introduction 1
1.2 A Complication 2
1.3 A New Look at a Falling Object 3
1.4 Some Important Programming Considerations 9
1.5 About = and *h* 11
1.6 Graphing the Velocity and Acceleration of a Falling Object 13
1.7 Some "What If?" Questions 15
1.8 Finding *s(t)* for a Falling Object 16
1.9 Summary 18
1.10 Additional Problems 19
1.11 References 20
1.12 Programs 20

Chapter 2 One-Dimensional Dynamics: An Object on a Spring 23

2.1 Introduction 23
2.2 What's the Problem? 25
2.3 An Object on the End of a Spring 26
  2.3.1 The Physics 26
  2.3.2 The Mathematics 28
  2.3.3 A Computer Programming Note 29
  2.3.4 The Solution 30
2.4 About Analytic and Numerical Solutions (Optional) 32
2.5 Complicating the Problem with Damping 34
2.6 Phase Space 35
2.7 An Improved Algorithm 37
2.8 Additional Problems 38
2.9 References 40
2.10 Programs 40

Chapter 3 Computer Model of an Accelerating Car 43

3.1 Introduction 43
3.2 A Matter of Horsepower 45
3.3 Toward a More Realistic Model of an Accelerating Car 46
  3.3.1 Motion with Limited Power 46
  3.3.2 The Accelerating Car with Air Resistance and Rolling Friction (Challenging) 49
  3.3.3 A Horsepower Study 51
  3.3.4 A Drag Coefficient Study 52
  3.3.5 Shifting Gears 53
  3.3.6 One Last Refinement 55
3.4 Additional Problems 59
3.5 References 60
3.6 Programs 60

Chapter 4 Dynamics in Two Dimensions 65

4.1 Introduction 65
4.2 Projectile Motion 68
  4.2.1 Projectile Motion with No Air Resistance 68
  4.2.2 Projectile Motion with Air Resistance 69
  4.2.3 Families of Projectiles 72
  4.2.4 Projectile Motion with Variable Gravitational Force 74

4.3 From Projectiles to Satellites 77
4.4 More About Orbits 80
4.5 The Orbit of the Earth Around the Sun and the Orbit of an Electron Around a Proton 81
4.6 Additional Problems 84
4.7 References 85
4.8 Programs 85

Chapter 5 Energy 91

5.1 Introduction 91
5.2 Energy of an Object on a Spring 92
5.2.1 Calculating and Graphing K, U, and E 92
5.2.2 Calculating the Internal Thermal Energy of a Dissipative System (Very Challenging) 96
5.3 Energy in Projectile Motion 101
5.4 Energy in Satellite Motion 101
5.4.1 Gravitational Potential Energy: Background 101
5.4.2 Potential Energy in the Vicinity of a Single Mass 102
5.4.3 Total Energy in Orbital Motion 103
5.5 Angular Momentum in Orbital Motion 105
5.6 Additional Problems 105
5.7 Programs 106

Chapter 6 Electric Fields and Potential 109

6.1 Introduction 109
6.2 Two-Dimensional Representations of the Electric Field 112
6.3 Graphing Equipotential Curves 116
6.3.1 What Are Equipotential Curves? 116
6.3.2 A Simple Scheme for Visualizing Equipotential Curves 117
6.3.3 A More Sophisticated Program for Drawing Equipotential Curves 118
6.4 Programs 120

Chapter 7 *RC*, *LR*, and *LRC* Circuits 123

7.1 Introduction 123
7.2 The *RC* Circuit 124
7.2.1 Charging and Discharging a Capacitor 124
7.2.2 Energy in the *RC* Circuit (Challenging) 128

7.3 The *LR* Circuit 131
7.3.1 Current in an *LR* Circuit 131
7.3.2 Power in an *LR* Circuit 134
7.3.3 Energy Relationships in an *LR* Circuit (Challenging) 135
7.4 The *LRC* Circuit 137
7.5 Additional Problems 142
7.6 Programs 142

Chapter 8 Driven *LR*, *RC*, and *LRC* Circuits 145

8.1 Introduction 145
8.2 The *LR* Circuit 146
8.2.1 Computer Model of an *LR* Circuit Driven by a Sine Function 146
8.2.2 An *LR* Circuit Driven by a Square Function 149
8.3 Computer Model of an *LRC* Circuit 150
8.3.1 An *LRC* Circuit Driven by a Sine Function 150
8.3.2 An *LRC* Circuit Driven by a Square Function 153
8.3.3 Phase Space for *LRC* Circuits 154
8.4 Additional Problems 155
8.5 Programs 159

Chapter 9 Rutherford Scattering and Other Collisions 163

9.1 Introduction 163
9.2 The Physics of the Rutherford Scattering Experiment 164
9.3 Doing a Rutherford Scattering Experiment 170
9.4 Additional Problems 172
9.5 References 172
9.6 Program 173

Chapter 10 The H_2^+ ION 177

10.1 Introduction 177
10.2 The H_2 Ion 178
10.3 Stability of the H_2 Ion 182
10.3.1 Energy 182
10.3.2 The Covalent Bond 184
10.3.3 Quantum Mechanics and the H_2 Ion 185
10.4 Additional Problems 186
10.5 References 187
10.6 Programs 188

Chapter 11 Using Random Numbers to Model Radioactivity 193

11.1 Introduction 193
11.2 Modelling Radioactive Decay 196
11.3 A Mathematical Model of Radioactive Decay 198
11.4 Radioactive Series 199
11.5 Programs 201

Chapter 12 Chaotic Dynamics 205

12.1 Introduction 205
12.2 Some Definitions 206
12.3 The Simple Pendulum 208
12.4 Runge-Kutta Solutions to the Equation of Motion 210
12.5 Potential Energy of a Dynamical System 212
12.6 Portraits in Phase Space 214
12.6.1 Undamped Motion 214
12.6.2 Damped Motion 215
12.6.3 Driven and Damped Oscillators: The Road to Chaos 217
12.7 Poincaré Sections 222
12.8 The Fractal Nature of Chaotic Attractors 225
12.9 Additional Problems 227
12.10 References 229
12.11 Programs 230

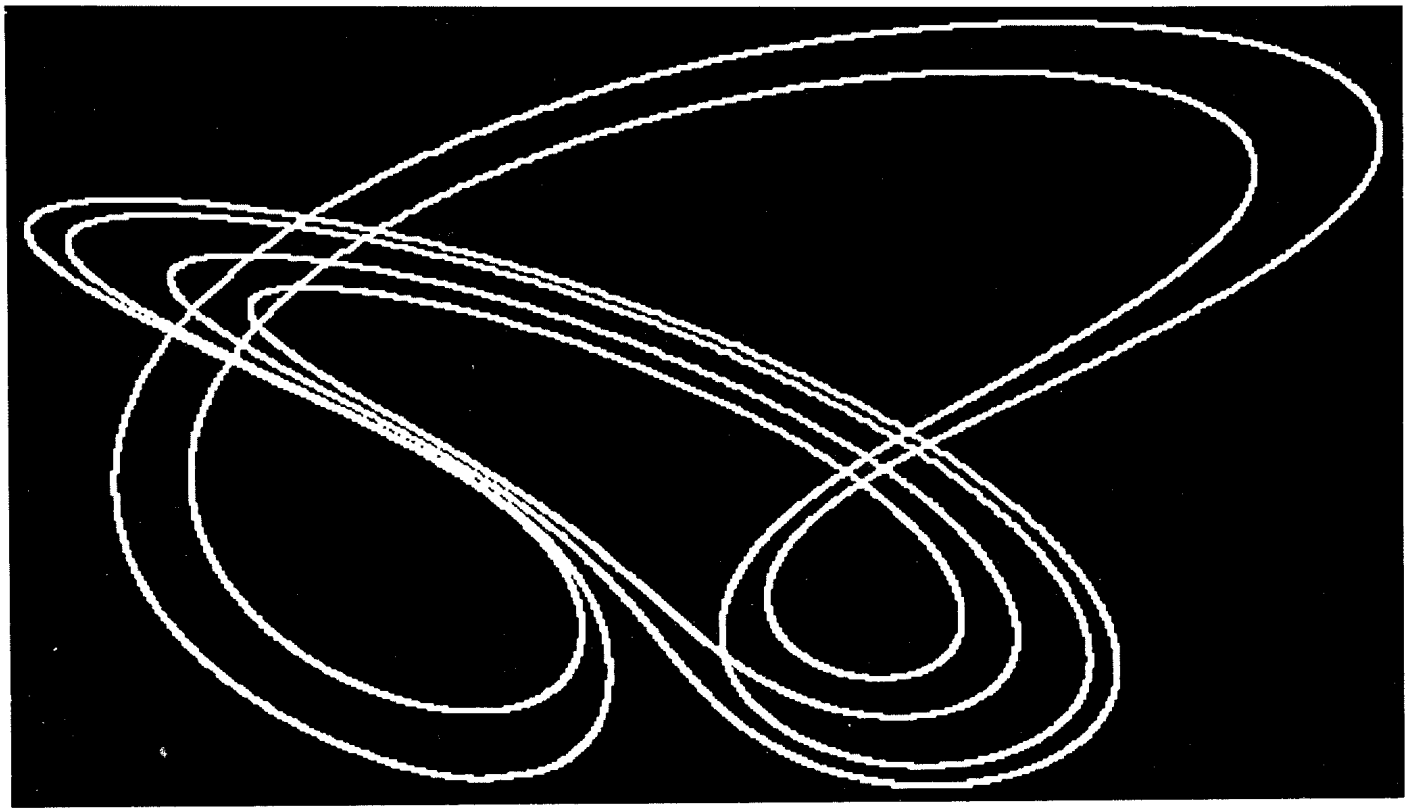
Chapter 13 Doing Physics with Random Numbers 235

13.1 Introduction 235
13.2 Using Random Numbers to Evaluate Integrals 237
13.3 Finding Masses and Moments of Inertia 240
13.3.1 Masses 240
13.3.2 Moments of Inertia 242
13.4 Noise 243
13.4.1 Calculating the Mean and Standard Deviation of a Noise Distribution 243
13.4.2 The Form of a Gaussian Noise Distribution 246
13.4.3 Extracting a Signal from Noise 249
13.5 Additional Problems 253
13.6 References 253
13.7 Programs 253

Supplement Graphics 259

S.1 Introduction 259
S.2 Coordinate-System Graphics 260
S.2.1 Coordinate-System Graphics in True BASIC 260
S.2.2 Coordinate-System Graphics in QuickBASIC 262
S.3 Pixel Graphics 264
S.4 Curves Graphed Parametrically 268
S.5 Aspect Ratio 269
S.6 Answers to the Programming Problems in this Supplement 272
S.7 Library ComPhys 274

Index 279



Numerical Solution

1 / The Motion of Falling Objects

1.1 Introduction

A review of the following topics may be helpful in this chapter:

- Displacement, velocity, and acceleration
- Motion with constant velocity
- Uniformly accelerated motion
- Motion of an object in free fall ($a = g$)

We will also briefly refer to Newton's Second Law of Motion. It would be most useful to review in your calculus text difference quotients and the definition of the derivative. The purpose of this chapter is to illustrate, as simply and quickly as possible, how the computer may be used to provide realistic solutions to real problems. In the next chapter we will analyze kinematics problems in more detail and with more rigor. For now, we wish to keep the

problem and the solution simple. To this end, we will also restrict ourselves to motion in one dimension.

We will consider the following problem: How do the velocity and acceleration of an object change as it falls? If an object of mass m is dropped, its downward acceleration in the *absence* of air resistance is

$$a = g,$$

where $g = 9.8\ \mathrm{ms}^{-2}$. Although you may not yet have discussed this in your physics class, the object falls this way because the only force acting on the object is its weight, mg. Applying Newton's Second Law, $F = ma$, gives

$$mg = ma,$$

in which the m divides, so $a = g$. If a is constant, calculus can be used to prove that the velocity at any time t is given by the simple formula

$$v(t) = v_0 + gt,$$

where we assume that positive is downward and the object starts with velocity v_0. This formula may be used to find v at any time t after the object is released (and before it hits the ground); merely substitute $9.8\ \mathrm{ms}^{-2}$ for g and a time for t. All physics problems such as this one in which the acceleration is constant may be solved using the tools of conventional mathematical analysis: algebra, trigonometry, and calculus.

1.2 A Complication

There is a difficulty, however, and almost everyone will know what it is. The formula for $v(t)$ is good only if there is no air resistance. Experiments show that an object moving in air experiences a force that depends on the velocity of the object, and, therefore, the acceleration is not constant. For the range of speeds experienced by a falling object, the force due to the air may be expressed as

$$F_a = \frac{1}{2} C\rho A v^2,$$

where C is a unit-less factor called the drag coefficient, A is the cross-sectional area of the object (its head-on profile), ρ is the density of air (about $1.2\ \mathrm{kg\ m}^{-3}$), and v is the velocity of the object.[1] The direction of the force is opposite to the velocity of the object. C is related to the shape of the object, for example, whether or not it is streamlined. Experiments show that for smooth or moderately smooth objects (such as a baseball) with circular cross sections ($A = \pi r^2$), C is approximately 0.46.[2,3] The derivation of this formula is beyond the scope of this book. Suffice it to say that the object moving through the air changes the momentum of the air molecules struck by the object. Thus the force on the object is related to A, ρ, and v^2. If you try to imagine what happens to the air when an object moves through it, the inclusion of these terms should be partly intuitive.

We can now use Newton's Second Law to obtain the downward acceleration of a spherical object of radius r. Thus

$$a = g - \frac{1}{2} C\rho \frac{\pi r^2}{m} v^2,$$

where m is the mass of the falling object, and + is down. Notice that the drag term has a sign opposite to the g term. This is as it should be for a falling object, tending to slow the object during its descent.

Of most importance to us is the fact that when the air resistance is included in the acceleration, it is very difficult to use algebra and calculus to find an exact analytic solution for the velocity $v(t)$, and that is why the problem does not appear in most introductory physics textbooks. To provide an algebraic solution to the falling-body problem, most physics textbooks assume air resistance is negligible, and they give this solution:

$$v(t) = v_0 + gt.$$

But the air resistance is not negligible, as paratroopers, sky divers, flying squirrels, katydids, and many smaller creatures might attest. Fortunately the computer gives us an opportunity to find an approximate solution to this problem, and we will outline this approach in the next few paragraphs.

1.3 A New Look at a Falling Object

We begin with a comment about notation. In the material that follows we will frequently use function notation in which "$f(x)$" means f is a function of x. Likewise "$a(t)$" means the acceleration is a function of time, and it is read "a of t." The notation "$v(t)$" implies that the velocity is also a function of time.

We are ready to begin our work by looking at the mathematics of the falling-object problem. One concept from calculus is needed, the definition of the *derivative*. To refresh your memory, here is the definition of the derivative of a function f at x:

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h},$$

as it almost universally appears in calculus textbooks. This concept was, in fact, invented by Newton to solve physics problems. It would be worthwhile to locate and review this definition in your calculus book. Physicists are more prone to use the "double-d" form of the definition of the derivative. Thus

$$\frac{d}{dt} f(t) = \lim_{h \to 0} \frac{f(t+h) - f(t)}{h},$$

where we have used t as the variable to suggest that there are variables other than x. The quantity

$$\frac{f(t+h) - f(t)}{h}$$

is called the *difference quotient*. This concludes our brief review of the calculus you need to understand the next few paragraphs. Now we will use this these concepts.

The acceleration of an object is defined as follows:

$$a(t) = \lim_{h \to 0} \frac{v(t+h) - v(t)}{h},$$

where the quantity $[v(t+h) - v(t)]$ is the *change* in the velocity during the time interval h. Figure 1.1 illustrates this idea. (As an aside, note that this definition is identical to the definition of the derivative given above, with v replacing f, and t replacing x.)

If the time interval h is *sufficiently small* (but not zero), we may then make the following *approximation* for $a(t)$:

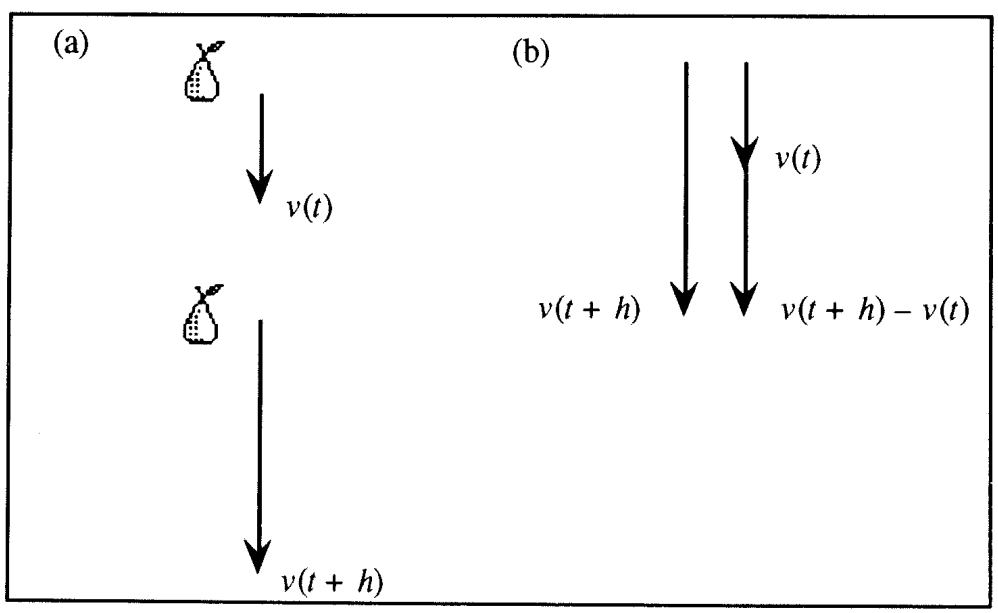
$$a(t) \cong \frac{v(t+h) - v(t)}{h}.$$

We replaced = with $\cong$ to symbolize the fact that we are making an approximation. After multiplying by h we may reorganize this relationship to obtain another approximation, namely,

$$v(t+h) \cong v(t) + ha(t),$$

which is of *central importance* to what follows. Indeed, if $a(t)$ is constant during the time interval h, then this expression is exact, not an approximation.

FIGURE 1.1 • (a) The Meaning of $v(t)$ and $v(t+h)$ for a Moving Object. (b) The Relationship Between $v(t)$, $v(t+h)$, and $v(t+h) - v(t)$. (In the one-dimensional case, vectors may be treated as algebraic quantities, needing only a sign to indicate the direction. We take + as downward.)



As in Figure 1.1, we drop an object. Assume that we start our stopwatch ($t = 0$) when the object is released. At $t = 0$ let $v = v_0$ and $a = a_0$. Given the initial velocity v_0, the acceleration a_0 may be calculated using the formula obtained from Newton's Second Law,

$$a_0 = g - \frac{1}{2} C\rho \frac{\pi r^2}{m} v_0^2,$$

with, in this case, $v_0 = 0$. Next we use the approximation mentioned above to find the velocity at the end of the small time interval h. Thus

$$v_1 \cong v_0 + ha_0.$$

This new velocity allows us to find a new acceleration using Newton's Second Law. Specifically, the acceleration at time $t_1 = h$ is given by the expression

$$a_1 = g - \frac{1}{2} C\rho \frac{\pi r^2}{m} v_1^2.$$

With this new acceleration, a_1, a new velocity may be found using the approximation

$$v_2 \cong v_1 + ha_1.$$

A pattern is developing. You might guess what the next two equations will be; find a_2 with the formula

$$a_2 = g - \frac{1}{2} C\rho \frac{\pi r^2}{m} v_2^2,$$

and use it to find v_3:

$$v_3 \cong v_2 + ha_2.$$

It is convenient to drop the approximation symbol, $\cong$. Thus we will replace the last expression by

$$v_3 = v_2 + ha_2.$$

We can describe the entire sequence of calculations with two *general equations*, namely

$$a_n = g - \frac{1}{2} C\rho \frac{\pi r^2}{m} v_n^2$$

and

$$v_{n+1} = v_n + ha_n.$$

Expressed in words, "the velocity at time $(n+1)h$ is the velocity at time nh plus the acceleration at nh times the small time interval h." It is more awkward, but we may also phrase the last equation with function notation; thus

$$v[(n+1)h] = v(nh) + ha(nh).$$

The calculations are repeated at integer multiples of the small time interval h until we have a table (or graph) of v versus t, or said more simply, a graph of $v(t)$. The table of velocities

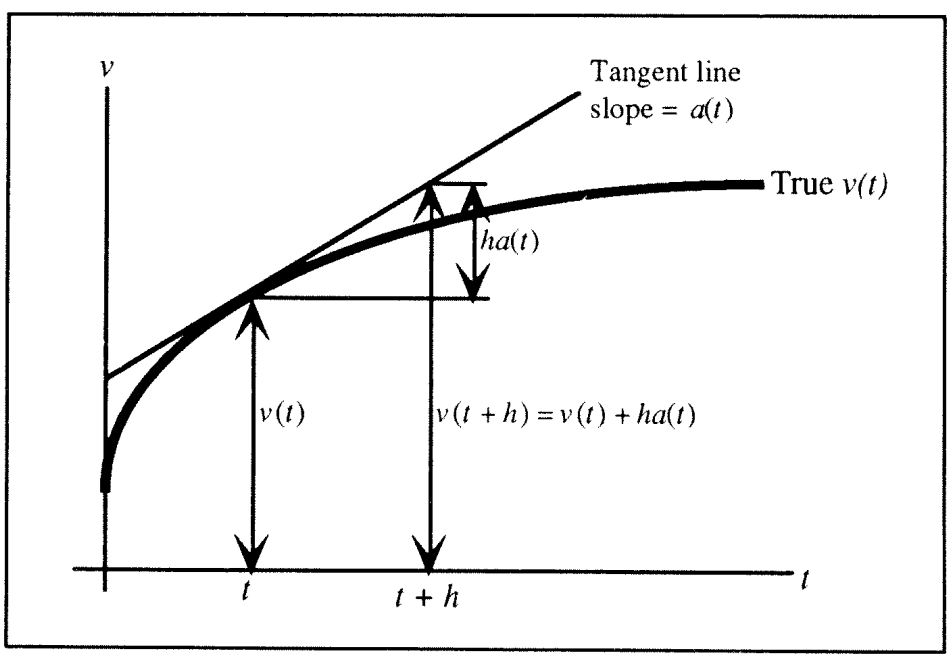
or the graph of velocity versus time, from $t = 0$ to some maximum time t_{max} (of our choosing), is called the *numerical solution* to the problem. It is called a numerical solution because it consists of a set of numbers rather than an equation, which is called an *analytic* solution. For example, the equation $v(t) = v_0 + gt$ is an analytic solution.

You should note that in order to begin the calculations we need v_0, or $v(0)$, the velocity at $t = 0$. This is called an *initial condition*. In the next few problems we will assume $v_0 = 0$.

Figure 1.2 is a graphical illustration of one step in the numerical solution, a step that takes us from $v(t)$ to $v(t + h)$. Assume the true $v(t)$ curve is the heavy line labeled $v(t)$. We approximate the solution at $t + h$ by multiplying the *slope* of the curve at t by the time interval h and adding this product to the velocity at time t. This is tantamount to assuming the true solution lies on the tangent line. *The slope of the graph of $v(t)$ at t is just the acceleration at that instant.* In Figure 1.2 we have exaggerated the error at $t + h$. The error is the distance between the tangent line and the curve labeled "True $v(t)$." In this figure h is clearly not "sufficiently small," but you can see that choosing a smaller value for h would improve our approximation. Of course, reducing h will also result in having more steps, and more computations, in the solution.

We are now ready to supply a procedure to calculate a numerical solution for the velocity of a falling object as a function of time. Figure 1.3 illustrates the procedure with a flowchart. An English language version of the procedure is provided in Algorithm 1.1 together with a generic computer program. A part of the numerical solution obtained with this algorithm is shown in Table 1.1. Taken together they should help you understand the calculations required to obtain a numerical solution.

FIGURE 1.2 • A Graphical Representation of the Technique for Finding a Numerical Solution for $v(t)$. Notice that we have shown only a single step in the solution, and we have greatly exaggerated the error.



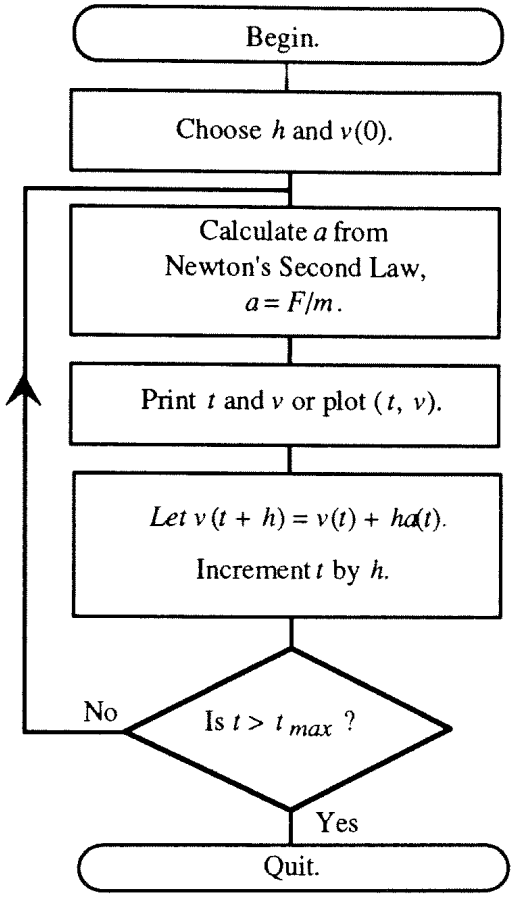


FIGURE 1.3 • Flowchart of the Procedure to Find a Numerical Solution with Algorithm 1.1

ALGORITHM 1.1 / Velocity of a Falling Object

STEP 1. Choose a value for the time interval h.
STEP 2. Choose the initial velocity, $v(0) = 0$.
STEP 3. Loop while $t \le t_{max}$.
STEP 4. Calculate the acceleration with the expression

$$a = g - \frac{1}{2} C\rho \frac{\pi r^2}{m} v^2.$$

STEP 5. Print the time t, acceleration a, and velocity v.
STEP 6. Multiply h and a and add the product to the current velocity to get the new velocity.
STEP 7. Increment the time by h.
STEP 8. Loop back to STEP 4 until $t > t_{max}$.
STEP 9. Quit.

A Generic Program to Implement Algorithm 1.1

```
h = 0.01
v = 0.
r = 75e-3
m = 3.9e-3
g = 9.8
C = 0.46
DO while t =< 15
   a = g -  0.5*C*1.2*pi*r^2/m*v^2
   PRINT t, a, v
   v = v + h*a
   t = t + h
LOOP
QUIT
```

TABLE 1.1 • Part of the Numerical Solution, $v(t)$, for a Falling Object. Parameters: h = 0.01 s, g = 9.8 ms^{-2}, C = 0.46, r = 0.01 m, m = 0.01 kg and v_0 = 0. Our computer program printed the results once for every 100 steps.

Time (s)	Acceleration (m/s/s)	Velocity (m/s)
0	9.8	0
.1	9.8	.98
.2	9.79	1.96
.3	9.77	2.94
.4	9.73	3.91
.5	9.67	4.88
.6	9.59	5.83
.7	9.5	6.78
.8	9.4	7.73
.9	9.28	8.65

PROBLEM 1.1 Velocity of a Falling Object • Implement Algorithm 1.1 with a computer program or spreadsheet, calculating the velocity of a falling object as a function of time. Assume the object is a marble with a radius of 7.5 mm and a mass of 3.9 g. Use $h = 0.01$ s. Calculate and print t, a, and v. Terminate the loop when $t > 15$ s. You should notice that the calculations take a lot of time. Speed up the program by printing the results every 0.1 s rather than every 0.01 s by printing only once every 10 times through the loop. It is also helpful to round the values of t, v, and a to two decimal places. After you have made these modifications, run the program once again, and use the numerical solution to answer the following questions:

- Is the acceleration of the marble even approximately constant as is usually assumed in falling-object problems?
- How does the acceleration change with time? Why?
- How does the velocity change with time? Why?
- What is meant by *terminal velocity*? Does the marble reach terminal velocity? How long does it take to reach terminal velocity?
- It has been reported that a vertical air velocity of 41.7 ms^{-1} will support a baseball in a vertical wind tunnel.[4] If that is true, what is the drag coefficient of a baseball, given that its mass is 0.145 kg and its diameter is 0.0732 m?
- It is sometimes surprising that small animals can fall to the ground with no apparent injury. Choose some small animal, a spider for example, and from its weight and size estimate its terminal velocity.
- What happens to the terminal velocity of a spherical object if its mass is doubled, tripled, or multiplied by ten, while the radius stays constant?

- What happens to the terminal velocity of an object if its cross section (area) is decreased by a factor of two? What if it is increased by a factor of two?

1.4 Some Important Programming Considerations

For computer programming purposes it is very important to notice there are patterns in the calculations we have just described. To help you recognize these patterns, we will elaborate on the calculations, using subscript notation. Assume that the initial condition, v_0, is given, and assume that we always calculate the acceleration from Newton's Second Law. Thus

$$a_n = g - \frac{1}{2} C\rho \frac{\pi r^2}{m} (v_n)^2 .$$

Here is how the pattern develops:

$$v_1 = v_0 + ha_0$$
$$v_2 = v_1 + ha_1$$
$$v_3 = v_2 + ha_2$$
$$\vdots$$
$$v_{n+1} = v_n + ha_n$$

Notice that each formula involves terms from the preceding formula. Let us show expanded versions of these formulas:

$$v_1 = v_0 + ha_0$$
$$v_2 = v_0 + ha_0 + ha_1$$
$$v_3 = v_0 + ha_0 + ha_1 + ha_2$$
$$v_4 = v_0 + ha_0 + ha_1 + ha_2 + ha_3$$
$$\vdots$$
$$v_{n+1} = v_n + ha_n = v_0 + ha_0 + ha_1 + ha_2 + \cdots + ha_n .$$

From this pattern you should see that, for example, to find v_8 from the expression

$$v_8 = v_7 + ha_7$$

means that we have already found the following sum,

$$v_0 + ha_0 + ha_1 + ha_2 + ha_3 + ha_4 + ha_5 + ha_6$$

to get v_7. We will call a sum such as

$$v_0 + ha_0 + ha_1 + ha_2 + ha_3 + ha_4 + ha_5 + ha_6 + \cdots + ha_n$$

a *running sum*. For conciseness, we may express our results this way:

$$v_n = v_0 + h[a_0 + a_1 + \cdots + a_{n-1}] = v_0 + h \sum_{i=0}^{i=n-1} a_i .$$

A running sum is easy to calculate with the aid of a programmable calculator, a spreadsheet, or a computer program. When programming in BASIC, FORTRAN, or Pascal, such a sum is implemented by placing in a loop instructions such as these:

```
LET v = v + h*a    BASIC,
v = v + h*a        FORTRAN,
v: = v + h*a       Pascal.
```

These instructions all mean the same thing: *Replace* v with $v + ha$, or, assign the value $v + ha$ to the variable v. Each time through the loop a new "ha" term is added to the previous sum. Thus the programing necessary to calculate a running sum and, therefore, a numerical solution, is remarkably simple.

Recall the general formula

$$v_{n+1} = v_n + ha_n .$$

In this form it is easier to visualize a spreadsheet solution. The contents of a cell in the $(n+1)$th row is determined (in part) by adding ha_n to the contents of an (upwardly) adjacent cell in the nth row. Specifically, if time is in Column A, velocity in Column B, and acceleration in Column C, then an expression such as

B7 = B6 + (A6 – A5)*C6,

with (A6 – A5) = h, will calculate a new velocity with a spreadsheet. Of course, the calculations may be arranged by columns rather than rows.

At this point we rephrase Algorithm 1.1 in both *function* and *subscript* notations. Pick the notation that seems simple to you. It is important to understand what is happening in Algorithm 1.1; that is why we have given several slightly different versions. Although there is no virtue in memorizing either the steps we have taken to get to this point nor the various forms of Algorithm 1.1, it is important to have a conceptual grasp of what we are doing. We are merely adding all the changes that take place in the velocity of an object, all the tiny "acceleration times time" terms. Examine the running sums once again, and notice all of the "ha_n" terms.

ALGORITHM 1.1 / Function Notation

STEP 1. Choose a value for h. Let $n = 0$.
STEP 2. Choose the initial condition, $v(0)$.
STEP 3. Begin a loop.

ALGORITHM 1.1 / Subscript Notation

STEP 1. Choose a value for h. Let $n = 0$.
STEP 2. Choose the initial condition, v_0.
STEP 3. Begin a loop.

STEP 4. Calculate $a(nh)$:

$$a(nh) = g - \frac{1}{2}C\rho\frac{\pi r^2}{m}[v(nh)]^2.$$

STEP 5. Print t, $a(nh)$, and $v(nh)$.
STEP 6. Calculate $v[(n+1)h]$:

$$v[(n+1)]h = v(nh) + ha(nh).$$

STEP 7. Calculate t and increment n:

$$t = nh \text{ and } n = n + 1.$$

STEP 8. Loop until $t > t_{max}$.
STEP 9. End.

STEP 4. Calculate a_n:

$$a_n = g - \frac{1}{2}C\rho\frac{\pi r^2}{m}v_n^2.$$

STEP 5. Print t_n, a_n, v_n.
STEP 6. Calculate v_{n+1}:

$$v_{n+1} = v_n + ha_n.$$

STEP 7. Calculate t and increment n:

$$t_n = nh \text{ and } n = n + 1.$$

STEP 8. Loop until $t > t_{max}$.
STEP 9. End.

1.5 About = and h

The computer programming notation

```
LET v = v + a*h     BASIC
v = v + a*h         FORTRAN
v := v + a*h        Pascal
```

is misleading. The equal sign is misrepresented in these instructions; these are neither mathematical equations nor forumulas. (At least the authors of Pascal included the colon to suggest that something other than equals was meant by the instruction.) In fact, these expressions would be more honestly put as $\neq$.

The problem is that = has various meanings. First, there are equations,

$$5x + 1 = 9,$$

for example, which may be solved for x. If, like this one, the equation admits only a few possibilities for x, then the equation is a *conditional* equation. If, like

$$\sin\left(\frac{\pi}{2} - x\right) = \cos x,$$

the equation is true for *all* x, then the equation is an *identity*. The equals sign also appears in formulas, such as

$$s = v_0 + \frac{1}{2}at^2.$$

Formulas are not to be solved in the same sense as a conditional equation; they merely make a statement about the relationship of several variables. Last, and incorrect, is the computer instruction of the form

$$v = v + ha,$$

which is meant to be neither conditional equation, identity, nor fomula. In this case = means "$v + ha$ replaces v" or "assign the value $v + ha$ to the variable v."

An alternative notation is shown here:

$$v \leftarrow v + ha.$$

The $\leftarrow$ symbol means "replace." Thus the preceding expression says "$v + ha$ replaces v," or "store $v + ha$ in the memory location (or cell) identified by v." The arrow implies that something is *going* somewhere, and indeed, the sum $v + ha$ is going into the memory location identified by v. With this notation and the realization that the expressions

$$v \leftarrow v + ha$$

and

$$v_{n+1} = v_n + ha_n$$

mean the same thing, we can put Algorithm 1.1, which follows, in its final form, a form that we will use most frequently in this book. Our new form eliminates the need for function notation and most of the subscripts, except perhaps to specify the initial conditions. The running sums are calculated correctly, the form is concise, and it means what it says. Notice that the acceleration is, indeed, calculated by a *formula*, but the velocity and time are calculated with "replacement" expressions.

ALGORITHM 1.1 / Velocity of a Falling Object

STEP 1. Choose a value for h.
STEP 2. Choose the initial condition, v_0.
STEP 3. Begin a loop.
  STEP 4. Calculate a:

$$a = g - \frac{1}{2} C\rho \frac{\pi r^2}{m} v.$$

  STEP 5. Print t, a, and v.
  STEP 6. Update v:

$$v \leftarrow v + ha.$$

  STEP 7. Increment t:

$$t \leftarrow t + h.$$

STEP 8. Loop until $t > t_{max}$.
STEP 9. End.

Spreadsheet users need to see a message in our algorithms that is different than that seen by computer programmers. For example, $v \leftarrow v + ha$ will mean "downward" to a spreadsheet

user. Perhaps it would be wiser to express this relationship with

$$v \downarrow v + ha$$

for spreadsheets for which time increases downward from cell to cell. In this case this instruction means "add ha to the contents of this cell in the v column, and place the sum in the next cell down in the v column." The instruction to loop also has a different meaning in a programming language environment and in a spreadsheet environment. In a spreadsheet environment "loop" means "move to the next row (or column) of cells."

We turn next to h. What value should we use for it? Looking again at Figure 1.1 you will see that if h is too large, significant errors are introduced because the acceleration is not constant during the time interval h, contrary to the assumptions we made. These errors are called *truncation errors* because, throughout the interval h, $a(t)$ is being represented by a constant, removing (truncating) any linear, quadratic, exponential or other terms in its representation.

On the other hand, if h is too small, the number of calculations performed by the computer is large, hence time consuming and subject to the accumulation of round-off errors. A computer uses a finite number of digits to store numbers and to perform arithmetic calculations. Numbers requiring repeating digits and irrational numbers such as the square root of two can never be represented exactly. (In binary arithmetic, which most computers use, even the decimal number 0.1 requires repeating binary digits, and therefore cannot be represented exactly.) The error associated by representing numbers approximately is called *round-off error*. There are error-analysis techniques to help estimate round-off errors, there are other algorithms that produce smaller truncation errors, and there is an extensive body of literature, much of which is beyond the scope of this book, devoted to both types of errors. In short, making a choice of h is complex. We do have four suggestions however:

- Solve a similar problem for which an analytic solution exists and choose an h small enough so that the numerical solution and the analytic solution agree to your satisfaction.
- Examine the solutions for different values of h. The solutions may vary with h. If the solutions appear to converge to a particular solution as h is decreased, that is the most likely the correct solution.
- Use the conservation of energy as an independent check on the accuracy of the calculations. We will describe this technique in more detail in Chapter 5.
- Sometimes the physics of the problem gives you some idea of how fast the solution changes. Choose an h small compared to the time scale over which you expect the solution to change.

1.6 Graphing the Velocity and Acceleration of a Falling Object

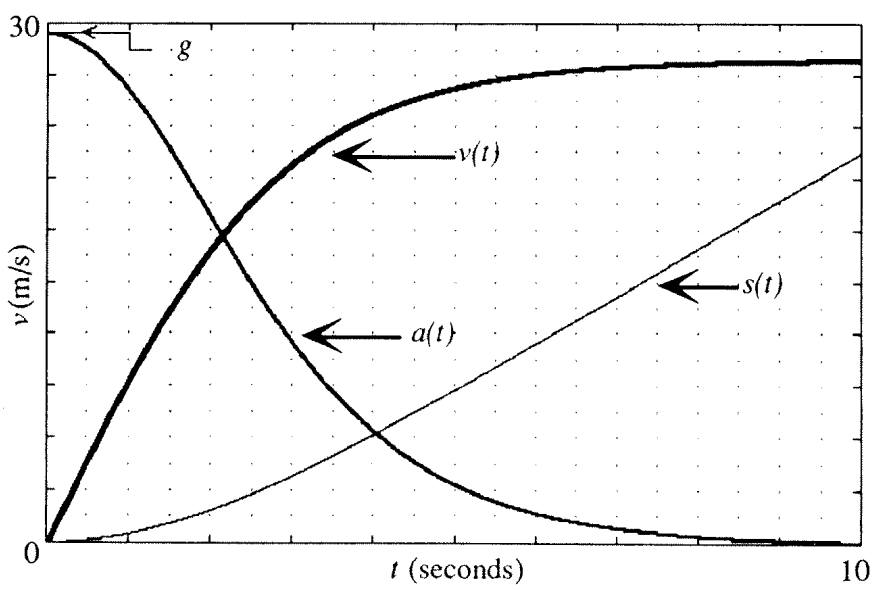
To find a numerical solution for $v(t)$ we must also calculate the acceleration at each step. The acceleration versus time data are also very interesting, and we will ask you to graph both $v(t)$ and $a(t)$. This may be an appropriate moment to describe more of our notation conventions.

A graph of $v(t)$ will mean the same thing as a graph of v versus t; both have t along the x-axis and v along the y-axis. Since so much of our work uses graphics, we need some kind of universal system to specify the limits on the x-axis and the y-axis. In the programs, regardless of which variables are being graphed, we will label the extremities of the x-axis as *xmin* and *xmax*, while the extremeties of the y-axis will be *ymin* and *ymax*. Because this is a bit cumbersome, in the text proper we will use *domain* and *range* expressed in interval notation, similar to but not identical with the meaning attached to these words in most calculus courses. Thus "*dom* = [–1, 1] m" means the x-axis extends from –1 to +1 and the units are meters. Likewise, "*ran* = [0, 50] ms^{-1}" means the y-axis extends from 0 to 50 and the units are those of velocity.

PROBLEM 1.2 Graphing the Solution • Relationships between variables are more easily grasped when graphed. Modify Algorithm 1.1 or the one you wrote for Problem 1.1 to graph v and a versus t. (In general this requires a simple change of a PRINT t, v instruction to a PLOT t, v instruction.) The output of our program is shown in Figure 1.2, where we have graphed $v(t)$, a(t) and $s(t)$. We will show how to calculate $s(t)$ in the next section.

In Figure 1.4 the scale and units of the vertical axis are for the graph of $v(t)$; the graphs of $s(t)$ and $a(t)$ are superimposed on the $v(t)$ curves so that we can study all three at once. In Section 1.8 we will describe how to calculate the distance the object falls. It is clear that for this object only a few seconds are required before the acceleration is significantly different than g. At what time does the object have a velocity within 10 percent of its terminal velocity?

FIGURE 1.4 • Velocity and Acceleration of a Falling Marble as a Function of Time. Parameters: r = 7.5 mm, m = 3.9 g, h = 0.02 s, C = 0.46, *dom* = [0, 10] s and *ran* = [0, 30] ms^{-1}. To fit on the velocity coordinate system, $a(t)$ was scaled upward by a factor of three, and $s(t)$ was scaled down by a factor of 10.



How do the $s(t)$ and $a(t)$ graphs change as the object reaches its terminal velocity? Study the curves and explain how you can tell from the slopes of the curve that

$$a(t) = \frac{dv}{dt} = v'(t)$$

and

$$v(t) = \frac{ds}{dt} = s'(t).$$

In these two expressions we have used both the "double d" notation and the prime notation to represent derivatives. We will continue to use both of these notations interchangeably. In fact, we have made an effort to keep the notation in this book consistent with that found in modern calculus texts.

1.7 Some "What If?" Questions

Use your computer programs to reply to the following questions, if necessary by running a program several times.

- What if the falling object is a ping-pong ball with $r = 0.019$ m and $m = 0.0024$ kg? Try to predict how the curves for the ping-pong ball will differ from the curves for a marble.
- Compare the $v(t)$ curve of a baseball with that of a tennis ball. A baseball has $r = 0.0366$ m and $m = 0.145$ kg, while a tennis ball has $r = 0.033$ m and $m = 0.055$ kg. Which has the largest terminal velocity?
- Why can baseballs be thrown farther by a human being than tennis balls or ping-pong balls? What is the approximate maximum distance a human being can throw a ping-pong ball? A baseball is much more massive than a ping-pong ball, yet it can be thrown farther. Why?
- What is the terminal velocity of a grain of sand? We found a grain of beach sand with $m = 1.7 \times 10^{-6}$ kg and $r = 0.8 \times 10^{-3}$ m. To answer this question you may wish to decrease h to 0.002 or less and rescale the domain and range of the graph. Incidentally, the formation and structure of sand dunes are related to the size and mass distributions of sand grains as well as the prevailing wind conditions.
- What if the falling object is a spherical speck of dust whose mass is 1/100 of the grain of sand and whose density is approximately that of water? You may want to rescale a, v, or s to make them all fit on the graph. Do you think the model of air resistance we used is applicable to dust particles?
- How fast does a raindrop fall? Do big raindrops fall faster than little ones?
- What happens to the terminal velocity of a particle if its radius is decreased while its mass remains constant? Run your program with several different values of the radius.
- What happens to the terminal velocity of a particle if its mass is increased while its density remains constant?

- Is $v_t \propto \sqrt{m}$ or is $v_t \propto m$? If m is the mass of the falling object and v_t is its terminal velocity, what is the nature of the relationship of v_t and the cross-sectional area of the object?

1.8 Finding $s(t)$ for a Falling Object

It is quite easy to find the distance fallen, $s(t)$, as a function of time. Once again, we begin with the definition of the derivative, this time a definition for the *instantaneous velocity*:

$$v(t) = \lim_{h \to 0} \frac{s(t+h) - s(t)}{h}.$$

Just as in the case of acceleration, we turn this equation into an approximation. Thus for a sufficiently small time interval h,

$$v(t) \cong \frac{s(t+h) - s(t)}{h},$$

which we rewrite as follows:

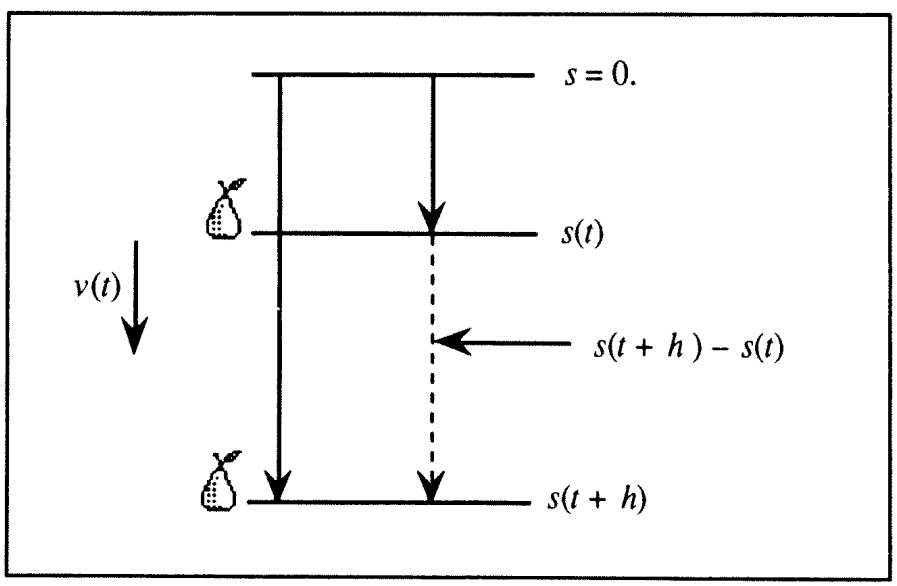
$$s(t+h) = s(t) + hv(t),$$

and we have replaced the approximation symbol with an equal sign. In subscript notation the last equation is

$$s_{n+1} = s_n + hv_n.$$

Refer to our calculations for the velocity if you need to refresh your memory on these steps. Figure 1.5 illustrates the main ideas.

FIGURE 1.5 • $s(t)$, $s(t+h)$, and $[s(t+h) - s(t)]$ for a Falling Pear.



The last equation becomes a series of calculations:

$$s_1 = s_0 + hv_0$$

$$s_2 = s_1 + hv_1$$

$$\vdots$$

$$s_{n+1} = s_n + hv_n.$$

It will be more convenient to use the velocity at the end of the interval. Thus in subscript notation,

$$s_{n+1} = s_n + hv_{n+1}.$$

The steps in the numerical solution are, therefore,

$$a_n = \frac{F}{m} = g - \frac{1}{2}C\rho\frac{\pi r^2}{m}v_n^2,$$

$$v_{n+1} = v_n + ha_n,$$

and

$$s_{n+1} = s_n + hv_{n+1}.$$

We are ready to describe a procedure to find the complete solution of the falling object problem. Our result is Algorithm 1.2. As in the case of Algorithm 1.1, we provide a generic computer program next to the algorithm so you can see how the algorithm translates into a computer program.

ALGORITHM 1.2 / Acceleration, Velocity, and Position of a Falling Object

STEP 1. Choose a value for the time interval h.
STEP 2. Let $v_0 = 0$, and $s_0 = 0$.
STEP 3. Begin a loop. Loop while $t \le t_{max}$.
STEP 4. Calculate the acceleration:

$$a = g - \frac{1}{2}C\rho\frac{\pi r^2}{m}v^2.$$

STEP 5. Print t, s, v, and a.
STEP 6. Update the velocity calculation:

$$v \leftarrow v + ha$$

A Generic Program to Implement Algorithm 1.2

```
h = 0.01
v = 0
s = 0
g = 9.8
C = 0.46
m = 3.9e-3
r = 75e-3
DO while t =< 15
   a = g - 0.5*C*1.2*pi*r^2/m*v^2
   PRINT t, s, v, a
   v = v + h*a
   s = s + h*v
   t = t + h
LOOP
END
```

STEP 7. Update the distance calculation:

$$s \leftarrow s + hv.$$

STEP 8. Increment the time:

$$t \leftarrow t + h.$$

STEP 9. Loop back to STEP 4.
STEP 10. Quit.

PROBLEM 1.3 Finding $s(t)$ for a Falling Object • Using Algorithm 1.2, modify the programs you have written to calculate and graph $s(t)$ as well as $v(t)$ and $a(t)$, then rerun the programs. Here are some parameters for a styrofoam ball: $m = 5$ g, $r = 3$ cm, $h = 0.01$ s, $v_0 = s_0 = 0$, $dom = [0, 5]$ s, and $ran = [0, 10]$ ms^{-1}.

PROBLEM 1.4 Experimental Confirmation • Make measurements of the time of fall for an umbrella dropped from several heights. Try dropping the umbrella right-side up and upside down. Plot your results on a graph of $s(t)$. Measure the cross-sectional area of the umbrella and its mass. Run your program with various values of C to obtain the best fit of the theoretical curve to the points you plotted. Does our theory agree with the experiment? Filters for coffee makers also make suitable objects for dropping.

PROBLEM 1.5 How Good Is the Solution? • The table produced by the program from Problem 1.1 and the graph produced by the program from Problem 1.2 are approximations to the solution. A critical assumption is that the acceleration and the velocity are constant if a *sufficiently small* time interval h is chosen. What is sufficiently small? To help answer this question, remove the drag term from the acceleration, run the program again, and compare the computed table or graph of $s(t)$ with what you get with a calculator and the formula $s = 1/2gt^2$. Repeat this for several different step sizes from 5 to 0.001. What do you think is a sufficiently small time interval?

1.9 Summary

To find a numerical solution to a dynamics problem, use the definition of the derivative and the derivative definition of acceleration to obtain the following approximation:

$$v(t + h) \cong v(t) + ha(t).$$

A numerical solution for $v(t)$ is found when this approximation is applied step by step, using Newton's Second Law for the acceleration.

The limit (or derivative) definition of velocity yields the approximation

$$s(t + h) \cong s(t) + hv(t).$$

A numerical solution for $s(t)$ is found when this approximation is applied step by step, using the numerical solution of $v(t)$.

Successive applications of the two preceding expressions yield running sums. Thus

$$v_n = v_0 + ha_0 + ha_1 + \cdots + ha_{n-1} = v_0 + \sum_{i=0}^{i=n-1} a_i h = v_0 + h \sum_{i=0}^{i=n-1} a_i,$$

and it easy to calculate this sum on a computer. The running sum for $s(t)$ is

$$s_n = s_0 + hv_1 + hv_2 + \cdots + hv_n = s_0 + \sum_{i=1}^{i=n} v_i h = s_0 + h \sum_{i=1}^{i=n} v_i.$$

Note the similarity of the v running sum to the integral definition of v, namely

$$v_n = v_0 + \int_0^{t_n} a(t)\, dt..$$

Also note the similarity of the s running sum with the integral definition of s, namely

$$s_n = s_0 + \int_0^{t_n} v(t)\, dt,$$

where h in the sum is analogous to dt in the integral. Because the sums that we find to produce a numerical solution are approximations to integrals, the process we have been describing in this chapter is sometimes called *numerical integration*.

When should numerical techniques be employed? When analytic techniques do not work, when numerical techniques are simpler, and when numerical analysis is faster. Rather than being mutually exclusive, analytic and numerical analysis should complement each other.

1.10 Additional Problems

PROBLEM 1.6 Effect of Drag on Objects Moving Horizontally (Challenging) • Assume that objects such as baseballs and tennis balls are projected horizontally with an initial velocity v_0, and for simplicity assume that gravity does not pull them to the ground. Modify the programs to study the effect of drag on the horizontal motions of these objects. Write a paragraph describing your results.

PROBLEM 1.7 Projectile Motion (Challenging) • Study the effect of drag on objects that are projected straight upward into the air with some initial velocity. Compare their trajectories with the motion of these objects in a vacuum.

PROBLEM 1.8 More Experiments (Challenging) • Try some distance-time measurements of objects descending in water and compare these measurements with calculations made on the computer. Aside from making sure the objects sink, how will you take buoyancy

into account? Once again, the measured time of fall should be longer than the time it takes to reach terminal velocity in water. Construct your experiment with that in mind. Compare your experimental results with those obtained from computational experiments.

Problem 1.9 An Additional Complexity (Challenging) • Studies of objects moving through fluids show that at low speeds there is a "v" term in the damping force.[2] This means that Newton's Second Law includes a force proportional to the velocity. This force may be described with a "$-Av$" term where A is a constant. Thus

$$ma = mg - Av$$

is the appropriate expression of Newton's Second Law. Let $A = 3.1 \times 10^{-4}\ \mathrm{kgm^{-1}s^{-1}}\ r$, where r is the radius of the object. Investigate how this term modifies the numerical solutions to the falling-object problems and express your conclusions in writing. Does an object dropped in water behave like it has an "Av" term or a "cv^2" term in its equation of motion?

1.11 References

[1] F. W. Sears, M. W. Zemansky, and H. D. Young, *University Physics Seventh Ed.* (Reading, MA: Addison-Wesley, 1987), 171.

[2] A. P. French, *Newtonian Mechanics* (New York: W. W. Norton & Co., Inc, 1971), 214.

[3] R. G. Watts and S. Baroni, "Baseball-Bat Collisions and the Resulting Trajectories of Spinning Balls," *Am. J. Phys.* 57 (January, 1989): 40.

[4] C. Frohlich, "Aerodynamic Drag Crisis and its Possible Effect on the Flight of Baseballs," *Am. J. Phys.* 52 (April, 1984): 325.

1.12 Programs

Program 1.11.1 implements Algorithm 1.1. It simply prints a table of t, $a(t)$ and $v(t)$ for a falling object. All of the parameters are defined in the program. Clearly one could modify this program to input C, m, r, and r, as well as the initial conditions. Program 1.11.2 graphs the numerical solution of a falling object. With suitable scaling, all three quantities, $a(t)$, $v(t)$, and $s(t)$ are graphed on the same coordinate system. The routine COMPHYS is a library of graphics subroutines that our programs frequently call (see the Supplement/Graphics Chapter at the end of this text).

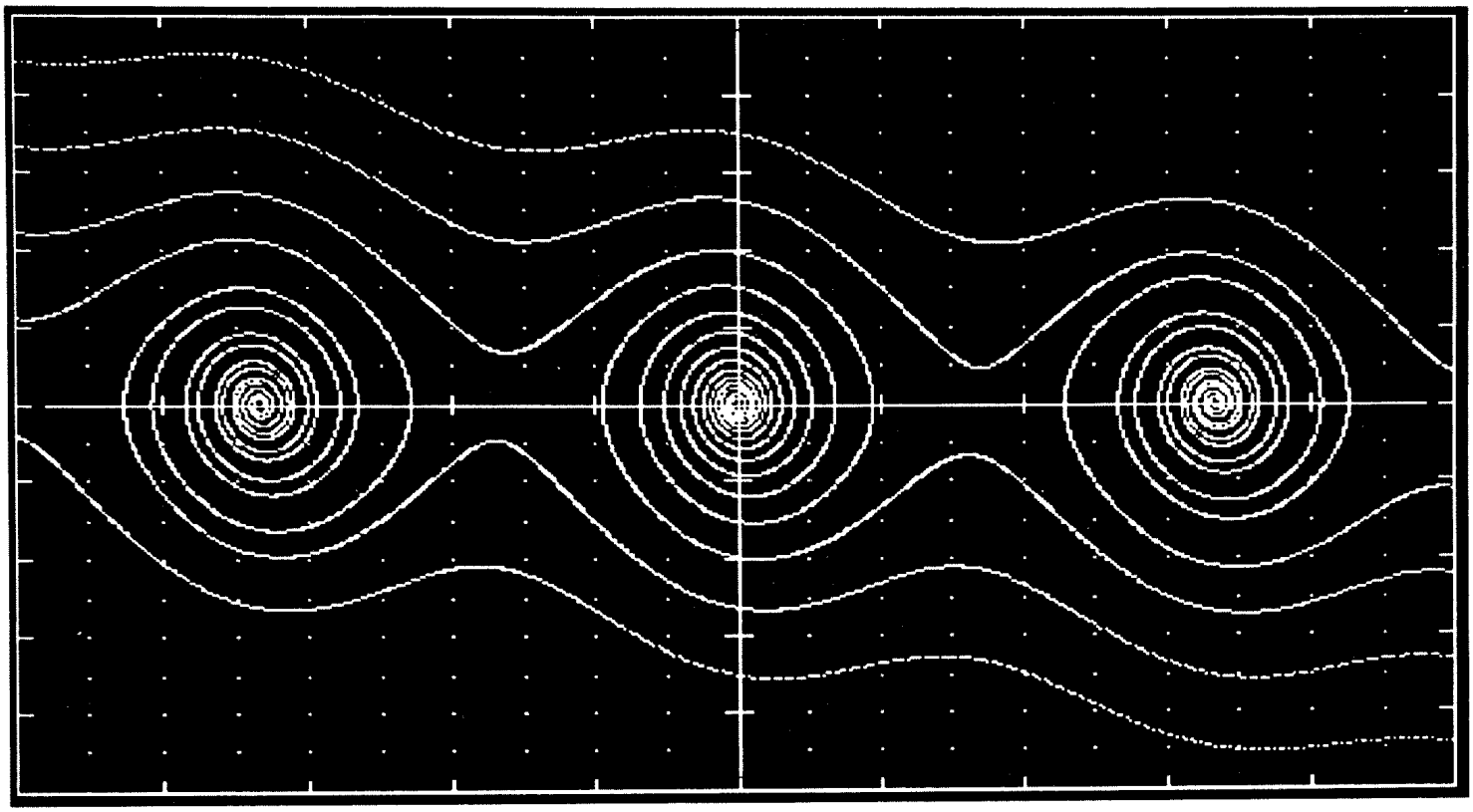
PROGRAM 1.11.1 / Motion of a Falling Object

```
! Algorithm 1.1, Problem 1.1.
LET h = 0.01                                  ! Make the small time interval 0.01
                                                seconds.
LET g = 9.8                                   ! g will be the acceleration of
                                                gravity.
LET C= 0.46                                   ! Drag coefficient.
```

```
LET R = 0.0075                                          ! Circular cross section with R =
                                                          7.5 mm.
LET m = 0.0039                                          ! Object has a mass of 3.9 g.
LET rho = 1.2                                           ! Density of air.
LET D = (0.5*C*pi*rho*R^2)/m                            ! Calculate this constant once and
                                                          for all.
DO while t <= 15                                        ! This is the main loop.
   LET a = g - D*v^2                                    ! Formula for the acceleration.
   If MOD (J, 10) = 0 THEN PRINT t, ROUND(a,2), ROUND(v,2)  ! Print every 10th result.
   LET v = v + a*h                                      ! Sum the a*h terms to get v.
   LET t = t + h                                        ! Increment the time.
LOOP                                                    ! Loop until t is 15 seconds.
END
```

PROGRAM 1.11.2 / Graphing the Numerical Solution of the Falling Object Problem

```
! Algorithm 1.1 and 1.2, Problems 1.2 to 1.5.
! Use domain [0, 15] and range [0, 30].
Library "Comphys"                                       ! Graphing routines.
CALL Open(xmin,xmax,ymin,ymax,wxmin,wxmax,wymin,wymax)  ! In Comphys library.
LET h = 0.02                                            ! Make the small time interval 0.02
                                                          seconds.
LET g = 9.8                                             ! g will be the acceleration of
                                                          gravity.
LET C= 0.46                                             ! Typical value for the drag
                                                          coefficient.
LET R = 0.0075                                          ! Circular cross section with R =
                                                          7.5 mm.
LET m = 0.0039                                          ! Object has a mass of 3.9 g
LET rho = 1.2                                           ! Density of air.
LET D = (0.5*C*pi*rho*R^2)/m                            ! Calculate this constant once.
LET s = 0                                               ! Start at s = 0.
LET v = 0                                               ! Starting value for v.
DO while t <= xmax                                      ! This is the main loop.
   LET a = g - D*v^2                                    ! Calculate the new acceleration.
   PLOT t, v                                            ! Graph v(t).
   PLOT t, 4*a                                          ! Graph a(t).
   PLOT t, s/10                                         ! Plot s(t).
   LET v = v + h*a                                      ! Calculate the new velocity.
   LET s = s + h*v                                      ! Calculate the new position.
   LET t = t + h                                        ! Increment the time.
LOOP                                                    ! Loop until t exceeds xmax.
CALL LABEL (wxmin, wxmax, wymin, wymax)                 ! Label graph. In Comphys.
END
```



Phase Portrait

2 / One-Dimensional Dynamics: An Object on a Spring

2.1 Introduction

A review of the following topics may be helpful in this chapter:

- Vectors, vector arithmetic
- Displacement, velocity, and acceleration
- Newton's Second Law
- Hooke's Law
- Limits, derivatives, and integrals

Reviewing simple harmonic motion and the motion of a simple pendulum would also be helpful.

Because the ideas introduced in Chapter 1 are central to one of the major mathematical themes in mechanics and in this book, we proceed to examine motion problems with more rigor. We will need the definitions of displacement, $\mathbf{r}(t)$, velocity, $\mathbf{v}(t)$, and acceleration, $\mathbf{a}(t)$. These definitions make use of calculus. If you have not had calculus, an acceptable course of action is to accept our results on faith.

The displacement $\mathbf{r}(t)$ is a vector drawn from some reference point, usually the origin of the coordinate system, to the object whose motion is being described. In Figure 2.1 we show $\mathbf{r}(t)$ for a projectile in mid-flight and $\mathbf{r}(t+h)$, the displacement vector h seconds later. The vector shown as $[\mathbf{r}(t+h) - \mathbf{r}(t)]$ is found by using the definitions of vector addition and subtraction. We also show the velocity vector, $\mathbf{v}(t)$ for the projectile at time t, and the projectile's acceleration, $\mathbf{g}$. Although $\mathbf{v}(t)$ and $\mathbf{a}(t)$ are free vectors, that is, they may be drawn anywhere, sometimes it is easier to visualize them as emanating from the object.

We are now ready to define $\mathbf{v}(t)$ and $\mathbf{a}(t)$ in terms of $\mathbf{r}(t)$ and t. These definitions can be phrased in terms of *derivatives* or *integrals*. Here are the derivative definitions of $\mathbf{a}$ and $\mathbf{v}$:

$$\mathbf{a}(t) = \frac{d}{dt}\mathbf{v}(t) = \lim_{h \to 0} \frac{\mathbf{v}(t+h) - \mathbf{v}(t)}{h},$$

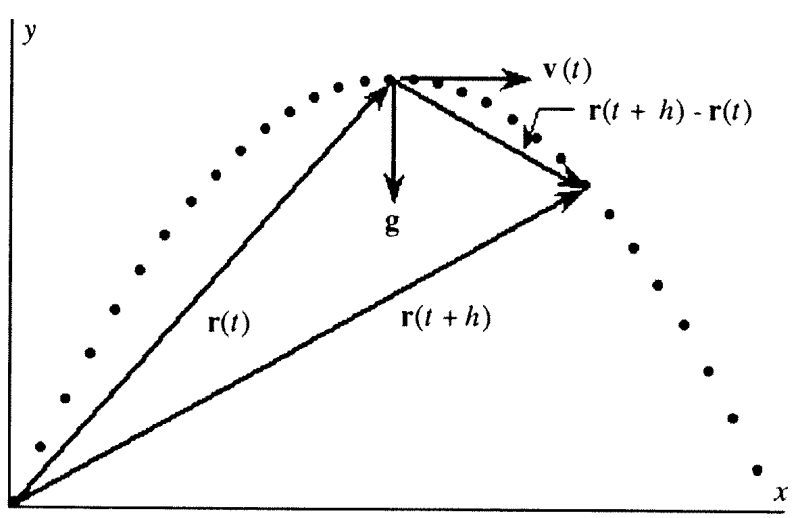
and

$$\mathbf{v}(t) = \frac{d}{dt}\mathbf{r}(t) = \lim_{h \to 0} \frac{\mathbf{r}(t+h) - \mathbf{r}(t)}{h}.$$

Study Figure 2.1 and note that as h tends toward zero, the vector difference quotient tends toward $\mathbf{v}(t)$, hence the definition.

The definition of a derivative is the only fact from calculus that we need to solve the physics problems in this and several subsequent chapters. However, to understand the reason

FIGURE 2.1 • Displacement, Velocity, and Acceleration Vectors for a Projectile.



why we chose a numerical solution rather than an analytic solution we need the integral forms of the relationships between $\mathbf{a}(t)$, $\mathbf{v}(t)$, and $\mathbf{r}(t)$. These relationships follow:

$$\mathbf{r}(t) = \int \mathbf{v}(t)\,dt,$$

and

$$\mathbf{v}(t) = \int \mathbf{a}(t)\,dt.$$

When dealing with motion in one dimension, we will drop the boldfaced vector notation in favor of the notation $a(t)$, $v(t)$, and $x(t)$, the latter being the one-dimensional displacement or position vectors. In two-dimensional problems we will work with the x and y components, $x(t)$ and $v_x(t)$ for example, of the vectors, rather than the vectors themselves. In this chapter we will be dealing with a one-dimensional problem; in Chapter 4 we will study two-dimensional systems. This completes our brief review of the concepts used in this chapter.

2.2 What's the Problem?

In this section we will describe more carefully the difficulty that is sometimes encountered in finding an analytic solution to dynamics problems. What we want to know about an object, a projectile or a planet, for example, is its position, velocity, and acceleration as a function of time, namely $x(t)$, $v(t)$ and $a(t)$. With this knowledge we can predict the position of the object at any later time. A common textbook problem in dynamics is solved as follows:

- Find the net force, $\mathbf{F}$, acting on the object and resolve this into x, y, and z components.
- Find the x, y, and z components of $\mathbf{a}$ from Newton's Second Law, $\mathbf{a} = \mathbf{F}/m$.
- Find the components of $\mathbf{v}$ by integrating $\mathbf{a}$, one integral for each of the x, y, or z components. If we are concerned only with the x-component, the integral is

$$v_x(t) = \int a_x(t)\,dt.$$

- Find the components of $\mathbf{r}$ by integrating $\mathbf{v}$ for each component. Once again we illustrate with the integral for the x-component of the motion; thus

$$x(t) = \int v_x(t)\,dt.$$

If $\mathbf{a}$ is a constant, or a function of time alone, it is usually easy to integrate it to find $\mathbf{v}$, and then integrate $\mathbf{v}$ to find $\mathbf{r}$. In non'calculus physics courses you are given the results of such integrations in the form of formulas. For example, if the acceleration is constant then

$$v(t) = v_0 + at,$$

and

$$x(t) = x_0 + v_0 t + \frac{1}{2}at^2,$$

where for the sake of convenience we have dropped the x-subscript symbolism for the x component.

In the case of a car or a projectile moving through air, the force **F**, and consequently the acceleration **a**, depends on **v**. In particular, at large enough speeds, **F** and **a** are proportional to the square of the magnitude of **v** and in the opposite direction, creating the following problem (refer to the integral definition of **v**). *To find* $\mathbf{v}(t)$ *we need to integrate* $\mathbf{a}(t)$, *but* $\mathbf{a}(t)$ *depends on* $\mathbf{v}(t)$ *so we cannot integrate it until we know* $\mathbf{v}(t)$, *which is what we are trying to find.*

If our mathematical tools are limited to algebra, geometry, trigonometry, and calculus, then we must confine our work to simple and sometimes unrealistic problems. We use the physicists' license to idealize or simplify the world in order to make problems soluble. If **a** is a constant, or if we assign $\mathbf{a}(t)$ to be some function of time that can be easily integrated, then we can solve the problems. The reason why physics textbooks contain this kind of problem is because they develop and refine previously acquired algebra, trigonometry, and calculus problem-solving skills. These problems have analytic solutions—that is, answers that can be expressed as an equation,

$$x(t) = x_0 \sin(2\pi f t),$$

for example. However, with the help of a computer it is possible to find *numerical* solutions to these and other frequently more difficult and more realistic problems. A numerical solution to a problem exists as a table of values of the variables or as a graph. That is why computer graphics are so important. The kind of mathematical work we do in this chapter is from a branch of mathematics called *numerical analysis*. The purpose of this chapter is to give you some numerical analysis tools, tools that are sufficient for solving many problems you will encounter in physics. The methods you learn in solving these problems will actually be applicable to a wide range of physics, chemistry, and engineering problems. We turn next to the specific problem of a vibrating object-spring system.

2.3 An Object on the End of a Spring[1]

2.3.1 The Physics

Figure 2.2 illustrates an object of mass m resting on an idealized frictionless surface and constrained by a spring, which we assume to have a spring constant k. The spring constant is large if the spring is stiff, and small if the spring is easily stretched or compressed. If the object is pushed to the left, the force of the spring is toward the right, and it increases with the magnitude of the displacement x of the object from its equilibrium position at $x = 0$. When released from a position of compression or tension, the object will oscillate (vibrate) back and forth. This kind of motion is called oscillatory or vibratory motion. A car without shock absorbers behaves in this way.

Perhaps this seems like a trivial and uninteresting problem, but it has enumerable applications. Almost everything in nature vibrates: trees, bridges, trombones, butterfly wings, cars going over bumps, charges in an electrical circuit, buildings, and molecules.

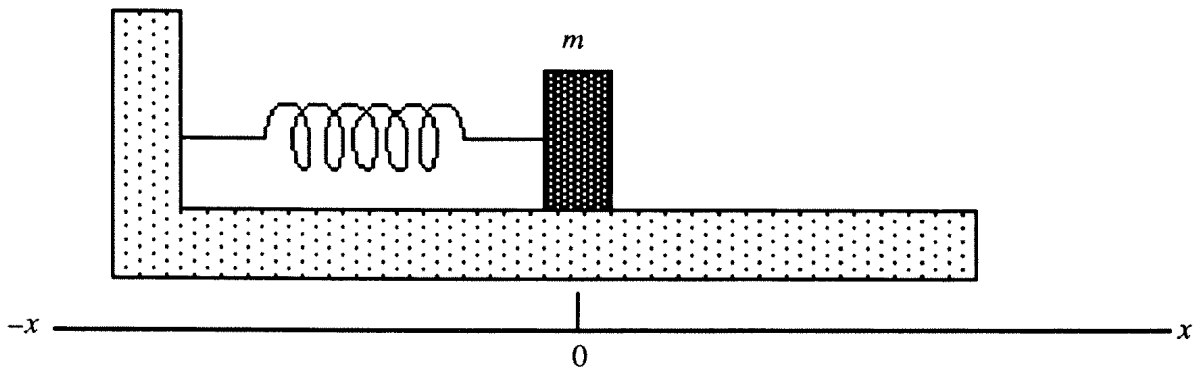


FIGURE 2.2 • Object on a Spring.

Studying a simple vibrating system such as the one in Figure 2.2 gives deeper insights into more complex systems.

If the object on the spring is pulled to the right, stretching the spring, the force exerted on the object by the spring is toward the left. We express this in the form known as Hooke's Law,

$$F = -kx,$$

where the minus sign means the force of the spring is opposite to the displacement, a fact you can experience when you stretch a spring. If the object is pulled to the right and released, the spring force causes the object to accelerate to the left. We may apply Newton's Second Law to this system and we find

$$-kx = ma.$$

Solving for the acceleration, a, gives

$$a(t) = -\frac{k}{m}x(t),$$

where we have used function notation to emphasize which quantities are (unknown) functions of time. Our task is to find $x(t)$ and $v(t)$ for all times after the spring is stretched and the object is released.

Applying the integral definition for acceleration gives this result:

$$v(t) = \int a(t)\,dt = \int -\frac{k}{m}x(t)\,dt = -\frac{k}{m}\int x(t)\,dt,$$

or

$$v(t) = -\frac{k}{m}\int x(t)\,dt.$$

Now we should see trouble. What we want to find is $x(t)$. The first step in finding $x(t)$ is to find $v(t)$ by performing the previous integration. But notice that the integrand of $v(t)$ is $x(t)$, which is what we don't know and are trying to find. The integral definitions of the kinematics variables are of no use to us as they stand.

2.3.2 The Mathematics

In this section we will show how an approximate numerical solution may be obtained directly from the limit definitions of velocity and acceleration. Refer to the definitions of $\mathbf{v}(t)$ and $\mathbf{a}(t)$ in Section 2.1. Because we are dealing with one-dimensional systems in this chapter, we drop the vector notation and exchange $x(t)$ for $\mathbf{r}(t)$. This will give the modified definitions that follow:

$$v(t) = \lim_{h \to 0} \frac{x(t+h) - x(t)}{h},$$

and

$$a(t) = \lim_{h \to 0} \frac{v(t+h) - v(t)}{h}.$$

As h gets smaller and smaller the quantity

$$\frac{x(t+h) - x(t)}{h}$$

gets closer and closer to $v(t)$. Therefore a good approximation to $v(t)$ can be obtained by making h small and we may improve the approximation by making h even smaller. Realizing that we are making an approximation to the definition of $v(t)$ we rewrite the definition as follows:

$$v(t) \cong \frac{x(t+h) - x(t)}{h}.$$

A more useful form for this approximation is

$$x(t+h) \cong x(t) + hv(t).$$

It is the latter form that we will use. Expressed in words, the position x at a time $t + h$ is the position x at an earlier time t, plus the velocity multiplied by the time interval h. It is important to realize that this approximation follows directly from the definition of velocity as a *limit*. Here there are no subtle tricks.

Applying these same ideas to the definition of acceleration gives

$$a(t) \cong \frac{v(t+h) - v(t)}{h},$$

or

$$v(t+h) \cong v(t) + ha(t).$$

In other words, the velocity at time $t + h$ is the velocity v at an earlier time t, plus the acceleration multiplied by the time h.

We will describe, once again but in more detail, how we will use these equations. First we must know the *initial conditions* of the spring-object system. The initial conditions describe the state of the spring-object system when $t = 0$. They might read as follows: at $t =$

0, $v = v_0$ and $x = x_0$. (If the spring is simply stretched to x_0 and released, then $v_0 = 0$.) The initial acceleration is $-kx_0/m$, and from it we can find the velocity h seconds later.

Here is the scheme in subscript form: Start at $t = 0$ with x_0 and v_0 given, and make the following calculations, in order:

$$a_0 = -\frac{kx_0}{m}, \quad v_1 = v_0 + ha_0, \quad x_1 = x_0 + hv_1, \quad t_1 = 0 + h.$$

Proceed to the next set of calculations, printing or plotting the results as you go and repeating the process for as long as you wish:

$$a_1 = -\frac{kx_1}{m}, \quad v_2 = v_1 + ha_1, \quad x_2 = x_1 + hv_2, \quad t_2 = t_1 + h.$$

We can generalize this in the following way:

$$a_n = -\frac{kx_n}{m}, \quad v_{n+1} = v_n + ha_n, \quad x_{n+1} = x_n + hv_{n+1}, \quad t_{n+1} = t_n + h.$$

The calculations are repeated until we have as much of the $x(t)$ and $v(t)$ curves as we want.

2.3.3 A Computer Programming Note

Before using these equations to solve the object-spring problem, recall that a BASIC, FORTRAN, or Pascal instruction equivalent to the instruction

```
V = V + H*A,
```

will, when placed in a suitable loop, perform a running sum. For example this loop

```
FOR J = 1 TO 10
K = K + J
NEXT J
PRINT K
```

will produce the sum of the first ten counting numbers. That is just the kind of structure we need because to find the displacement x after n intervals we must have previously added all the little displacements that we calculated. For example, we note once again that finding x_9 means we have already added

$$x_0 + hv_1 + hv_2 + hv_3 + \cdots + hv_8$$

to get x_8. To perform this summation we will use the following generic instructions for v and x:

```
v = v + a*h
x = x + v*h.
```

With a spreadsheet, the running sums will have the form

B(N) = B(N – 1) + A3*D(N – 1),

where N and N – 1 symbolize row numbers, the cell A3 holds h, B is a column symbol, and B cells contain the velocity while column D cells contain the acceleration. When placed in

the fifth row, for example, the instruction in cell B5 will read

B5 = B4 + A3*D4.

Also, the spreadsheet instruction

C(N) = C(N – 1)+ A3*B(N)

will calculate a new position if the C column contains positions and the B column contains velocities. In cell C5 this instruction reads

C5 = C4 + A3*B5.

Now we are ready to write a procedure to find the numerical solution to the object-spring problem.

2.3.4 The Solution

A procedure to solve the object-spring problem is given in Algorithm 2.1. Once again, we use the ← notation to mean "replace." If you are using a spreadsheet, remember that "loop" means *repeat* the calculations in the next row of cells.

ALGORITHM 2.1 / Motion of an Object on a Spring.

STEP 1. Choose values for k and m. Our work will be simpler if we choose $k = 1$ and $m = 1$.
STEP 2. Choose a value for the time interval h.
STEP 3. Choose the initial conditions, for example $v_0 = 0$, and $x_0 = 1$ m.
STEP 4. Begin a loop. Loop while $t \leq t_{max}$.
STEP 5. Calculate the acceleration from

$$a = -\frac{k}{m}x.$$

STEP 6. Plot or tabulate the current value of t, x, v, or a.
STEP 7. Update v:

$$v \leftarrow v + ha.$$

STEP 8. Update x:

$$x \leftarrow x + hv.$$

STEP 9. Increment t:

$$t \leftarrow t + h.$$

STEP 10. Loop back to STEP 5.
STEP 11. Quit.

PROBLEM 2.1 Oscillations of an Object on a Spring • Use the algorithm above to write a program to find $x(t)$, $v(t)$, and $a(t)$ for a mass on a spring. For simplicity, make $m = 1$ kg and $k = 1$ Nm^{-1}. Experiment with the step size h, but begin with $h = 0.02$. Choose $v_0 = 0$ and $x_0 = 1$, *dom* = [0, 12] s, and *ran* = [−1.2, 1.2] m. Begin by having the program make a table of t, $a(t)$, $v(t)$, and $x(t)$ for t in [0, 12] s. Once the program is working in the print mode, then modify it to graph $x(t)$. In order to see how the solution evolves with time, make the program clear the screen and replace the coordinate system when $t > t_{max}$. Then answer the questions that follow:

- Does the curve you obtain resemble any function you already know? What function?
- How closely does this function represents our numerical solution?
- Add instructions to your program that will graph not only $x(t)$ but also $\cos(t)$. Is the cosine function a good approximation to your solution?
- Is the cosine function really a solution? Check your physics textbook and see how this problem is solved.
- What if $m = 1$ kg and $k = 2$ Nm^{-1}? How does the solution change? Write this in the form "If k is increased then ..."
- What if $m = 2$ kg but $k = 1$ Nm^{-1}? How does the solution change? Write this in the form "If m is increased then ..." Do these last two results seem reasonable to you in terms of your physical intuition?
- What happens to the solution when you change the step size? Try using values of h such as 1, 0.5, 0.1, or 0.05 s.
- Run the program again, this time plotting $v(t)$ on the same graph. Can you write an analytic solution for $v(t)$? If so, does this solution make sense in terms of the derivative definitions of $v(t)$ and $a(t)$? In other words, do the graphs show that

$$v(t) = \frac{dx(t)}{dt}$$

and

$$a(t) = \frac{dv(t)}{dt}?$$

- What happens to the solution if you change the initial conditions to $v = 1$ ms^{-1} and $x = 0$? Change the initial conditions and also introduce instructions to plot $\sin(t)$. Run the program with these new initial conditions. Is the numerical answer close to an analytic solution $x = \sin(t)$?
- What happens if you change the initial conditions to $v = 0.5$ ms^{-1} and $x = 0.5$ m?

You should realize that the program you have written in Problem 2.1 can be generalized to other kinds of motion, and we will investigate another one of these in the next chapter. A key step in Algorithm 2.1 is STEP 5. Calculate the acceleration from

$$a = -\frac{k}{m}x.$$

The acceleration a is that found by applying Newton's Second Law to the problem. Here is the important point: *in principal, any problem for which we can use Newton's Second Law to find the acceleration explicitly may be solved with the numerical techniques just described.*

PROBLEM 2.2 The Simple Pendulum • Use the same procedure you used to solve the object-spring problem to find a numerical solution to the motion of the simple pendulum. The rotational form of Newton's Second Law is

$$\tau = I\alpha,$$

where we have used a scalar equation because the axis of rotation of the simple pendulum is fixed. In this equation τ is the torque about the pivot point, I is the moment of inertia about the pivot, and α is the angular acceleration. For the simple pendulum this equation becomes

$$mgL\sin(\theta) = -mL^2\alpha,$$

where m is the mass of the pendulum bob, L is the distance from the pivot to the center of mass of the bob, and θ is the angle between the vertical and the line connecting pendulum to pivot. We wish to find θ as a function of time. Dividing by mL and rearranging terms gives

$$\alpha = -\frac{g}{L}\sin\theta.$$

To find the solution use the example of the object-spring problem with the following subsitutions:

$$a = -\frac{k}{m}x \quad \text{becomes} \quad \alpha = -\frac{g}{L}\sin\theta,$$

$$x \quad \text{becomes} \quad \theta,$$

and

$$v \quad \text{becomes} \quad \omega.$$

Of course, ω is the angular velocity of the bob about the pivot. In fact, in your program for the object-spring system you may simply reassign x to be the angular displacement θ, v to be the angular velocity ω, and a to be the angular acceleration α. For parameters, use $g = 9.8$ ms^{-2}, $L = 1$ m, $h = 0.1$ s, $\theta_0 = 0.1$ radian, $\omega_0 = 0$, $dom = [0, 4]$ s, and $ran = [-0.1, 0.1]$ radians. Refer to your textbook to find the formula for the period. Compare the calculated value with the period of the pendulum you measure from the graph of your solution. Replace $\sin\theta$ with θ in the equation of motion and find the numerical solution. How does this new solution compare with the previous solution?

2.4 About Analytic and Numerical Solutions (Optional)

We will add several incidental considerations in this section, and then, in the next section, we will proceed to complicate the problem by including air resistance. Yes, the object-spring problem does have an analytic solution; no, you can't find it by integrating the kinematics

definitions. The analytic solution is usually found by first rewriting Newton's Second Law, $-kx = ma$, in the form

$$\frac{d^2x}{dt^2} = -\frac{k}{m}x.$$

In this form Newton's Second Law is called a *differential equation*. If you have some expertise with trigonometric functions and differential equations, you will recognize that this particular differential equation has a solution of the form

$$x = A\sin(\omega t) + B\cos(\omega t),$$

where A and B are determined by the initial conditions and

$$\omega = \sqrt{\frac{k}{m}}.$$

It is also possible to use the conservation of energy to find a relationship between the variables that can be integrated to give an analytic solution.

Our purpose in doing the problem numerically, aside from providing some experience, is to show how close the numerical solution is to the analytical solution, giving you some confidence that the numerical approach works. Moreover, you should see that the numerical solution is no more difficult, perhaps even simpler, to find than the analytic solution, a good enough reason to use a numerical solution rather than an analytic solution. Frequently, a numerical solution will suggest an analytic solution. However, *there is a fundamental and important difference between numerical and analytic solutions. You cannot* ***prove*** *that a numerical solution is correct, whereas analytic solutions are provable.*

Differential equations are ubiquitous in physics, and they also pop up in many places in biology, chemistry, and engineering. For example, in physics we find differential equations when we study dynamics, heat transfer, propagation of sound waves, magnetic induction, electrical circuits, electromagnetic radiation, radioactive decay, and quantum mechanics. The differential equation is likely the most common of all the mathematical metaphors in physics. In each of the aforementioned topics the differential equation will have symbols for the variables that are different. It is, therefore, important to recognize the *form* of the problem we have solved in this chapter. Acceleration is the second derivative of the displacement; thus Newton's Second Law is a formula for the second derivative. In the object-spring problem this is expressed by the equation

$$\frac{d^2x}{dt^2} = -\frac{k}{m}x.$$

What we obtained is a numerical solution to this second-order differential equation by reducing it to two first-order differential equations, namely

$$\frac{dv}{dt} = -\frac{k}{m}x,$$

and

$$\frac{dx}{dt} = v.$$

The technique we have used is called Euler's method. If equations of this form are seen again, they may be profitably attacked with the same technique. There are faster and more accurate techniques to be found in numerical analysis textbooks.

To help recognize when the algorithms developed above apply to a problem, we will repackage our equations once more, this time in the form a mathematician would prefer. We have found a numerical solution to a second-order differential equation of the form

$$\frac{d^2y}{dx^2} = f(x, y, \frac{dy}{dx})$$

by reducing it to two first-order differential equations:

$$\frac{dz}{dx} = f(x, y, z)$$

and

$$\frac{dy}{dx} = z,$$

using Euler's method. If these equations are applied to our problem, y becomes the displacement x, x becomes the time t, z is the velocity, and

$$\frac{d^2y}{dx^2}$$

is the acceleration. For the present these facts are incidental, but knowing them may help recognize similar problems when they are encountered.

2.5 Complicating the Problem with Damping

At the low speeds encountered by vibrating systems in air, the force of the air resistance is proportional to the negative of the velocity rather than the negative of its square. Newton's Second Law becomes

$$F = ma = -kx - cv,$$

which gives

$$a = -\frac{k}{m}x - \frac{c}{m}v,$$

for the acceleration. Problem 2.3 below asks you to study oscillations with the damping term cv. In Problem 2.8 at the end of this chapter you will be asked to find solutions to the object-spring problem when the object rests on a surface with friction, giving rise to a damping force that is independent of the magnitude of v.

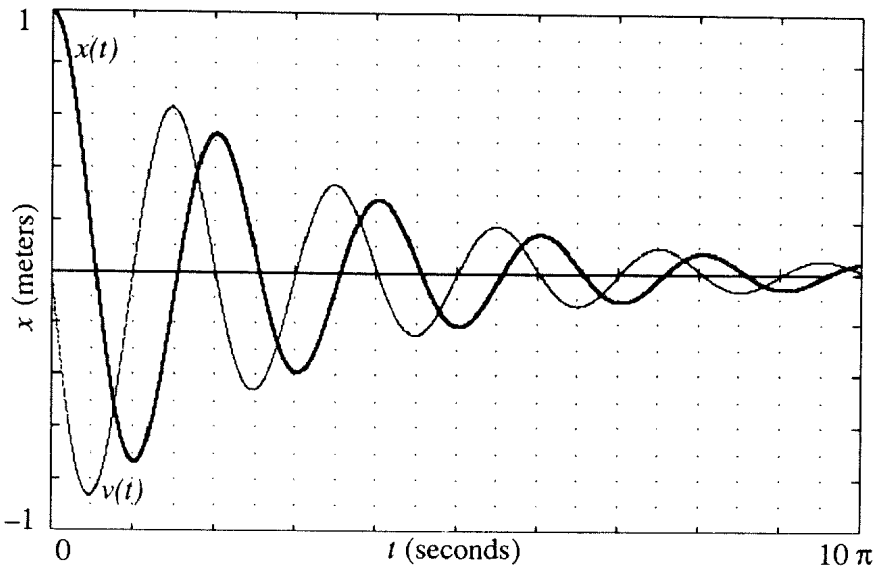


FIGURE 2.3 • Damped Oscillations of an Object on a Spring. Parameters: $k = 1$ Nm^{-1}, $m = 1$ kg, $c = 0.2$ kgs^{-1}, $x_0 = 1$, $v_0 = 0$, *dom* = [0, 10π] s, and *ran* = [-1, 1] m.

PROBLEM 2.3 A Damped Oscillator • Modify your object-spring program to include a $-cv/m$ term. Choose $k = 1$ Nm^{-1}, $m = 1$ kg, and $c = 0.1$ kgs^{-1} to start, but experiment with different values of both c and k.

Our results for an object-spring system with damping are shown in Figure 2.3, where we have drawn both $x(t)$, the heavy line, and $v(t)$. The $-cv/m$ term provides a force opposite to the velocity, and this force is zero when $v = 0$. This kind of damping would be close to the truth if the object were hanging from the spring in air. What role do shock absorbers on a car play? How do shock absorbers work? How would a car behave without damping?

PROBLEM 2.4 A Damped Pendulum • With damping, the simple pendulum's equation of motion is

$$mL^2\alpha = -mgL\sin\theta - c\omega,$$

where c is the damping coefficient. Notice the similarity of this equation to the equation of motion of an object on a spring. Find a numerical solution using the parameters in Problem 2.2 and $c = 0.1$ kgm^{2}s^{-1}. Study the effect of damping on the solution of the simple pendulum.

2.6 Phase Space

Sometimes the behavior of systems, such as the object-spring system, may be more clearly understood by graphing velocity versus displacement, $v(x)$, rather that displacement versus time, $v(t)$. This x-v coordinate system is known as *phase space*. A trajectory in phase space simply gives another perspective of a problem, and sometimes it is more valuable than the $x(t)$ trajectory in space time. A plot of the trajectory of the damped object-spring system in

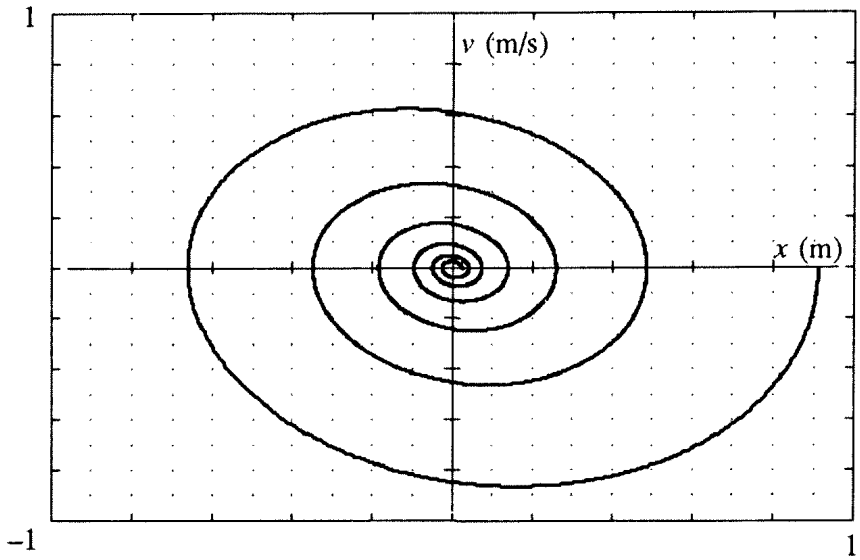


FIGURE 2.4 • Trajectory of an Object-Spring System in Phase Space with an Attractor at (0, 0). The parameters are the same as those in Figure 2.3.

phase space is shown in Figure 2.4. Initial conditions were $x_0 = 1$ m and $v_0 = 0$. The trajectory in phase space spirals in as the system loses energy to the surrounding air. For our object-spring system, the point (0, 0) is a *stable attractor*. No matter what the initial conditions, each time you start this system oscillating it will ultimately be drawn to the same point in phase space, (0, 0).

PROBLEM 2.5 The Trajectory of an Object-Spring System in Phase Space

- Modify your program to graph $v(x)$ rather than $x(t)$. This may be as simple as changing a plot (t, x) instruction to a plot (x, v) instruction accompanied by a change in the scale of the coordinate system; use $dom = [-1.1, 1.1]$ m and $ran = [-1, 1]$ ms^{-1}.
- On the phase-space trajectory, identify the initial point ($t = 0$) in the solution.
- Where is the object when the phase-space trajectory crosses the x-axis to the left of zero?
- Where is the object when the trajectory crosses the positive v-axis?
- Each point on a space-time trajectory (Figure 2.3) maps into a point on the phase-space trajectory (Figure 2.4). Make this mapping for several points in addition to the points you have already found.
- What makes the curve spiral toward the point (0, 0)?
- What set of points is the attractor if $c = 0$? (Note: an attractor may be more than a single point.)
- Explain how, if $c = 0$, the phase-space diagram can be drawn using the conservation of energy (COE) principle. In phase space what kind of curve is associated with the COE equation,

$$\frac{1}{2}mv^2 + \frac{1}{2}kx^2 = E,$$

where E is a constant?

PROBLEM 2.6 The Trajectory of the Simple Pendulum in Phase Space • Using the parameters in Problem 2.2, graph the trajectory of the simple pendulum in phase space. Identify points on the phase-space trajectory that correspond to the pendulum at the one end of its swing, the middle of its swing when it is moving the fastest, and the other end of its swing. Identify the attractor. Add a damping term to the system and identify the new attractor. The figure at the head of this chapter is a phase portrait of a damped simple pendulum.

2.7 An Improved Algorithm

It is possible to make an improvement in our technique of finding a numerical solution. We have been calculating the solution this way:

- At $t = 0$, $x = x_0$ and $v = v_0$.
- Calculate the acceleration a.
- Tabulate or plot the variables.
- Use a to calculate the velocity $v(t + h)$ from the expression

$$v_{n+1} = v_n + ha_n.$$

- Use the velocity to calculate the position

$$x_{n+1} = x_n + hv_{n+1}.$$

- Replace the current time t with $t + h$.
- End one loop in the procedure.

With this procedure, we use a at the beginning of an h-interval to find v at the end of an h-interval, then we use v at the end of the h-interval to find x at the end of an h-interval. We might reasonably expect that we would acquire greater accuracy if we would calculate x at the end of an interval by using v in the *middle* of an h-interval. Symbolically,

$$x_{n+1} = x_n + hv_{n+\frac{1}{2}},$$

where $v_{n+\frac{1}{2}}$ is the velocity halfway through an h interval.

How can we find $v_{n+\frac{1}{2}}$? The answer is simple; use the expression

$$v_{n+\frac{1}{2}} = v_{n-\frac{1}{2}} + ha_n$$

to find the velocity halfway through the h-interval.

There is a minor problem concerning how to start the calculations because substituting $t = 0$ into the preceding formula yields

$$v_{\frac{1}{2}} = v_{-\frac{1}{2}} + ha_0,$$

which means we must know v when $t = -h/2$, rather than when $t = 0$. To solve this problem, we will use the following special case to start the calculations:

$$v_{\frac{1}{2}} = v_0 + \frac{h}{2}a_0.$$

Now the algorithm may be completed; see Algorithm 2.2 for the one-dimensional procedure. We will call this method the Feynman-Newton method. It is also called the half-step method.

ALGORITHM 2.2 / Using the Feynman-Newton Method to Solve Dynamics Problems

STEP 1. Find the sum of the forces acting on the object and find the acceleration, $a = F/m$.
STEP 2. Choose the step-size h and the initial conditions v_0 and x_0.
STEP 3. Using the initial conditions, calculate the acceleration at $t = 0$.
STEP 4. Let

$$v_{\frac{1}{2}} = v_0 + \frac{h}{2} a_0 .$$

STEP 5. Begin a loop. Loop until $t > t_{max}$ (while $t \leq t_{\max}$).
STEP 6. Tabulate or plot $x(t)$, $v(t)$, or $a(t)$.
STEP 7. Update x:

$$x \leftarrow x + hv.$$

STEP 8. Using the current values of x and v, calculate a from Newton's Second Law.
STEP 9. Update v:

$$v \leftarrow v + ha.$$

STEP 10. Increment t:

$$t \leftarrow t + h.$$

STEP 11. End of STEP 5 loop.
STEP 12. Quit

PROBLEM 2.7 The Feynman-Newton Method • Solve one or two of the previous problems in this chapter with the Feynman-Newton method. Compare numerical solutions with analytic solutions for different values of the step-size h. By what factor can the step-size h be decreased with little or no loss of accuracy when the Feynman-Newton rather than Euler's method is used?

2.8 Additional Problems

PROBLEM 2.8 Damped Oscillations Revisited • In the object-spring problem we assumed a frictionless surface. Suppose that the surface is not frictionless, but assume that the coefficient of friction is independent of the magnitude of the speed. Also assume that air

resistance is zero. In the instruction for the acceleration we need a constant term having a sign opposite that of the velocity. Multiplying the c/m term by

$$-\frac{v}{|v|}$$

will suffice. This gives

$$a = -\frac{1}{m}kx - \frac{c}{m}\frac{v}{|v|}.$$

Run the program again with various values of c. Does this damping appear to be different from air damping? We have neglected static friction in this problem.

PROBLEM 2.9 Comparing Damping Curves • Choose $k = 20\ \mathrm{Nm}^{-1}$ and $m = 1$ kg. Produce and print one damped oscillation curve for air damping and another for surface-friction damping. For each graph sketch a curve, called the *envelope*, that just touches the tops of the oscillations. What is the difference in the curves between air damping and surface-friction damping? Is either one linear? Is either one exponential?

PROBLEM 2.10 Driven Oscillations (Challenging) • Suppose that there is some periodic driving force applied to our object-spring oscillating system. An example of a driven oscillator is a child in a playground swing being pushed by a parent. In the world of things we see and feel most oscillators are driven. Also assume that v damping is present, so Newton's Second Law becomes

$$ma = -kx - cv + F_0 \cos(\omega t),$$

where F_0 is the magnitude of the driving force and $\omega = 2\pi f$ is its angular frequency. Notice that we are continuously driving the oscillating system rather than giving it a bump each time.

Make this assignment:

$$d = \frac{F_0}{m},$$

giving

$$a = -\frac{1}{m}(kx + cv) + d\cos\omega t.$$

Replace the earlier expression for the acceleration with this one, run the program, and experiment with ω equal to one-half the natural frequency of vibration, ω_n, where

$$\omega_n = \sqrt{\frac{k}{m}} \quad \text{or} \quad f_n = \frac{1}{2\pi}\sqrt{\frac{k}{m}}.$$

Also try ω near ω_n, equal to ω_n, and ω five times ω_n. Study the region near ω_n quite closely. To begin, let $k = 1\ \mathrm{Nm}^{-1}$, $m = 1$ kg, $c = 0.1\ \mathrm{kgs}^{-1}$, $d = 2\ \mathrm{ms}^{-2}$, *dom* = [−1.2, 1.2] s, and *ran* = [−1.4, 1.4] m. Study this motion both in phase space and space time. What general conclusions can you draw about the behavior of the system when the driving force is close

to the natural frequency of the system? Write these conclusions down in the form "If ... then ...". Is there any reason for an army to break out of step as it crosses a bridge? What would happen if the army's step frequency were close to the natural frequency of vibration of the bridge? Why did the Tacoma Narrows bridge fail?[2]

PROBLEM 2.11 Projectiles and Air Resistance (Challenging) • We will attack this problem in more detail in Chapter 4, but it is possible to modify the programs in this chapter to deal with air resistance in the case of a projectile shot vertically into the air. Assume that

$$a = -g - \frac{c}{m} v^2 \frac{v}{|v|}.$$

Positive is upward. Start the clock ($t = 0$) with $x = 0$ but choose some initial upward (positive) velocity for v. Experiment with different values of c. Expand your work to include two-dimensional projectile problems.

2.9 References

[1] Some readers will recognize my indebtedness to *The Feynman Lectures on Physics* Vol. 1 (Reading, MA: Addison-Wesley, 1963), 9–4.

[2] An excellent reference for the Tacoma Narrows Bridge collapse is "Galloping Gertie: The Tacoma Narrows Bridge Collapse" by R. G. Fuller and D. A. Zollman in R. A. Serway's *Physics for Scientists and Engineers, Third Edition* (Philadelphia, PA: Saunders College Publishing, 1990), 352.

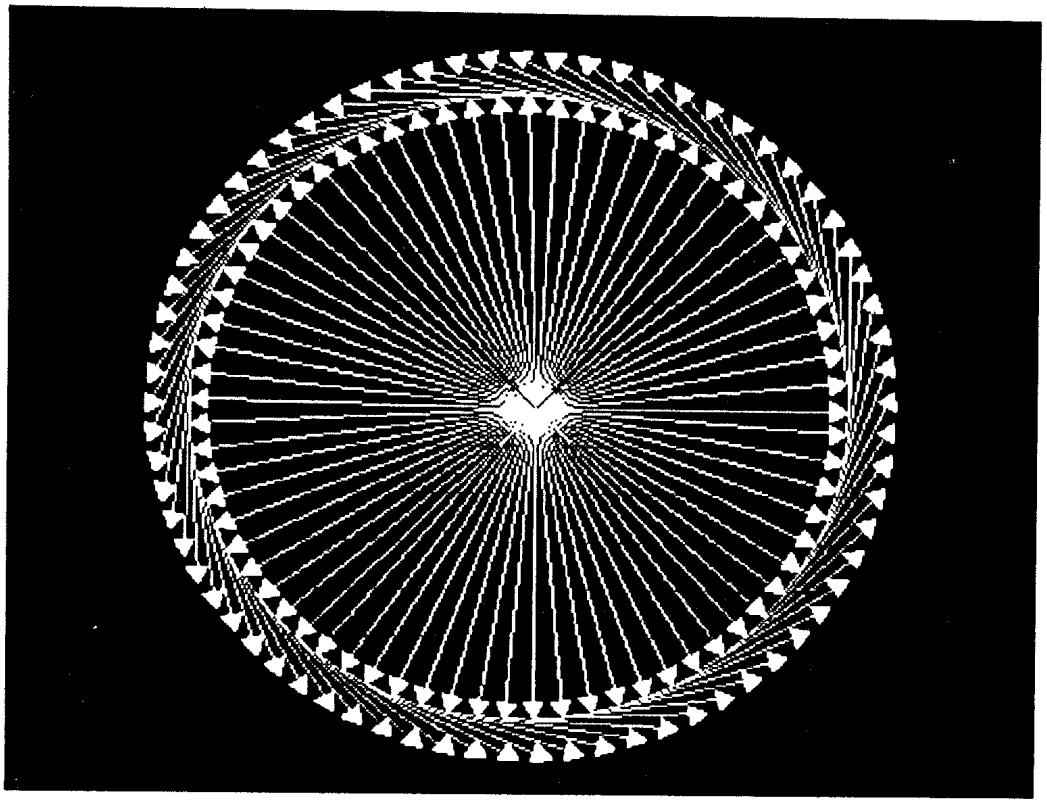
2.10 Programs

Program 2.10.1 is used to graph the solution to the object-spring problem. It is written with the Feynman-Newton method. This program allows for damping and driving forces. It can be modified to plot x, v, and a versus t, and it can be modified to make phase-space graphs. The formula for the acceleration is convenient to change because it is a Defined Function in BASIC. To show trajectories in phase space, remove the entire IF tp > xmax THEN...END IF sequence and use a PLOT X, V instruction rather than the PLOT tp, x instruction. Change the domain to [–1.2, 1.2] m. Pressing a key ends the calculations.

PROGRAM 2.10.1 / Motion of an Object on a Spring

```
! Algorithm 2.2.
! Begin with k = m = 1 and c = d =0. Then try c = 0.2, d=0. Finally try c = 0.1, d = 1.
! Use domain [0, 12], range [-1.2,1.2], and h = 0.05.
!.............
! Acceleration from Newton's Second Law is defined here. c is damping constant, d is the
! driving force, and w is the angular frequency, w = 2πf.
```

```
DEF a(x,v,t) = -k*x/m - c*v/m + d*cos(w*t)
!............
LIBRARY "Comphys"
CALL GET_INFO (x,v,k,m,c,d)                                  ! Get initial conditions and
                                                               parameters.
CALL Open(xmin,xmax,ymin,ymax,wxmin,wxmax,wymin,wymax)       ! Graphics routine in Comphys.
CALL Save (wxmin, wxmax,wymin,wymax,keep$)                   ! Hold the screen information.
LET w = 5*sqr(k/m)                                           ! Frequency of driving force.
LET h = 0.05                                                 ! Make the small time interval 0.05
                                                               seconds.
LET v = v + a(x,v,t)*h/2                                     ! Starting value for v using
                                                               Feynman-Newton.
DO until KEY INPUT                                           ! This is the main loop, exit by
                                                               pressing key.
   PLOT tp, x;                                               ! Plot the trajectory; connect
                                                               points.
   LET x = x + h*v
   LET v = v + a(x,v,t)*h
   LET t = t + h                                             ! Clock time.
   LET tp = tp + h                                           ! Plotting time.
   IF tp > xmax then
      PLOT                                                   ! Terminate the point connecting
                                                               sequence.
      LET tp = xmin                                          ! Start a new frame?
      CLEAR                                                  ! Clear the screen.
      CALL Show (wxmin,wymin, keep$)                         ! Bring the axes back.
   END IF
LOOP
GET KEY z                                                    ! Clear the keyboard.
CALL LABEL (wxmin, wxmax, wymin, wymax)                      ! Label graph.
END
!............
SUB GET_INFO (x,v,k,m,c,d)
    INPUT PROMPT "What are the initial conditions? Input xo and vo. ":x,v
    INPUT PROMPT "Please enter k and m. ":k,m
    INPUT PROMPT "Please enter the damping coefficient c. ":c
    INPUT Prompt "Please enter the driving force coefficient d. ":d
END SUB
```



Circular Motion

3 / A Computer Model of an Accelerating Car

3.1 Introduction

A review of the following topics may be helpful in this chapter

- Kinematics (displacement, velocity, and acceleration)
- Newton's Second Law
- Work-kinetic energy theorem (WKET)
- Friction
- Power

The following problem is a typical one for physics students. Such a problem is found in physics textbooks precisely because as long as complications like air resistance are ignored, the problem can be solved with algebra and simple kinematics formulas.[1] A police car is at rest by the side of the road. In it is a state trooper doing what police do best with high-tech equipment, catching speeders. With her radar gun the state trooper notes that a car is

approaching at 100 km/hr, a speed in excess of the speed limit on this particular section of the road. The state trooper accelerates her car from rest with a constant acceleration of 6 ms^{-2} in the hope of overtaking the speeder. If the police car starts at the instant the speeder passes and the speeder moves at constant velocity, how long will it take for the police car to overtake the speeder?

What does it mean to overtake? Let x_p symbolize the position of the police car and let x_s symbolize the position of the speeder. Let $x_p = x_s = 0$ and let $t = 0$ at the instant the speeder passes the police car. For some time, x_s will be greater than x_p, but eventually x_p will be equal to x_s. To overtake will mean $x_p \geq x_s$.

Although this problem may be solved with algebra and the formulas from kinematics, we will use the Feynman-Newton method introduced in Chapter 2, because when we add complicating factors we will need this technique. In Algorithm 3.1 we tailor the Feynman-Newton method for the problem of the speeder versus the police. You will notice that there are almost identical sets of instructions for both the police car and the speeder. Once again, we could use a simpler algorithm or algebra to solve the problem, but with this problem we choose to illustrate the Feynman-Newton method.

ALGORITHM 3.1 / Police versus Speeder

STEP 1. Decide on a value for h.

STEP 2. Choose the initial conditions.

- For the police car, at $t = 0$ let $x_p = 0$, $v_p = 0$, and $a_p = 6$ ms^{-2}.
- For the speeder, at $t = 0$, let $x_s = 0$, $v_s = 100$ km hr^{-1}, and $a_s = 0$.

STEP 3. Calculate the "half-step" velocity for the Feynman-Newton method.

- For the police car,

$$v_{p\frac{h}{2}} = v_{p^0} + \frac{h}{2} a_{p^0}$$

- For the speeder,

$$v_{s\frac{h}{2}} = 100 \text{ km h}^{-1}\left(\frac{1000 \text{ m}/\text{km}}{3600 \text{ s}/\text{h}}\right) = 27.8 \text{ m s}^{-1},$$

a constant.

STEP 4. Begin the main loop.

STEP 5. Tabulate or graph the solution.

- For the police car, tabulate or plot $x_p(t)$.
- For the speeder, tabulate or plot $x_s(t)$.

STEP 6. Apply the Feynman-Newton method to update the position.

- For the police car,
  add the distance the police car goes during the time interval h:

$$x_p \leftarrow x_p + hv_p.$$

- For the speeder,
  add the distance the speeder goes during the time interval h:

$$x_s \leftarrow x_s + hv_s.$$

STEP 7. Apply the Feynman-Newton method to update the velocity.

- For the police car,
  increment the velocity of the police car;

$$v_p \leftarrow v_p + ha_p.$$

- For the speeder's car,
  the velocity is constant so

$$v_s \leftarrow v_s + ha_s,$$

  with $a_s = 0$.

STEP 8. Update the time:

$$t \leftarrow t + h.$$

STEP 9. Loop back to STEP 4 until $x_p \geq x_s$.
STEP 10. Print t and x_p.
STEP 11. Quit.

PROBLEM 3.1 Police versus Speeder • Write a program to implement Algorithm 3.1 and find the time and position at which the police car overtakes the speeder. The simplest approach is to make a table with columns for time, distance, and speed. Be sure to find the speed of the police car when it overtakes the speeder. Is this a realistic speed?

3.2 A Matter of Horsepower

Perhaps a police car could reach the speed suggested by Problem 3.1, but there is a more serious question: Does the police car have the necessary horsepower? In this section we will relate the horsepower of the police car to its velocity and acceleration. The work-kinetic energy theorem (WKET) will help us find these relationships. Here is the WKET for the car if the *only* horizontal force acting on it is F:

$$\int_0^{x_t} F(x)\,dx = \frac{1}{2}mv_t^2 - \frac{1}{2}mv_0^2.$$

In this equation F is the horizontal force of the road on the car, a reaction to the force of the car on the road. The integral is taken from $x = 0$ to x_t, the position at time t. On the right-hand side of the equation v_t is the velocity at time t and v_0 is the initial velocity. All the quantities in this equation refer to the position and velocity of the center-of-mass of the car.

Automobile specifications do not include $F(x)$. What they do tell us is the *rate* at which *mechanical energy* can be delivered by the engine, its power, which we shall call P_c. Specifically, if we take the derivative of the WKET, the left-hand side is the power that appears in the automobile's specifications. Thus

$$P_c = \frac{d}{dt}\int_0^{x_t} F(x)\,dx.$$

We can make this statement because the force of the road on the car and the force of the car on the road are an action-reaction pair. In the case of a car accelerating on a level surface in the absence of friction, the mechanical energy gained is the translational kinetic energy of the car (neglecting the rotational kinetic energy of the wheels). What the car specifications give us, therefore, is the derivative of the WKET. Thus

$$P_c = \frac{d}{dt}\int_0^{x_t} F(x)\,dx = \frac{d}{dt}\left(\frac{1}{2}mv_t^2 + \frac{1}{2}mv_0^2\right) = mv\frac{dv_t}{dt} = mv_t a_t.$$

In this formula we made use of the fact that the derivative of a constant, $\frac{1}{2}mv_0^2$, is zero. We drop the t subscripts and write

$$P_c = mva.$$

P_c is the mechanical power the car's engine supplies to the wheels to give it an acceleration a when moving at velocity v.

PROBLEM 3.2 Horsepower of the Police Car • Assume the police car weighs 3500 lb. Its mass is then (3500/2.205) kg, where we make use of the fact that if $g = 9.8\ \text{ms}^{-2}$, then 1 kg weighs 2.205 lb. Modify the program written for Problem 3.1 to print (or plot) the horsepower of the police car as a function of time until the police car overtakes the speeder. The conversion from watts to horsepower is 1 hp = 746 watts.

3.3 Toward a More Realistic Model of an Accelerating Car

3.3.1 Motion with Limited Power

It should be obvious from the calculations required to solve Problem 3.2 that the assumption of constant acceleration for the police car or any other car is totally unrealistic because it demands an automobile engine with unlimited horsepower. The horsepower of the engines that power today's cars is seldom larger than 250 hp, much less than that demanded by the police car of Problems 3.1 and 3.2. In the next few paragraphs we will explore the motion of an automobile with limited power. Our focus is shifting away from the speeder versus police problem, and our goal for the remainder of the chapter is to construct a realistic model of an accelerating automobile.

We begin by examining the effects of limited power. Let P_c be the maximum power generated by a car's engine, and let this be a constant. We may solve the power formula given above for the acceleration to obtain

$$a = \frac{P_c}{mv}.$$

This equation requires that an increase in the velocity of a car be accompanied by a decrease in its acceleration. In short, the acceleration of a car is a function of its velocity, and the acceleration decreases as its velocity increases, a reality we have all experienced. It is a kind of "seat of your pants" feeling you get when you drive a small car and attempt to pass another car travelling at highway speeds. It also explains why it is difficult to spin your wheels at large speeds.

The fact that the acceleration is a function of the velocity introduces another problem; the kinematics equations for constant acceleration no longer apply. We find ourselves in the circle mentioned before; we want to find $v(t)$ and it is the integral of $a(t)$, but $a(t)$ is an unknown function of $v(t)$. Once again, we will bail ourselves out of this difficulty by using numerical analysis techniques to find a numerical solution. Specifically we will use the Feynman-Newton method outlined in Algorithm 3.1. We will simplify it by omitting the calculations for the speeder, and we will complicate it by taking into account the limited horsepower of the car's engine.

There is a problem related to the initial conditions that we must discuss before going further. In the case of a car accelerating from rest, $x_0 = v_0 = 0$, and that is the case we will study. What is a_0? If we use the formula $a = P_c/mv$ then at $t = 0$ the acceleration is undefined, and at very small velocities the acceleration is huge, a phenomenon we know is not true. We will resolve the problem this way. Given a coefficient of static friction μ (assume $\mu = 0.7$, approximately the coefficient of static friction of rubber on blacktop), we know that the acceleration of an ordinary car will be no larger than μg, or about 6.9 ms^{-2}. (Drag racers obtain accelerations larger than g, but they use special tires and techniques to get very large coefficients of friction.) Because v is small near $t = 0$, the equation $a = P_c/mv$ will yield an acceleration much larger than 6.9 ms^{-2}, a situation we have decided is impossible. Therefore we will assume the acceleration is the smaller of 6.9 or P_c/mv.

Algorithm 3.2 shows how to proceed to calculate both $s(t)$ and $v(t)$, with the assumption of constant power, using the Feynman-Newton method.

ALGORITHM 3.2 / The Motion of a Car with Limited Power

STEP 1. Assign a small value to h.
STEP 2. Choose the length of time for which we want $x(t)$ and $v(t)$ and call this t_{max}.
STEP 3. Assign the car an initial position, velocity, and acceleration, x_0, v_0, and a_0. An estimate for a_0 is 6.9 ms^{-2}.
STEP 4. Calculate $v_{\frac{1}{2}}$:

$$v_{\frac{1}{2}} = v_0 + \frac{h}{2} a_0.$$

STEP 5. Begin a loop. Loop while $t \le t_{max}$ (until $t < t_{max}$).

STEP 6. Tabulate or plot $x(t)$, $v(t)$, or $a(t)$.

STEP 7. Update x:

$$x \leftarrow x + hv.$$

STEP 8. Let a be the smaller of 6.9 ms^{-1} or

$$\frac{P_c}{mv},$$

where m is the mass of the car and P_c is its horsepower.

STEP 9. Update v:

$$v \leftarrow v + ha.$$

STEP 10. Increment t:

$$t \leftarrow t + h.$$

STEP 11. End of STEP 5 loop.

STEP 12. Quit.

PROBLEM 3.3 Motion of a Car with Constant Power • Write a program to implement Algorithm 3.2. Choose $h = 0.1$, $P_c = 200$ hp (1 hp = 746 watts), and assume a weight of 3170 lb (a 1 lb weight has a mass of 2.205 kg). Assume $x_0 = v_0 = 0$ and $a_0 = 6.9$ ms^{-2}; use $dom = [0, 25]$ s and $ran = [0, 250]$ $kmhr^{-1}$

PROBLEM 3.4 Experimental Data • In order to deal with a concrete example, we obtained some parameters and other data from tests on a Toyota MR2 Turbo from *Road & Track* magazine.[2] These data include three $x(t)$ points as follows: (2.9 s, 100 ft), (7.9 s, 500 ft), and (13.7 s, 1320 ft), and 9 $v(t)$ points as follows: (1.9 s, 30 mph), (3.3 s, 40 mph), (3.6 s, 50 mph), (6.2 s, 60 mph), (8.3 s, 70 mph), (10.3 s, 80 mph,), (13.2 s, 90 mph), (16.7 s, 100 mph), (13.7 s, 94 mph). Modify your program to convert this data to SI units and superimpose them on your $x(t)$ and $v(t)$ curves.

The results of our calculations are shown in Figure 3.1, where the disagreement between theory and experiment is obvious and significant. Of course, we should have expected this since we did not take into account either rolling friction or drag. Our results obtained from Problem 3.3 allow the velocity of the accelerating car to increase without limit, it just takes much longer when the power rather than the acceleration is constant. Furthermore, the actual car has a terminal speed of 225 $kmhr^{-1}$ (estimated by the test drivers), and our model has no limiting speed.

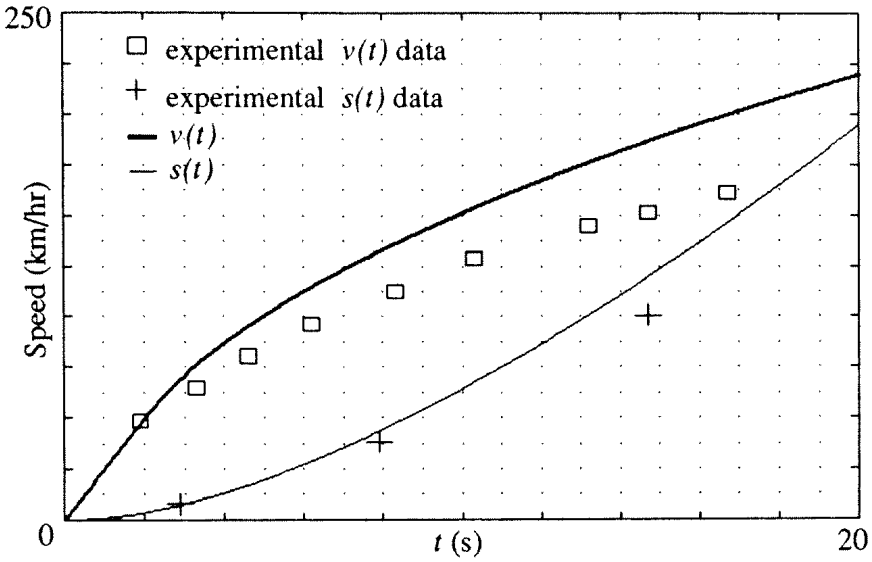


FIGURE 3.1 • $V(t)$ and $x(t)$ for Constant Power. Data printed with permission of *Road & Track.*

3.3.2 The Accelerating Car with Air Resistance and Rolling Friction (Challenging)

The derivation of the equation for P_c assumed that the power of the car changes only its kinetic energy. We know this is not true. Some energy appears as a change in the temperature of the car's surface and the air surrounding it. The necessity for special ceramic heat-resistant tiles on the surface of the space shuttle should be sufficient evidence that at high speeds there is considerable heat. The beautiful trail of a meteoroid gives similar evidence. Additional energy losses of the car are a result of rolling friction.[3] This energy appears as a change in temperature of the tires and the road surface. In summary, to analyze the motion of an accelerating car we must take into account both the limited power of the car and its thermal energy losses. The goal of the next few paragraphs is to show that when these factors are taken into account, the acceleration of the car is

$$a = \frac{1}{mv}\left(P_c - \frac{1}{2}CA\rho v^2 - \tau mgv\right),$$

where C is the drag coefficient of the car and τ is its tractive resistance. These calculations are not simple, and you may wish to simply skim over them.

Let us identify the horizontal forces on the car as F_r, F_a, and F_t. F_r is the horizontal forward force of the road on the car, F_a is the air resistance force, and F_t is the tractive force (rolling friction). The latter two forces are opposite to the velocity. The WKET becomes

$$\int_0^{x_t} (F_r - F_a - F_t)\,dx = \frac{1}{2}mv_t^2,$$

where we have assumed the initial velocity, hence the initial kinetic energy, was zero. Once again, what the manufacturer specifies is P_c where

$$P_c = \frac{d}{dt}\int F_r\, dx.$$

Taking the derivative of the WKET and substituting in P_c, we obtain

$$P_c - \int_0^{x_t} F_a\, dx - \int_0^{x_t} F_t\, dx = \frac{d}{dt}\left(\frac{1}{2}mv_t^2\right) = mva.$$

Experiments show that air resistance increases with the speed of the car. In a small, low-power car you can experience this as the car reaches a limiting speed, even on level ground, at which point all of the car's power is going into thermal energy. Air resistance experiments also show that the force of air friction is proportional to the square of the speed of the car, at least at large enough speeds. Once again, we use the following equation for air resistance:

$$F_a = \frac{1}{2}CA\rho v^2,$$

where C is a drag coefficient, A is the cross-sectional area of the car (seen head on), ρ is the density of air, and v is the speed of the car. C depends in part on how well the car is streamlined.

We must also take rolling friction into account. If you have ever pushed a stalled car on a level surface you will have experienced resistance to rolling. We adopt the results given in Sears;[4] the force of rolling friction is the tractive resistance, $\tau = 0.015$, times the weight. Thus

$$F_r = \tau mg,$$

with $\tau = 0.015$. Putting all of our algebraic work into the derivative of the WKET gives

$$P_c - \tfrac{1}{2}CA\rho\frac{d}{dt}\int_0^{x_t} v^2\, dx - \tau mg\frac{d}{dt}\int_0^{x_t} dx = mva.$$

Notice that we factored all constants out of the derivatives and the integrals. Now we proceed to take the two derivatives on the left-hand side:

$$\frac{1}{2}CA\rho\frac{d}{dt}\int_0^{x_t} v^2\, dx = \frac{1}{2}CA\rho v^2\frac{dx}{dt} = \frac{1}{2}CA\rho v^3,$$

and

$$\tau mg\frac{d}{dt}\int_0^{x_t} dx = \tau mgv.$$

Substitution back into the derivative of the WKET gives

$$P_c - \frac{1}{2}CA\rho v^3 - \tau mgv = mga.$$

Finally, we finish this task by solving for a explicitly. Thus

$$a = \frac{1}{mv}(P_c - \frac{1}{2}CA\rho v^2 - \tau mgv).$$

We will take the acceleration to be the smaller of $a_0 = 6.9\ \mathrm{ms}^{-2}$ or the acceleration calculated in the previous formula.

Perhaps we should finish by providing a simple conceptual framework for what is happening as the car accelerates. With the improbable situation that the car is accelerating without drag and rolling friction, the mechanical power, P_c, of the car, when integrated, appears as a change in the car's kinetic energy. In the real world some of this power is wasted, and, when integrated, shows up as an increase in the internal thermal energy of the car, tires, and surrounding air.

PROBLEM 3.5 A More Realistic Solution • Complicate the program you wrote for Problem 3.3 to include air resistance and rolling friction. Assume $C = 0.5$, $A = 2.5\ \mathrm{m}^2$, $\rho = 1.2\ \mathrm{kgm}^{-3}$, $\tau = 0.015$, $P_c = (190)(746)$ watts, and $m = (3170/2.205)$ kg. The 3170 lb is made up of a 2970 lb car and a 200 lb driver. Do your results agree with the data from *Road & Track*?

When you solve Problem 3.5, you will be disappointed with the results. The theoretical curves seem to have the right shape, but they do not go through the experimental points. What could be wrong? We could increase the drag coefficient, but it is typically about 0.5, and the MR2 is streamlined quite well.[5] Our assumption that the horsepower of the car is a constant 200 hp is probably more questionable than our choice of a drag coefficient. We know the horsepower is not really constant, it depends on the angular velocity (rpms) of the crankshaft. While accelerating, the angular velocity of the car's engine is certainly not constant. Thus the average horsepower is somewhat lower than the maximum horsepower, and it is not constant in any particular gear while the car is accelerating.

3.3.3 A Horsepower Study

PROBLEM 3.6 The Effect of Horsepower on the $x(t)$ and $v(t)$Curves • Run the program you wrote in Problem 3.4 and 3.5 for a number of different horsepower values. Keep the drag coefficient $C = 0.5$, or modify your program to implement the modification of Algorithm 3.2 listed in Algorithm 3.3. What do you think is the best choice of horsepower? What part of the $x(t)$ and $v(t)$ curves are most sensitive to the horsepower?

ALGORITHM 3.3 / Constructing a Family of Horsepower Curves

STEP 1. Assign a small value to h, and choose the initial conditions: $x_0 = 0$, $v_0 = 0$, $a_0 = 6.9$ ms^{-2}.
STEP 2. Determine the length of time for which we want $x(t)$ and $v(t)$ and call this t_{max}.
STEP 3. Loop through STEPS 4 through 9 starting with 140 horsepower and incrementing it by 40 horsepower until the horsepower is 220.
STEP 4. Find the half-step *speed*:

$$v_{\frac{1}{2}} = v_0 + \frac{h}{2} a_0 .$$

STEP 5. Loop while $t \le t_{max}$.
STEP 6. Tabulate or plot $x(t)$, $v(t)$, or $a(t)$.
STEP 7. Increment the position:

$$x \leftarrow x + hv.$$

STEP 8. LET a be the smaller of

$$a = \frac{1}{mv}(P_c - \frac{1}{2}CA\rho v^2 - \tau mgv),$$

or 6.9 ms^{-1}, where m is the mass of the car and P_c is its limiting power.
STEP 9. Update v:

$$v \leftarrow v + ha.$$

STEP 10. Increment t:

$$t \leftarrow t + h.$$

STEP 11. End of STEP 5 loop.
STEP 12. End of STEP 3 loop.
STEP 13. Quit.

Our results are show in Figure 3.2. Studying the figure, you will see that 160 hp is a good guess for the average horsepower of the Toyota MR2, but the agreement between the theoretical curves and the experimental data is still disappointing, especially since the rated horsepower of the car is 200 hp.

3.3.4 A Drag Coefficient Study

PROBLEM 3.7 Determining the Drag Coefficient • There is one other parameter in our program that is not known very well, the drag coefficient C, which we assumed to be 0.5. Once again, let us modify our program, this time to vary the drag coefficient C rather than the

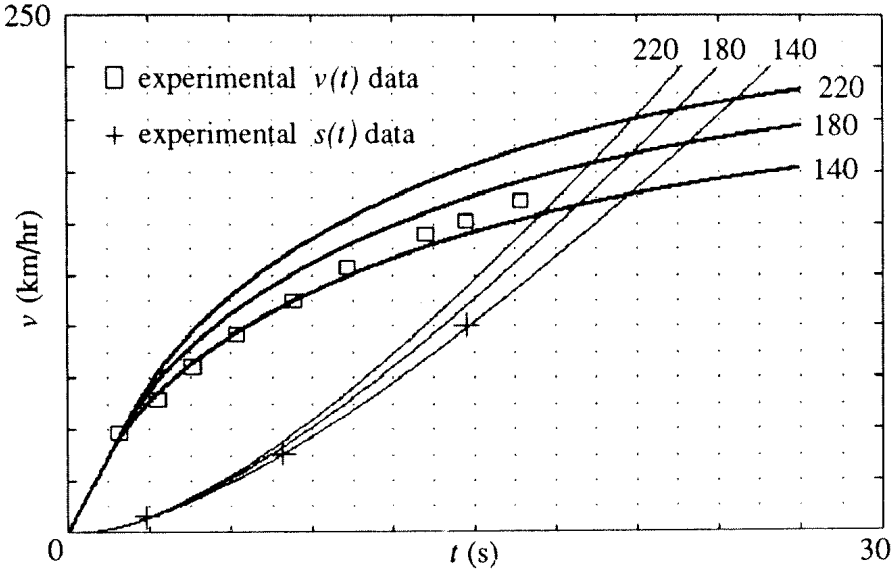


FIGURE 3.2 • A Family of $v(t)$ and $x(t)$ Curves for Several Values of the Horsepower. The car is the Toyota MR2 Turbo. Parameters: $W = 3170$ lbs, $C = 0.5$, $\tau = 0.015$, $dom = [0, 30]$ s, and $ran = [0, 250]$ kmhr^{-1}.

horsepower. In Algorithm 3.3 replace the "horsepower" loop with a "drag coefficient" loop. What do you conclude is the best estimate for the drag coefficient? What part of the $v(t)$ and $x(t)$ curves are most sensitive to the value of the drag coefficient? An experimental determination of the drag coefficient of a car has been done by Kwasnoski and Murphy.[6]

The agreement between theory and experiment is still not good. We must search for other parameters of our model that will bring it into closer agreement with the experimental data. There are at least three inadequacies in the theory. First, the driver shifts gears and we have ignored this. Second, the horsepower is not constant in any specific gear. It increases as the speed increases. Indeed, at the low rpms that a car has when it starts moving, the power is small. The purpose of "revving" the engine is to gain more power from the engine when the car starts from rest. Do you think it is better to spin the wheels to get more power but a smaller coefficient of friction, or is it better to start at low rpms and not slip? The final inadequacy in the theory is our neglect of the power that goes into the rotational kinetic energy of the wheels. Do you think this is a significant factor?

3.3.5 Shifting Gears

Cars have transmissions and gears to keep the rpms high at low speeds. During the gear shift time itself, the power supplied to the wheel is zero and the acceleration of the car is negative. The engine slows down and the speed drops. We knew we should try to take this effect into account, but it was not obvious how to do this because the published data do not include information about when and how long the shifting periods are. A communication with the *Road & Track* editors informed us that the test driver shifts gears approximately when the maximum rpms are reached, a fact that allowed us to find a good approximation to the times

at which the car was shifted to higher gears because the speeds at maximum rpms are published. There is, apparently, a maximum in the power versus rpm curve, and the test driver goes through the maximum before shifting to a higher gear, keeping the power as high as possible.

We also learned from the editors that typical shift times are 0.5 s, except the one from second gear to third gear, which takes 0.6 s or 0.7 s. In our improved model of an accelerating car we will assume that the acceleration equation has $P_c = 0$ during the gear shift process, forcing the car to slow. We will also assume the power is a constant, somewhat less than the published power. As before, we adopt a value of 6.9 ms^{-2} for the initial acceleration, a value that is consistent with a coefficient of static friction of 0.7. Our results were not sensitive to this value because the available horsepower at low rpms limits the acceleration more severely. Finally, we learned that in the tradition of drag racing, the distance and time measurements do not begin until the car has moved a distance of one foot. Using $a_0 = 6.9\ ms^{-2}$, we find these initial conditions: at $t = 0$, $x = 0$ but $v = 1.5\ ms^{-1}$. Algorithm 3.4 takes into account the time it takes to shift gears, during which the acceleration is negative.

ALGORITHM 3.4 / The Motion of a Car with Limited Power and Gears

STEP 1. Assign a small value to h, and choose the initial conditions, $x_0 = 0$ and $v_0 = 1.5\ ms^{-1}$.
STEP 2. Determine the length of time for which we want $x(t)$ and $v(t)$ and call this t_{max}.
STEP 3. Calculate the half-step velocity:

$$v_{\frac{1}{2}} = v_0 + \frac{h}{2} a_0 .$$

STEP 4. Loop until $t \geq t_{max}$.
STEP 5. Tabulate or plot the current $x(t)$, $v(t)$, or $a(t)$.
STEP 6. Update x:

$$x \leftarrow x + hv.$$

STEP 7. Let a be determined in this way:

- When v reaches 35, mph let $P_c = 0$ for 0.5 s. Shift from first to second.
- When v reaches 60, mph let $P_c = 0$.for 0.6 s. Shift from second to third.
- When v reaches 91, mph let $P_c = 0$ for 0.5 s. Shift from third to fourth.
- When v reaches 125, mph let $P_c = 0$ for 0.5 s. Shift from fourth to fifth.
- While $P_c = 0$,

$$a = \frac{1}{mv}\left(-\frac{1}{2} CA\rho v^2 - \tau mgv\right).$$

- Otherwise

$$a = \frac{1}{mv}\left[P_c - \frac{1}{2}CA\rho v^2 - \tau mgv\right].$$

where P_c = 190(746) watts.

- Let a = minimum of 6.9 ms^{-2} or a as calculated above.

STEP 8. Update v:

$$v \leftarrow v + ha.$$

STEP 9. Increment t:

$$t \leftarrow t + h.$$

STEP 10. End of STEP 4 loop.
STEP 11. Quit.

PROBLEM 3.8 Shifting Gears • Modify the accelerating car program to include corrections for shifting gears; in other words, implement Algorithm 3.3. Describe your results.

3.3.6 One Last Refinement

The results are still disappointing, and at this point it would be easy to give up trying to model an accelerating car with a computer program. But we have mentioned but not included an important consideration: The power of the car varies while it is in a particular gear; and the power increases with the angular velocity (rpms) of the engine. In fact, the reason for having gears is simply to maintain high rpms and keep the rate at which mechanical energy is delivered to the wheels as high as possible. To a first approximation, the horsepower varies *linearly* with the angular velocity (rpms) of the crankshaft, which means that in a specific gear the horsepower varies linearly with the speed of the car. Shifting gears merely changes the slope of the straight-line horsepower versus velocity relationship.

For example, according to the information from *Road & Track*, the maximum speed in first gear is 35 mph. Its horsepower will vary, therefore, from zero to approximately 200 hp according to the following rule:

$$P_c = \frac{v}{35}200$$

watts, and we have taken great liberties with units to make a point. In the computer program, 35 mph must be converted to ms^{-1}, and 200 hp is converted to watts. The point is that the horsepower is not constant, and a better approximation to the truth is that it varies linearly with the speed of the engine.

According to the *Road & Track* editors, the horsepower curve (hp versus rpm) reaches a point where increasing rpms produce *less* horsepower. The drivers shift when the angular velocity of the car's engine is just past the maximum horsepower, or approximately at the

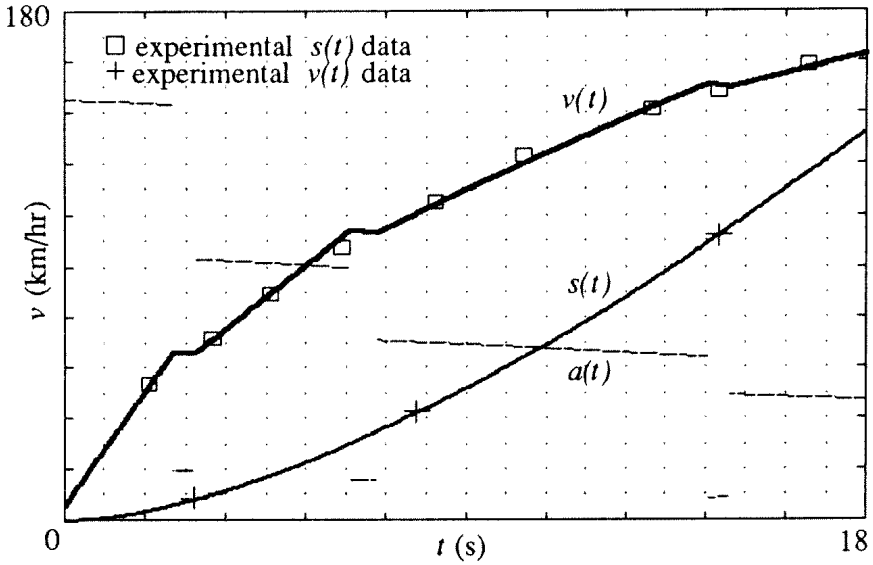


FIGURE 3.3 • The "Best Fit" of the Computer Model to the Experimental Data for the Toyota MR2 Turbo. The $x(t)$ curve is scaled to fit on the $v(t)$ graph, and the $a(t)$ curve is shifted upward and scaled to fit on the $v(t)$ graph. The model assumes P_c = 195 hp, C = 0.35, t = 0.015, v_0 = 1.5 s^{-1}, a_0 = 6.9 ms^{-1}, *dom* = [0, 18] s, *ran* = [0, 180] $kmhr^{-1}$.

maximum speed in each gear. We constructed a new computer model by incorporating the variation in power. The results are shown in Figure 3.3.

To have the appropriate scale and to keep the acceleration on the $v(t)$ graph, we actually graphed $20[a(t) + 1.1]$. The acceleration actually becomes negative during the shifting gears phases. With some thought, you should be able to explain the shape of the $a(t)$ curve. By running the program several times, we found that 195 hp and a drag coefficient of 0.35 gave a good fit of our model to the data. The agreement between our theory and the experimental data is quite good. We could make it even better if we made minor adjustments in two shifting speeds, but we chose to let the model stand.

Our computer model of the accelerating automobile is outlined in Algorithm 3.5. This is the most complicated algorithm we have listed so far. Clearly there are many complicating factors that arise in modeling an accelerating car, and we have not reached the limit of complexity. We choose to leave the problem at this point because we feel that we are as close to the truth as can reasonably be expected. Perhaps we gave ourselves too much freedom and we may have overworked the data. In any case, our model of the accelerating car is consistent with the experimental data. To proceed further, we need additional and more precise data. Specifically we need better times and durations of the shifting-gears events, a better estimate of the tractive resistance, more $x(t)$ and $v(t)$ points, especially for large $x(t)$, and even the temperature, atmospheric pressure, and the relative humidity. The last three quantities are needed to know the density of air, which can vary significantly. It is very likely that we could find other values of the horsepower and the drag coefficient, which would give us an equally good fit of our theory and the published experimental data. At this point we have an excellent

understanding of the factors that influence the motion of an accelerating car, but we can make our model more precise only with more data.

With our theory, we can predict the terminal velocity and see if it agrees with the terminal speed estimated by the test drivers; see Problem 3.9.

ALGORITHM 3.5 / A Computer Model of an Accelerating Automobile

STEP 1. Choose h, and assign the car these initial conditions: $x_0 = 0$, $v_0 = 1.5\ \text{ms}^{-1}$, $a_0 = 6.9\ \text{ms}^{-2}$.
STEP 2. Determine the length of time for which we want $x(t)$ and $v(t)$ and call this t_{max}.
STEP 3. Choose the tractive resistance force:

$$F_r = 0.015\, mg.$$

STEP 4. Choose $C = 0.35$, $A = (66.9 \times 48.8\ \text{in}^2)(6.45 \times 10^{-4}\ \text{m}^2\text{in}^{-2})$ and $\rho = 1.2\ \text{kgm}^{-3}$, with

$$F_a = \frac{1}{2} CA\rho v^2.$$

STEP 5. Calculate the half-step velocity,

$$v_{\frac{1}{2}} = v_0 + \frac{h}{2} a_0.$$

STEP 6. Use flags to signal gear shifts. Clear flags #1, #2, #3, and #4 (FLG = 0).
STEP 7. Loop until $t \geq t_{max}$.
STEP 8. Tabulate or plot the current $x(t)$, $v(t)$, or $a(t)$.
STEP 9. Update x:

$$x \leftarrow x + hv.$$

STEP 10. Let a be determined in the following way:

- When v reaches 35 mph, set flag #1 (FLG1 = 1).
- When v reaches 60 mph, set flag #2 (FLG2 = 1).
- When v reaches 91 mph, set flag #3 (FLG3 = 1).
- When v reaches 125 mph, set flag #4 (FLG4 = 1).
- If FLG 1 = 0 (in first gear), then P_c is given by the expression

$$P_c = \frac{v}{(35)(0.447)}(195)(746).$$

- If FLG 1 = 1 and FLG 2 = 0 (shift to second gear), then starting at zero, increment t_f by h:

$$t_f \leftarrow t_f + h.$$

If $t_f \geq 0.5$, then

$$P_c = \frac{v}{(60)(0.447)}(195)(746).$$

Otherwise $P_c = 0$. (This makes P_c zero for 0.5 s for shift from first to second.)

- If FLG 2 = 1 and FLG 3 = 0 (shift to third gear), then starting at zero, increment t_s by h:

$$t_s \leftarrow t_s + h.$$

If $t_s \geq 0.6$, then

$$P_c = \frac{v}{(91)(0.447)}(195)(746).$$

Otherwise $P_c = 0$. (This makes P_c zero for 0.6 s for shift from second to third.)

- If FLG 3 = 1 and FLG 4 = 0 (shift to fourth gear), then starting at zero, increment t_t by h:

$$t_t \leftarrow t_t + h.$$

If $t_t \geq 0.5$, then

$$P_c = \frac{v}{(125)(0.447)}(195)(746).$$

Otherwise $P_c = 0$. (This makes P_c zero for 0.5 s shift from third to fourth.)

- If FLG 4 = 1, then starting at zero, increment t_i by h:

$$t_i \leftarrow t_i + h.$$

If $t_i \geq 0.6$, then

$$P_c = \frac{v}{(145)(0.447)}(195)(746).$$

Otherwise $P_c = 0$. (This makes P_c zero for 0.6 s shift from fourth to fifth.)

- Set a = minimum of 6.9 ms^{-2} or

$$\frac{1}{mv}\left[P_c - \frac{1}{2}CA\rho v^2 - \tau mgv\right].$$

STEP 11. Update v:

$$v \leftarrow v + ha.$$

STEP 12. End of STEP 7 loop.

STEP 13. Quit.

PROBLEM 3.9 Terminal Velocity of the Automobile • Make the program run for longer time intervals, say 150 seconds, and compare the terminal speed that our model predicts with that estimated by the test drivers, 145 mph.

PROBLEM 3.10 The Density of Air and the Drag Coefficient • Is the density of air sufficiently constant so that $\rho = 1.2\ \text{kgm}^{-3}$ is a good assumption? By how much does the density of air change from early in the morning when the temperature might be 40° F to the afternoon when it might be 75° F? The *Handbook of Physics and Chemistry* is useful to answer these questions. Can our computer model distinguish between (large C, small ρ) and (small C, large ρ)?

PROBLEM 3.11 $X(t)$ and $V(t)$ Curves for Other Cars • Obtain a *Road & Track* magazine and perform a study similiar to ours on one of the cars they describe. If you cannot obtain a *Road & Track*, here is some pertinent data[7]: car = BMW 535i; weight = 3800 lb; width = 68.9 in; height = 55.6 in; stopping distances using ABS (automatic braking system): 60 mph to 0 in 129 ft, 80 mph to 0 in 228 ft (check these out, to see if they are possible). Acceleration data, time to speed: 0 - 30 mph, 2.7 s; 0 - 40 mph, 3.3 s; 0 - 50 mph, 5.8 s; 0 - 60 mph, 7.7 s; 0 - 70 mph, 10.2 s; 0 - 80 mph, 12.8 s; 0 - 90 mph, 15.7 s; 0 - 100 mph, 19.5 s. Acceleration data, time to distance: 0 - 100 ft, 3.3 s; 0 - 500 ft, 8.7 s; 0 - 1320 ft; 15.9 s at 91 mph. Top speed (estimated) 143 mph. Transmission is 5-speed manual with these top speeds for first, second, third, fourth, and fifth gears are 34, 59, 95, 132, 143 mph, respectively. The gear shifts take place at the top speeds for each gear.

3.4 Additional Problems

PROBLEM 3.12 How Good Are the Results? • Generate a set of $x(t)$ and $v(t)$ curves for various values of the horsepower and generate another set of $x(t)$ and $v(t)$ curves for various values of the drag coefficient. Print these curves and then superimpose them using a window or a light box. Which part of the curves is affected most by the drag coefficient? Which part of the curves is affected most by the horsepower? Is any solution unique? In other words, could we find a certain low C and low P_c that would fit the data as closely as another high C and high P_c?

PROBLEM 3.13 Car Overtaking a Train • A car travelling at $100\ \text{kmhr}^{-1}$ overtakes the rear of a train travelling at $60\ \text{kmhr}^{-1}$ parallel to the highway. The train is 1 km long. How long will it take the car to reach the front of the train? You can do this problem with algebra, but it is informative to solve it with a computer. Begin by writing an algorithm that describes the problem. Make a sketch to help you in constructing the algorithm.

PROBLEM 3.14 A Tale of Two Trucks • Two trucks start from the same point and travel the same distance. Truck A accelerates with an acceleration a for first half of the distance and then at $2a$. Truck B accelerates at $2a$ for the first half of the distance, then accelerates with a for the second half. Which will win? Model this problem with a computer program.

3.5 References

[1] R. T. Weidner, *Physics*, rev. ed. (Boston: Allyn and Bacon, 1989), 45.
[2] *Road & Track* "1991 Toyota MR2 Turbo,"(April 1990): 81.
[3] F. W. Sears, I. Zemansky, and M. W. Young, *University Physics* (Reading, MA: Addison-Wesley, 1987), 171.
[4] ibid.
[5] ibid.
[6] J. Kwasnoski and R. Murphy, "Determining the Aerodynamic Drag Coefficient of an Automobile," *Am. J. Phys.* 53 (8) (August 1985): 776.
[7] *Road & Track* "The BMW 535i," (January 1989): 125.

3.6 Programs

PROGRAM 3.6.1 / Police versus Speeder

```
! Find x(t) and v(t) for a car accelerating from rest with constant acceleration.
! Algorithm 3.1. Graphical solution to Problem 3.1.
! Try domain [0, 10] s, range [0, 300] m.
LIBRARY "ComPhys"                                          ! Graphing routines.
CALL OPEN (xmin,xmax,ymin,ymax,wxmin,wxmax,wymin,wymax)    ! In ComPHys.
LET h = 0.03                                               ! Choose a small value for h.
LET t = 0                                                  ! Determine the initial
                                                             conditions.
LET x = 0                                                  ! Police car starts at x = 0
LET a = 6                                                  ! with a constant acceleration
                                                             of 6 m/s/s.
LET v = v + a*h/2                                          ! Calculate v(h/2).
LET xs = 0                                                 ! Initial x for speeder.
LET as = 0                                                 ! Speeder does not accelerate.
LET vs = 100*1000/3600                                     ! Convert to m/s.
DO                                                         ! Start the main loop to create
                                                             the s versus t graph.
   PLOT t, x                                               ! Plot position of police car.
   PLOT t, xs                                              ! Plot position of speeder.
   PLOT t, v*3600/1000                                     ! Convert v to km/hr.
                                                           ! Plot police car's speed on the
                                                             x versus t graph.
   CALL FEYNMAN_NEWTON (x,v,a,h)                           ! Calculate new x & v.
   CALL FEYNMAN_NEWTON (xs,vs,as,h)                        ! Calculate new xs.
   LET t = t + h                                           ! Update the time.
LOOP until x >=xs                                          ! Stop calculations.
LET hp = 3000/2.205*v*a/746                                ! Calculate final horsepower.
PRINT t - h;" s ", v*3600/1000;" km/hr ",hp; " horsepower." ! Print info.
CALL LABEL (wxmin, wxmax, wymin, wymax)                    ! Subroutine to put information
                                                             on
END                                                        ! graph. In ComPhys.
!.............
SUB FEYNMAN_NEWTON (xx,vv,aa,h)                            ! Subroutine to solve problem.
   LET xx = xx + vv*h                                      ! Calculate the new position.
   LET vv = vv + aa*h                                      ! Increment small velocity.
END SUB
```

PROGRAM 3.6.2 • Modelling an Accelerating Car

```
! Algorithm 3.2 with C = 0 and TR = 0, and Problem 3.5.
! Use domain [0, 20] s, range = [0, 250] km/hr.
LIBRARY "ComPhys"
!................
DEF a                                                   ! Definition of acceleration.
    LET lima = (HP - TR*m*9.81*v - 0.5*C*AR*rho*v^3)/(m*v)  ! Take smaller of lima or
    LET a = min(6.9, lima)                              ! 6.9 m/s/s.
END DEF
!................
CALL GET_INFO (h, m, HP, AR, rho, C, TR)                ! Fetch the parameters of the
                                                          problem.
CALL OPEN (xmin,xmax,ymin,ymax,wxmin,wxmax,wymin,wymax) ! Open a graphics window.
LET t = 0                                               ! The next few statements
                                                          determine the
LET ao = 6.9                                            ! initial values or boundary
                                                          conditions
LET v = v + ao*h/2                                      ! for displacement, velocity,
                                                          and acceleration.
DO                                                      ! Start the main loop to create
                                                          the s versus t graph.
   PLOT t, v*3600/1000                                  ! Plot speed of car in km/hr.
   PLOT t, x/4                                          ! Plot car's position scaled to
                                                          fit on velocity curve.
   LET x = x + v*h                                      ! Calculate the new position
                                                          from v.
   LET v = v + a*h                                      ! Increment small velocity.
   LET t = t + h                                        ! Increment the time.
LOOP until t >=xmax                                     ! Stop calculations.
CALL EXP_INFO (wxmin, wxmax, wymin, wymax, keep$)       ! Subroutine to plot
                                                          experimental  v versus t
CALL LABEL (wxmin, wxmax, wymin, wymax)                 ! Subroutine to put information
                                                          on graph.
END                                                     ! and s versus t data.
!.............
SUB GET_INFO (h, m, HP, AR, rho, C,TR)                  ! Define the parameters of the
                                                          problem.
! Data for Toyota MR2 Turbo, Road & Track, April 1990, p. 81
    LET h = 0.04                                        ! Small time interval.
    LET m = 3170*4.448/9.80                             ! Car and driver weight.
    LET HP = 190*746                                    ! Convert horsepower to watts.
    LET AR = 66.9*48.8*6.452e-4                         ! Area of car in square meters.
    LET rho = 1.2                                       ! Density of air in kg/cubic
                                                          meter.
    LET C = 0                                           ! Drag coefficient.
    LET TR = 0                                          ! Tractive Resistance.
END SUB
!.............
SUB EXP_INFO (wxmin, wxmax, wymin, wymax, keep$)        ! Plot experimental data.
    FOR j = 1 to 9                                      ! Use loop to read 9 (t, v)
                                                          points.
        READ t,v                                        ! Reads pairs of points in the
                                                          DATA statement below.
```

```
        LET v = v*0.447*3600/1000                          ! Convert mph to kilometers per
                                                             hr.
        CALL PLOT_DATA_B (wxmin,wxmax,wymin,wymax,t,v)     ! Plot the point on v versus t
                                                             curve.
    NEXT j
    DATA 1.9,30,3.3,40, 4.6,50,6.2,60,8.3,70,10.3,80,13.2,90,16.7,100,14.7,94
    FOR j = 1 to 3                                         ! Read 3 experimental (t, s)
                                                             points.
        READ t,y                                           ! Reads coordinate pairs in the
                                                             DATA statement below.
        LET v = y*.305/4                                   ! Convert ft to m and scale y to
                                                             fit on v versus t graph.
        CALL Plot_Data_A (wxmin, wxmax,wymin, wymax, t,v)
    NEXT j
    DATA 2.9,100,7.9,500,14.7,1320     ! (t, s) data from Road and Track.
END SUB
```

PROGRAM 3.6.3 • Generating a Family of Horsepower Curves

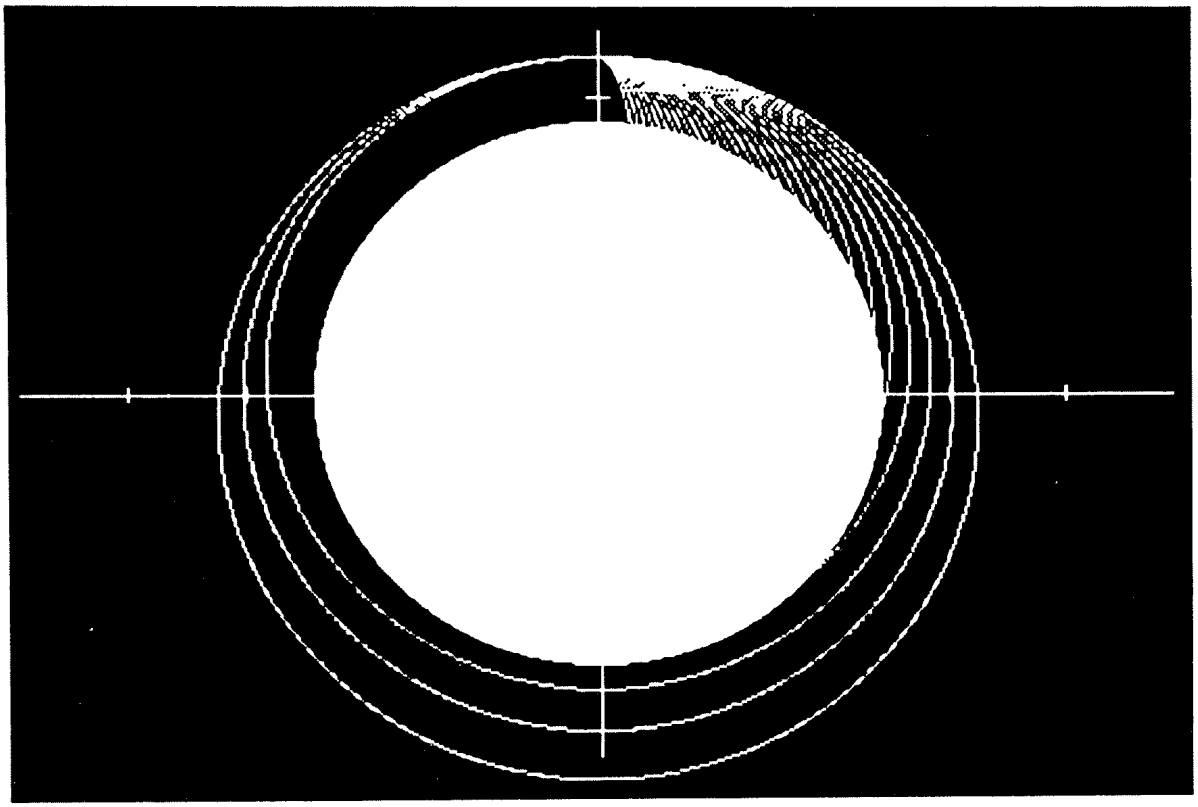
```
! Algorithm 3.3, Problem 3.7.
! Try dom = [0, 25} s ran = [0, 250] km/hr.
! Subroutines GET_INFO, and EXP_INFO are given in Program 3.6.2,
! and they are needed in this program.
!.................
DEF a
    LET lima = (PC - TR*m*9.81*v - 0.5*C*AR*rho*v^3)/(m*v)  ! Calculate a.
    LET a = min(6.9, lima)                                  ! Use smallest acceleration.
END DEF
!.................
LIBRARY "ComPhys"
CALL GET_INFO (h, m, PC, AR, rho, C, TR)                    ! Fetch the parameters of the
                                                              problem.
CALL OPEN (xmin,xmax,ymin,ymax,wxmin,wxmax,wymin,wymax)     ! Open a graphics window.
FOR i = 140 to 220 step 40
    LET PC = i*746
    LET x = 0
    LET t = 0                                               ! The next few statements
                                                              determine the
    LET ao = 6.9                                            ! initial values
    LET v = 0
    LET v = v + ao*h/2                                      ! for displacement, velocity,
                                                              and acceleration.
    DO                                                      ! Start the main loop to create
                                                              the s versus t graph.
       PLOT t, v*3600/1000                                  ! Plot speed of car in km/hr.
       PLOT t, x/4                                          ! Plot car's position scaled to
                                                              fit on velocity curve.
       LET x = x + v*h                                      ! Calculate the new position
                                                              from v.
```

```
      LET v = v + a*h                                    ! Increment velocity.
      LET t = t + h                                      ! Increment the time.
   LOOP until t >=xmax                                   ! Stop calculations.
NEXT i
CALL EXP_INFO (wxmin,wxmax,wymin,wymax, keep$)           ! Subroutine to plot
                                                           experimental  v versus t
CALL LABEL (wxmin, wxmax, wymin, wymax)                  ! and s versus t data.
END
! Subroutine to put information on graph.
! Subroutines GET_INFO, and EXP_INFO are given in Program 3.6.2,
! and they are needed in this program.
```

Program 3.6.4 • Model of an Accelerating Car

```
! Algorithm 3.5, Problems 3.9, 3.10, and 3.11.
! Domain [0, 18],range [0, 180]
! Subroutines GET_INFO and EXP_INFO must be included. Find them in Program 3.6.2.
LIBRARY "Comphys"
CALL GET_INFO (h, m, HP, AR, rho, C, TR)                 ! Fetch the parameters of the
                                                           problem.
CALL OPEN (xmin,xmax,ymin,ymax,wxmin,wxmax,wymin,wymax)  ! Open a graphics window.
LET t = 0                                                ! Begin.
LET a = 6.9                                              ! To begin acceleration is 6.9.
LET v = 1.5+a*h/2                                        ! Calculate the half-step
                                                           velocity. Note initial
                                                           v = 1.5 m/s.
LET x = 0                                                ! Initial condition for x.
DO                                                       ! Start the main loop to create
                                                           the s versus t graph.
   PLOT t, v*3600/1000                                   ! Plot speed of car in km/hr.
   PLOT t, x/4                                           ! Plot car's position scaled to
                                                           fit on velocity curve.
   PLOT t, 20*(a + 1.1)
   LET x = x + v*h                                       ! Calculate the new position
                                                           from v.
   IF v >= 35*.477 then LET flg1 = 1                     ! 35 is top speed in first.
   IF v >= 60*.477 then LET flg2 = 1                     ! 60 is top speed in second.
   IF v >= 91*.477 then LET flg3 = 1                     ! 91 is top speed in third.
   IF v >= 125*.477 then LET flg4 = 1                    ! 125 is top speed in fourth.
   IF flg1 = 0 then LET Pc = (v/(35*0.447))*HP
   IF flg1 = 1 and flg2 = 0 then                         ! Shift from 1 to 2.
      LET tf = tf + h
      IF tf >= 0.5 then LET Pc =v/(60*.447)*HP else LET Pc = 0  ! No power during shift.
   END IF
   IF flg2 = 1 and flg3 = 0 then                         ! Shift from 2 to 3.
      LET ts = ts + h
      IF ts >= 0.6 then LET Pc = v/(91*.447)*HP else LET Pc = 0 ! No power during shift.
   END IF
   IF flg3 = 1 and flg4 = 0 then                         ! Shift from 3 to 4.
```

```
      LET tt = tt + h
      IF tt >= 0.5 then LET Pc = v/(125*.447)*hp else LET Pc = 0! No power during shift.
   END IF
   IF flg4 = 1 then                                          ! Shift to fifth.
      LET tfi = tfi + h
      IF tfi >= .5 then LET Pc = v/(145*.447)*hp else LET Pc = 0! No power during shift.
   END IF
   LET lima = (Pc- TR*m*9.81*v - 0.5*C*AR*rho*v^3)/(m*v)     ! Is power limited by
                                                               horsepower?
   LET a = min (lima,6.9)                                    ! Choose the smallest a.
   LET v = v + a*h                                           ! Increment v.
   LET t = t + h                                             ! Increment t.
LOOP until t >=xmax                                          ! Stop calculations.
CALL EXP_INFO (wxmin, wxmax, wymin, wymax, keep$)            ! Subroutine to plot
                                                               experimental  v versus t
CALL LABEL (wxmin, wxmax, wymin, wymax)                      ! Subroutine to put information
                                                               on graph.
END
! Subroutines GET_INFO and EXP_INFO must be included. Find them in Program 3.6.2.
```



Projectiles and Satellites

4 / Dynamics in Two Dimensions

4.1 Introduction

A review of the following topics may be helpful in this chapter:

- Newton's Law of Universal Gravitation
- Newton's Second Law of Motion
- Vectors, components, and vector operations
- Displacement, velocity, and acceleration

In Chapters 1 and 2, which we assume you have studied, we solved one-dimensional dynamics problems numerically, using Newton's Second Law to find an expression for the acceleration of an object and numerical integration techniques that we identified as either Euler's method or the Feynman-Newton method. In this chapter we will extend our work to two-dimensional problems, studying projectile motion with air resistance and orbital motion.

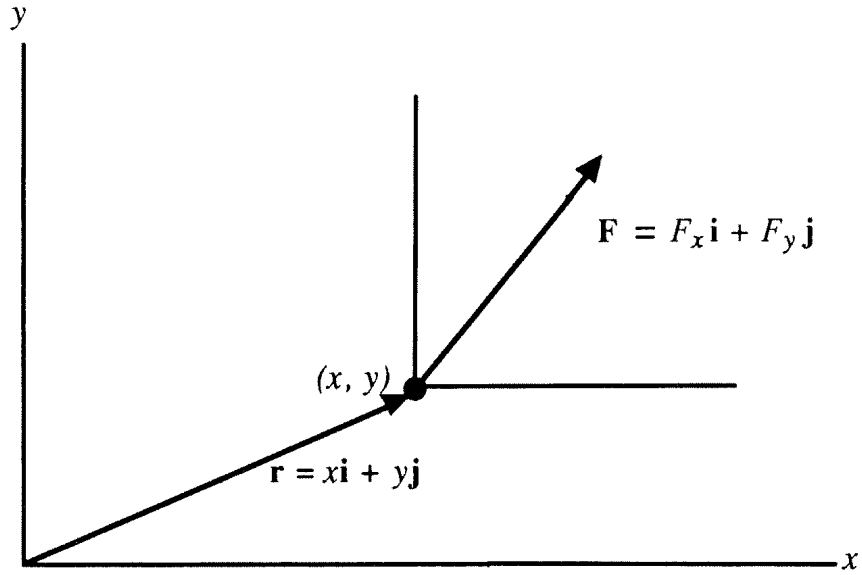


FIGURE 4.1 • The Force and Displacement Vectors in Two Dimensions.

Figure 4.1 shows the important quantities associated with motion in two dimensions. An object of mass m, with a displacement **r**, which may vary with time, is acted on by an external force **F**. The vector **r** has components x and y, the coordinates of the mass. The vector **F** has components F_x and F_y. We know that F_x produces the acceleration in the x-direction, a_x, and F_y produces the acceleration in the y-direction, a_y. We can treat the motion in each direction independently of the motion in the other. Thus

$$a_x = \frac{F_x}{m}$$

and

$$a_y = \frac{F_y}{m}.$$

In other words, we can solve the problem of motion in two dimensions using the same methods we introduced in Chapters 1 and 2, but we must have one set of equations for each coordinate. Our method of finding the numerical solution is outlined in Algorithm 4.1, which is the foundation for the solutions of all of the problems in this chapter.

ALGORITHM 4.1 / Numerical Solution of Two-Dimensional Dynamics Problems

STEP 1. Decide what step-size h will be used and the criteria for ending the calculations.
STEP 2. Determine the initial conditions: at $t = 0, x = x_0, y = y_0, v_x = v_0\cos(\theta)$, and $v_y = v_0\sin(\theta)$, where q is the angle between the initial velocity vector v_0 and the x-axis. Determine all the other parameters of the motion, drag coefficient, mass, temperature, etc.

STEP 3. From the geometry of the problem, determine the force components F_x and F_y.
STEP 4. Use Newton's Second Law to find the components of the acceleration:

$$a_x = \frac{F_x(\mathbf{r}, \mathbf{v}, t)}{m}$$

and

$$a_y = \frac{F_y(\mathbf{r}, \mathbf{v}, t)}{m},$$

where we have explicitly acknowledged the fact that **F** may be a function of the displacement **r**, the velocity **v**, and time t. For this step $t = 0$.
STEP 5. Find the half-step velocity components with these formulas:

$$v_{x\frac{1}{2}} = v_{x0} + \frac{h}{2} a_{x0}$$

and

$$v_{y\frac{1}{2}} = v_{y0} + \frac{h}{2} a_{x0}.$$

STEP 6. Begin a loop. Loop until some specific set of ending criteria are met, $t > t_{max}$ for example.
STEP 7. Tabulate or plot $x(t)$, $y(t)$, $v_x(t)$, etc.
STEP 8. Update the coordinates:

$$x \leftarrow x + hv_x$$

and

$$y \leftarrow y + hv_y.$$

STEP 9. Calculate a_x and a_y with Newton's Second Law.
STEP 10. Update the velocity components:

$$v_x \leftarrow v_x + ha_x$$

and

$$v_y \leftarrow v_y + ha_y.$$

STEP 11. Update the time:

$$t \leftarrow t + h.$$

STEP 12. End of Step 5 loop.
STEP 13. Quit

4.2 Projectile Motion

4.2.1 Projectile Motion with No Air Resistance

We will begin by using the Feynman-Newton method to solve projectile motion problems in two dimensions, with no air resistance. This problem has an analytical solution as well as a numerical solution, so we can compare the two results. A simple modification to the program will allow us to use the same program to take air resistance into account. Indeed, the procedure will handle many two-dimensional dynamics problems.

The physics of the problem is quite simple, the only force acting on the projectile being its weight. Thus

$$a_x = 0$$

and

$$a_y = -g,$$

which means we are ready to begin. Using these two equations in Algorithm 4.1 will give us the solution we desire.

PROBLEM 4.1 Modelling the Motion of a Projectile • Write a program based on Algorithm 4.1 and the equations just given to model the motion of a projectile. The output of the program should be a graph of height y versus distance x. The program should allow you to enter the initial conditions, including x_0, y_0, the magnitude of $\mathbf{v}_0$ and the angle θ that $\mathbf{v}_0$ makes with the x-axis. Begin with a step-size h = 0.01 s. For initial conditions let $x_0 = y_0 = 0$ and $|\mathbf{v}_0| = 25\ \mathrm{ms}^{-1}$. Also assume $\mathbf{v}_0$ makes an angle of 37° with respect to the x-axis. Graph the trajectory of the projectile on a coordinate system with domain [0, 70] m and range [0, 20] m.

PROBLEM 4.2 Finding a Suitable Step-Size *h* • Problem 4.1 also has an analytic solution that you can find in your textbook. Try different step sizes, 0.04, 0.1, 0.2, 0.5, and 1 s in the numerical solution and compare the results with the analytic solution. What is a good step size for this problem?

PROBLEM 4.3 Maximum Range • Show that the maximum range is achieved if θ is 45°. (You may wish to refer to Section 4.2.3.)

PROBLEM 4.4 Dropping Supplies • An airplane travelling at 200 kmhr^{-1} at an altitude of 150 m drops hay bales to cattle trapped in snow. Use the program you wrote in Problem 4.1 to determine the trajectory of a bale. When and where should the bales be released? You do not want to hit anything, especially not cows, with a bale of hay moving at more than 200 kmhr^{-1}. Do you need to know the mass of a hay bale for this problem?

4.2.2 Projectile Motion with Air Resistance

Air resistance significantly modifies the motion of a projectile. Just as we did in Chapter 1, we will assume the drag term has a magnitude given by

$$F_a = \frac{1}{2} CA\rho v^2,$$

where C is the drag coefficient, A is the cross-sectional area, ρ is the density of air, and v is the magnitude of the velocity. A choice of $C = 0.5$ gives sufficient precision for most purposes, at least until specific objects, such as baseballs, are considered.

We will determine the x and y components of the net force on the projectile. Take a clean sheet of paper, on it draw a simple x-y coordinate system with a large first quadrant, and place a dot in the first quadrant to represent the projectile. What are the forces acting on the projectile? Draw a force to represent the weight, mg. Draw a vector pointing toward the upper-right-hand corner to represent the velocity of the projectile. Assume this vector makes an angle ϕ with respect to the $+x$-axis. Draw the drag force $\mathbf{F}_a$ directly opposite to the velocity $\mathbf{v}$. Now you should be able to write

$$F_x = \underline{\hspace{3cm}}$$

and

$$F_y = -mg - \underline{\hspace{3cm}}$$

in terms of sines and cosines of ϕ.

How can we find ϕ? Refer to Figure 4.2. From the diagram it is evident that

$$\cos\phi = \frac{v_x}{v}$$

and

$$\sin\phi = \frac{v_y}{v},$$

where v is the magnitude of $\mathbf{v}$, that is

$$v = \sqrt{v_x^2 + v_y^2}.$$

It follows from Newton's Second Law that

$$a_x = -\frac{1}{2} C\rho \frac{A}{m} v^2 \cos\phi,$$

and

$$a_y = -g - \frac{1}{2} C\rho \frac{A}{m} v^2 \sin\phi.$$

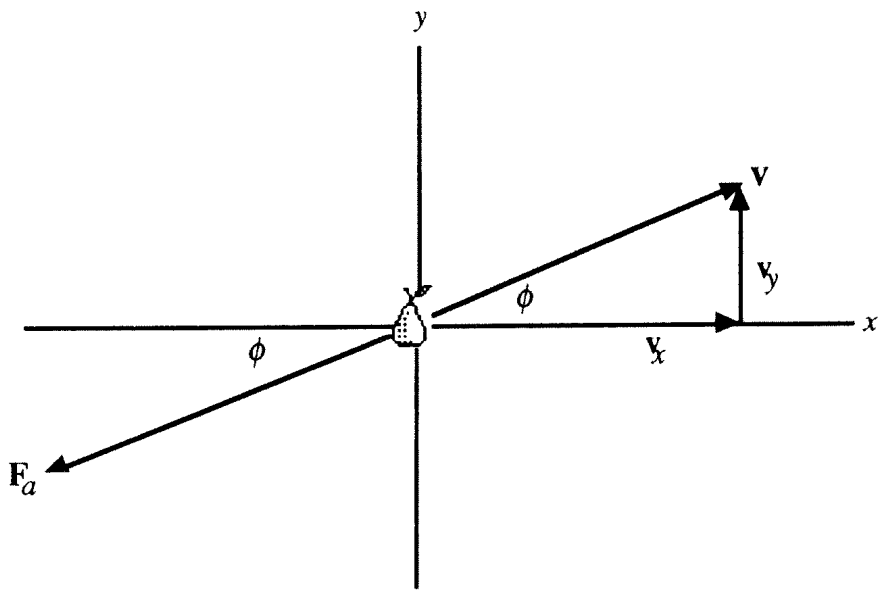


FIGURE 4.2 • The Relationship of $\mathbf{F}_a$ to $\mathbf{v}$, with a Pear as a Projectile.

Using simple algebra and trigonometry, we find that

$$a_x = -kvv_x,$$

and

$$a_y = -g - kvv_y,$$

where

$$k = \frac{1}{2} C\rho \frac{A}{m}.$$

These equations, when used in Algorithm 4.1, will model the flight of a projectile in a resistive medium and a constant gravitational field.

PROBLEM 4.5 Projectile Motion with Air Resistance • Modify the program in Problem 4.1 to include air resistance. Once again, the program should produce a graph of height y versus distance x for the projectile. Assume that the object is a pear with a diameter of 6 cm and a mass of 100 g and it is thrown with an initial velocity of 25 ms^{-1} at an angle of 37° with respect to the horizontal. Use *dom*= [0, 50] m, *ran* = [0, 15] m. Compare your results, maximum height and range, to the motion of the object without air resistance.

PROBLEM 4.6 On Throwing and Hitting Baseballs • Some pitchers achieve speeds close to 100 mph when they throw a baseball. Using $m = 0.145$ kg, $r = 0.0367$ m, and $C = 0.46$, how far can an outfielder throw a baseball if it is thrown at 100 mph at an angle of 40° with respect to the horizontal? 100 mph = 44.7 ms^{-1}. Use the program from Problem 4.5 with domain [0,100] m, range [0, 30] m, and $h = 0.01$ s.

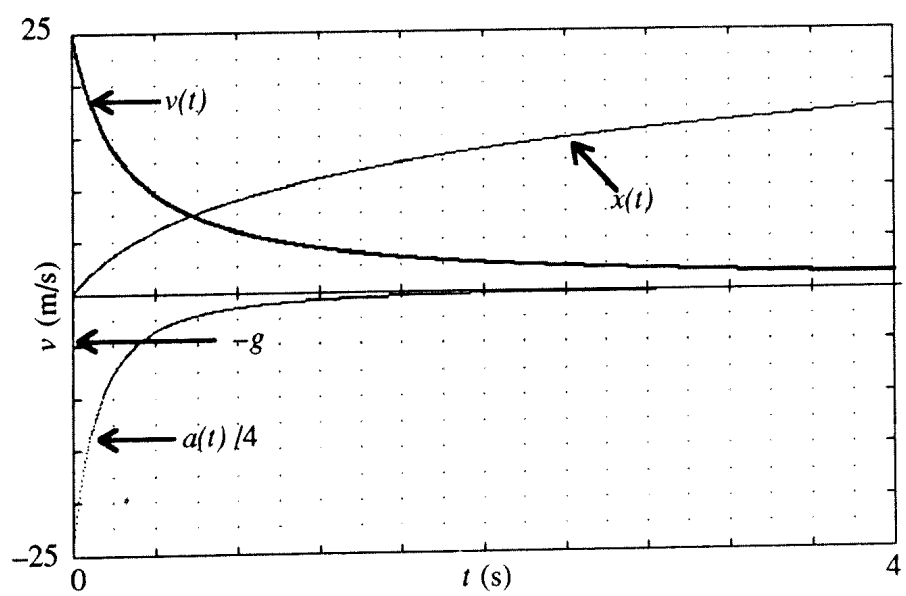
- Can a good outfielder make a throw to home plate from deep in left field?
- What variation in the distance would result from a change from afternoon temperatures to evening temperatures as a result of the change in the density of the air?
- How far would the ball go without air resistance? Is air resistance a significant part of the game of baseball?
- Does air resistance have an appreciable affect on the speed of a pitched ball as it travels to home plate?

Assume a hitter can give the ball an initial velocity of 50 ms^{-1} at an angle of 40°. Model the trajectory of the baseball.

- How far will the baseball travel?
- Watch the trajectory being plotted by your program. Does a baseball seem to hang in the air, especially a high fly? Is this a real effect or an illusion?
- Have the computer plot a new trajectory and use a stopwatch to measure the time to reach maximum height and the time it takes to go from maximum height to the ground. These times are the same with no air resistance. Are they the same with air resistance? If not, which is longer?
- Why do outfielders get *under* the ball?
- When should the most home runs be hit, afternoon or night? Why?

(It is interesting that the drag coefficient of the spheroid used in our national pastime is still somewhat uncertain and the parameters that determine its flight are still being investigated.[1,2,3])

FIGURE 4.3 • *X*-Component of the Displacement, Velocity, and Acceleration of a Ping-Pong Ball. *Dom* = [0, 4] s, *ran* = [–25, 25] ms^{-1}, $x_0 = y_0 = v_{y0} = a_y = 0$, $v_{x0} = 25$ ms^{-1}.



You can perform interesting experiments with the program we have developed; in Figure 4.3 we show the results of such an experiment. We were interested in the horizontal velocity and acceleration of a ping-pong ball. The projectile-motion program was modified to make $a_y = v_y = 0$, and the ping-pong ball was given an initial velocity of 25 ms^{-1}. The x position, velocity, and acceleration of the ping-pong ball were calculated and graphed. As expected, in the first second drag dramatically reduces the initial velocity from 25 to approximately 2.5 ms^{-1}, and the initial acceleration is several times g. What do you think is an approximate value for the practical maximum distance a ping-pong ball can be hit by a human being? Try both a computational experiment and a laboratory experiment.

4.2.3 Families of Projectiles

There are many instances where a family of graphs give more insights than a single graph. For example, we might want to make a graph of projectile motion for several different angles of projection, keeping all other parameters the same. In Problem 4.6, posed above, we might wish to make graphs of the motion of a baseball for various values of ρ, the density of air, to see how this parameter affects throwing and hitting. Simple modifications of Algorithm 4.1 will do the trick. The new algorithm is given below.

ALGORITHM 4.2 / Constructing a Family of Curves for Projectile Motion

STEP 1. Decide what step-size h will be used and the criteria for ending the calculations.
STEP 2. Choose the desired initial and final values of the parameter to be varied, and call these values p_i and p_f, respectively. A reasonable step size for p is dp, where $dp = (p_f - p_i)/10$.
STEP 3. Determine the initial conditions at $t = 0, x = x_0, y = y_0, v_x = v_0\cos(\theta)$, and $v_y = v_0\sin(\theta)$, where θ is the angle between the vector $\mathbf{v_0}$ and the x-axis.
STEP 4. Let $p = p_i$.
STEP 5. Begin a loop; loop until $p > p_f$ (while $p \le p_f$).
STEP 6. Let

$$a_x = -kvv_x$$

and

$$a_y = -g - kvv_y,$$

where

$$k = \frac{1}{2}C\rho\frac{A}{m}.$$

STEP 7. Find the half-interval velocity components with the formulas

$$v_{x\frac{1}{2}} = v_{x0} + \frac{h}{2}a_x$$

and

$$v_{y\frac{1}{2}} = v_{y0} + \frac{h}{2} a_y .$$

STEP 8. Begin a loop. Loop until the ending criteria are met, $y < 0$ for example.

STEP 9. Tabulate $x(t)$, $y(t)$, or plot the position (x, y).

STEP 10. Update the coordinates x and y:

$$x \leftarrow x + hv_x$$

and

$$y \leftarrow y + hv_y .$$

STEP 11. Calculate the accelerations a_x and a_y with the formulas given above.

STEP 12. Update the velocity components:

$$v_x \leftarrow v_x + ha_x$$

and

$$v_y \leftarrow v_y + ha_y .$$

STEP 13. Increment the time:

$$t \leftarrow t + h.$$

STEP 14. End of STEP 8 loop.

STEP 15. Increment the parameter:

$$p \leftarrow p + dp.$$

STEP 16. End of STEP 5 Loop.

STEP 17. Quit.

PROBLEM 4.7 Generating a Family of Projectile Motion Curves • Implement Algorithm 4.2 with a computer program using the mass of a baseball as the parameter, assuming the other parameters of the baseball are constants. Vary the mass from 0.06 kg to 0.2 kg in steps of 0.02 kg. Assume the initial velocity is 50 ms^{-1} and the angle from the horizontal is 40°. From this study you should be able to make a statement such as: "All other factors being constant, the distance a projectile travels in a resistive medium depends on the mass of the projectile in the sense that as the mass increases ..." You may have to elaborate to make your point.

PROBLEM 4.8 The Best Throwing and Hitting Angle • Use the solution to Problem 4.7 to find the angle at which the range is a maximum for a baseball hit at 50 ms^{-1}.

PROBLEM 4.9 Dropping More Supplies • Repeat Problem 4.4 with air resistance. A bale of hay has a cross-section of about 1 ft by 3 ft and weighs approximately 75 lbs. Compare the two solutions.

PROBLEM 4.10 The Effect of the Density of Air on the Trajectories • Make a family of curves for different values of ρ. Try values of ρ from 1.0 to 1.4 in steps of 0.04. Assume that the object is a softball with $m = 0.19$ kg, $\rho = 0.097$ m, and a drag coefficient of 0.4. Let the initial velocity be 50 ms^{-1} at an angle of 40 degrees from the horizontal. Formulate some thoughts about what to expect, then run the program and, if necessary, revise your conclusions. What change in range occurs when ρ changes from 1.2 to 1.3 kg m^{-3}? Would a change like this be reasonable as part of a daily fluctuation?

4.2.4 Projectile Motion with Variable Gravitational Force

We know that the force of gravity, and therefore the acceleration of gravity, decreases with height. We can derive an equation for the acceleration quite easily. Begin by making a sketch of the unusual coordinate system we are going to use. On a sheet of paper draw a circle representing the earth. Draw a line from the center of the earth toward the top of the sheet of paper. Label this line as the y-axis. Assume that we are "on top of the world" and that "up" on the video monitor or paper is the same as "up" on the earth. Draw the x-axis tangent to the earth's surface at the intersection of the y-axis and the circle . The zero for the y-axis is the center of the earth, but the zero for the x-axis is the point where the circle, the y-axis, and the x-axis intersect. Somewhere above this point, on the y-axis, place a dot and write "m" to represent the projectile.

We will use Newton's Universal Law of Gravitation; the force acting on the projectile is

$$F = \frac{GMm}{y^2},$$

where G is the gravitation constant, M is the mass of the earth, m is the mass of the projectile, and y is the distance between the center of the earth and the projectile. The force vector points downward, toward the center of the earth. Draw and label the force vector and erase the circle. On your paper you have constructed a "flat earth" with the origin of the y-axis at a distance R below the surface of the flat earth, where R is the radius of the real earth.

Apply Newton's Second Law, $F = ma$, to the projectile to obtain

$$\frac{GMm}{y^2} = ma_y.$$

Divide both of the sides of the equation by m and solve for a_y to find

$$a_y = \frac{GM}{y^2},$$

which shows that the gravitational acceleration of an object is independent of its mass.

5.98

For reference purposes $G = 6.67 \times 10^{-11}$ Nm^{2}kg^{-2}, $M = 4.98 \times 10^{24}$ kg, and the radius of the earth, R, is 6.38×10^6 m. If $y = R$ and the values for G, M, and R are substituted into this equation, we find $a_R = 9.81$ ms^{-2}. The ratio a_y / a_R is

$$\frac{a_y}{a_R} = \frac{a_y}{g} = \frac{\frac{GM}{y^2}}{\frac{GM}{R^2}} = \frac{R^2}{y^2},$$

which when solved for a_y yields

$$a_y = g\frac{R^2}{y^2}.$$

In other words, in Algorithm 4.1 modify the equation for a_y to read

$$a_y = -(9.8)(\frac{6.38 \times 10^6}{y})^2,$$

and use $y = 6.38 \times 10^6$ m for the initial value for y. It is also convenient to graph the ratio y/R versus x, rather than y versus x, which means the plot instruction should graph the point $(x, y/6.38 \times 10^6)$ rather than the point (x, y). Finally, the calculation loop should terminate when $y <= 6.38 \times 10^6$ because we need not follow the projectile while it buries itself into the earth.

Graphs showing the trajectory of projectiles for variable acceleration and constant acceleration ($g = 9.8$ ms^{-2}) are given in Figure 4.4. We also allowed the projectile to have a tiny motion in the x coordinate ($\theta = 89.9$ degrees) so we can see a trajectory, and we superimposed a trajectory made with constant acceleration. It should not be difficult for you to understand the difference between the two trajectories. Note that a distance of 1.1 R corresponds to an altitude of 640 km or about 400 miles. The graphs in Figure 4.4 should tell you something about the validity of assuming g is a constant at low altitudes.

PROBLEM 4.11 Trajectory of a Projectile with Variable Acceleration • Change your projectile-motion program to include the modifications just mentioned and plot the trajectory of a projectile launched upward with a speed of 3100 ms^{-1} at an angle 89.9 degrees from the horizontal. The projectile will travel from 1.0 R (by definition) to 1.1R. Use *dom* = [0, 4000] m and *ran* = [1, 1.1] earth radii. Superimpose a graph made with constant acceleration, $a = -g$; the instructions to make such a graph follow. Over what range of altitudes may we assume that $a = -g$, a constant, with less than 1 percent error? You may need to make a table.

A theoretical graph showing the trajectory of a projectile launched from the surface of the earth with constant acceleration ($-g$), may be made with the following considerations. We know

$$x = (v_0 \cos\theta)t$$

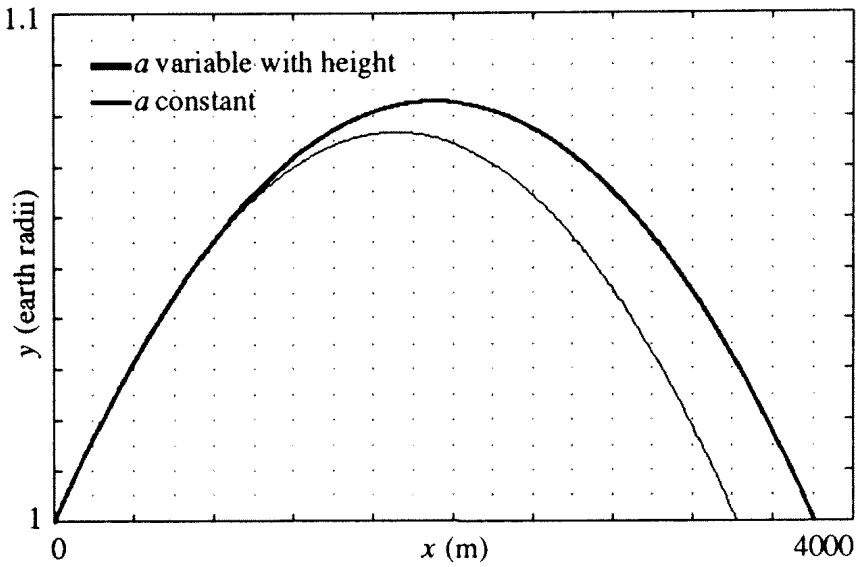


FIGURE 4.4 • Trajectory of a Projectile When Acceleration Is a Function of Altitude. Parameters: v_0 = 3100 ms^{-1}, x_0 = 0, y_0 = 1 earth radius, h = 0.2 s, q = 89.9 degrees, *dom* = [0, 4000] m, *ran* = [1, 1.1] earth radii.

and

$$y = y_0 + (v_0 \sin\theta)t - \frac{1}{2}gt^2.$$

Solving the first equation for t and substituting it into the second gives:

$$y = y_0 + x\tan\theta - \frac{1}{2}g\left(\frac{x}{v_0\cos\theta}\right)^2.$$

If we want y in terms of earth radii and y_0 = 1 earth radius, then

$$y = 1 + \frac{x\tan\theta - \frac{1}{2}g\left(\frac{x}{v_0\cos\theta}\right)^2}{6.38 \text{ x } 10^6}.$$

This gives us the theoretical relationship between x and y that we need to plot a graph of x versus y for constant acceleration.

A worthwhile project would be to investigate trajectories with variable acceleration and variable drag, taking into account that the density of air decreases with altitude. What other factors might influence the trajectory of a high-altitude projectile? Over large distances both Coriolis and centrifugal forces must be taken into account because we live in a rotating, non-inertial reference frame.[4] Indeed, these factors are taken into account in weaponry.

4.3 From Projectiles to Satellites

We are going to change the flat-earth of the previous section to a spherical planet, and we will choose a new coordinate system, as shown in Figure 4.5. We will assume that our planet does not rotate and that it has a mass M such that $GM = 1$. (This assumption simplifies the calculations; in a subsequent section we will include more realistic details.) On this unique planet we have once again given ourselves the great privilege of living on top of the world. In fact, we are standing on the top of a mountain, the planet's only mountain, located on top of the planet. On this mountain we have placed a cannon that can shoot only horizontally, giving the projectile an initial velocity $\mathbf{v}_0$. Refer again to Figure 4.5. The force of gravity, $\mathbf{F}$, acting on the projectile is shown directed toward the center of the planet. If the mass of the mountain is negligible, $\mathbf{F}$ will indeed point to the center of the planet, so we will assume that the mass of the mountain is zero. The displacement of the projectile from the center of the planet is $\mathbf{r}$ and the mass of the projectile is m.

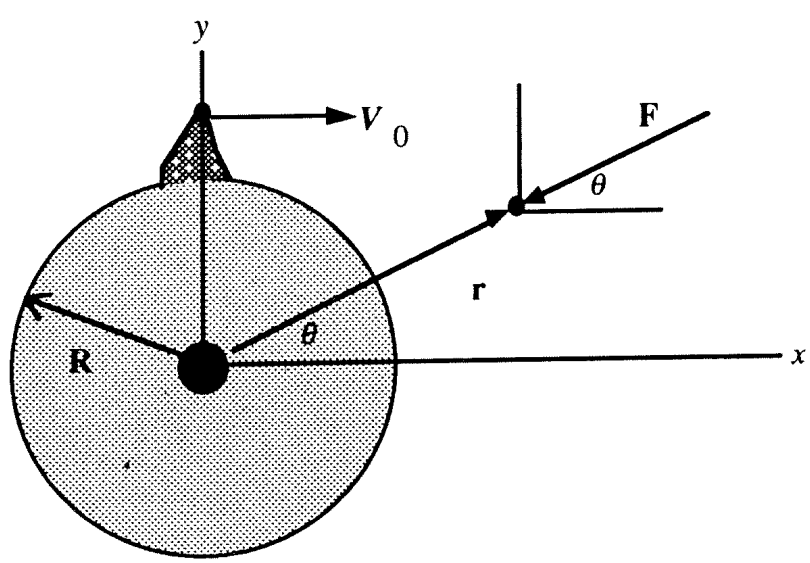
We need to make a few mathematical notes about the projectile's motion. The force acting on it is

$$\mathbf{F} = -\frac{GMm}{r^2}\frac{\mathbf{r}}{r}.$$

The term $\mathbf{r}/r$ is a unit vector along $\mathbf{r}$ and coupled with the minus sign indicates that the force $\mathbf{F}$ is pointed toward the center of the planet, just as Newton would have it. To find the acceleration $\mathbf{a}$ of the projectile, we use Newton's Second Law to obtain

$$\mathbf{a} = \frac{\mathbf{F}}{m} = -\frac{GM}{r^2}\frac{\mathbf{r}}{r}.$$

FIGURE 4.5 • Shooting a Projectile from a Mountain on a Planet.



Once again we see that, in the absence of a resistive medium, the acceleration is independent of the mass of the projectile. (What we have found is the acceleration of the projectile with mass m relative to the earth with mass M, if $m << M$. For our purposes, this is an excellent approximation, and we proceed with the calculations.)

Using our assumption that $GM = 1$ and solving for the x and y components of the acceleration, we obtain

$$a_x = -\frac{\cos\theta}{r^2}$$

and

$$a_y = -\frac{\sin\theta}{r^2}.$$

Clearly the assumption that the product $GM = 1$ simplifies the equations.

Study Figure 4.5 once more and note that

$$\cos\theta = \frac{x}{\sqrt{x^2 + y^2}}$$

and

$$\sin\theta = \frac{y}{\sqrt{x^2 + y^2}}.$$

Substituting these equations into the equations for the components of the acceleration, we will make use of the fact that

$$r^2 = x^2 + y^2.$$

We find that

$$a_x = -\frac{x}{(x^2 + y^2)^{\frac{3}{2}}}$$

and

$$a_y = -\frac{y}{(x^2 + y^2)^{\frac{3}{2}}}.$$

Projectile acceleration on a round planet is more complex than on a flat earth.

Having the equations for the x and y components of the acceleration in our algorithm, we can replace the flat-earth equations with these equations and see what happens when a projectile is launched horizontally. However, a few minor modifications to the program that implements the algorithm will prove to be worthwhile. Up until now we have not worried

about the fact that on the video monitor the x and y scales might be different. They usually are different, and therefore trajectories on the screen of the video monitor are distorted. To solve this problem a quantity called the aspect ratio—the ratio of horizontal to vertical lengths on the screen—is introduced. For example, on the Macintosh this ratio is 1.5. In order to make the scales in the x and y direction identical, it is necessary to multiply each y value by 1.5 or divide each x value by 1.5, whatever is most convenient. If window or world coordinates are used, make the x-window 1.5 times larger than the y-window. The aspect ratio may be different for another computer. To find the aspect ratio for your computer, have the computer program draw a circle on the screen, and adjust the aspect ratio until the circle is indeed a circle rather than an ellipse.

PROBLEM 4.12 Motion of a Projectile Shot Horizontally from a Mountain on a Spherical Planet • Modify previously written projectile programs to make the x and y accelerations of the projectile

$$a_x = -\frac{x}{(x^2 + y^2)^{\frac{3}{2}}}$$

and

$$a_y = -\frac{y}{(x^2 + y^2)^{\frac{3}{2}}}.$$

The initial velocity should have no y component, and the condition

$$x^2 + y^2 < R$$

should terminate the loop. (You may wish to refer to Algorithm 4.3, below.) Here are some parameters with which to begin your calculation experiments: $R = 8, x = 0, y = 10, v_x = 0.15, v_y = 0, h = 0.2$, with $dom = [-18, 18]$ and $ran = [-12, 12]$. What happens to the projectile? Remember, it is being shot from the top of the mountain shown in Figure 4.5.

Here are some additional experiments and questions.

- Except for v_x, keep these same initial conditions. Modify v_x in several steps from 0.2 to 0.3. You can use Algorithm 4.2 in which one parameter, v_x, is varied. What happens to the trajectory of the projectile?
- Next, try $v_x = 0.3162$. What is the trajectory of the projectile in this case? (The picture at the beginning of the chapter is a family of trajectories with v_x as a parameter.)
- Make $v_x = 0.32$ and describe what happens. Try some larger v_x and put your conjectures into words.
- If the planet is the earth, then *perigee* is that point on the orbit closest to the earth and *apogee* is the most distant point on the orbit. Relative to the mountain, where are perigee and apogee?

- Suppose that you have no mountain. Is it possible to put a projectile in orbit from earth in that case?
- Give the projectile some initial velocity in the y direction and describe the results.
- Make R small and make another family of projectile trajectories, with $v_x = 0.2$ to $v_x = 3.2$. If the projectile does not crash, does the projectile always orbit back to its starting point?

If you did the exercises above, then you found that $v_x = 0.3162$ produces a circular orbit. Why? In order for a satellite to move in a circular orbit, the gravitational force must produce the required centripetal acceleration for *constant* speed. In our simple case with $GM = 1$, the gravitational force is m/r^2. We chose $y = 10$ and $x = 0$, so $r = 10$. The centripetal acceleration is v^2/r. Thus Newton's Second Law gives

$$m\frac{1}{r^2} = m\frac{v^2}{r}.$$

Cancelling the mass, substituting $r = 10$, and solving for v gives $v = 0.3162$ for the speed of the projectile in circular motion.

4.4 More About Orbits

PROBLEM 4.13 Velocity of a Satellite • Modify the program of Problem 4.11 to plot the points (v_x, v_y) as well as or instead of the points (x, y) on the orbit. A line drawn from the origin to the point (v_x, v_y) is the velocity vector at that instant. You must multiply the velocity vector by some constant to make a suitable graph. What is the direction of the velocity vector relative to the displacement vector at any instant? What kind of curves do you obtain when you graph (v_x, v_y)? Interpret these curves, and identify corresponding points on the v_x-v_y curve with points on the x-y orbit.

PROBLEM 4.14 Acceleration of a Satellite • Modify the program of Problem 4.11 to plot the points (a_x, a_y) as well as the points (x, y) on the trajectory of the satellite. What is the direction of the acceleration vector relative to the displacement vector at any instant? In the case of a satellite orbiting the earth, what causes this acceleration?

PROBLEM 4.15 More Orbits • So far we have seen only circular orbits. The program of Problem 4.12 will also generate other orbits; simply change the initial conditions. We drew a family of orbits with the program, using $x_0 = 25$, $y_0 = 0$, and $v_x = 0$, with v_y as the parameter to be varied. The results are shown in Figure 4.6. You might begin with $x, y, v_x, v_y = 25, 0, 0, 0.1$, respectively. The graph should have $dom = [-75, 75]$ and $ran = [-50, 50]$. An h of 0.05 is sufficiently small to give good results. Experiment with the program yourself. Here are some questions to be discussed.

- How could you modify your program to take advantage of the symmetry, namely, if (x, y) is a point on the orbit, then $(x, -y)$ is also on the orbit?
- What happens if the initial velocity has both an x component and a y component?

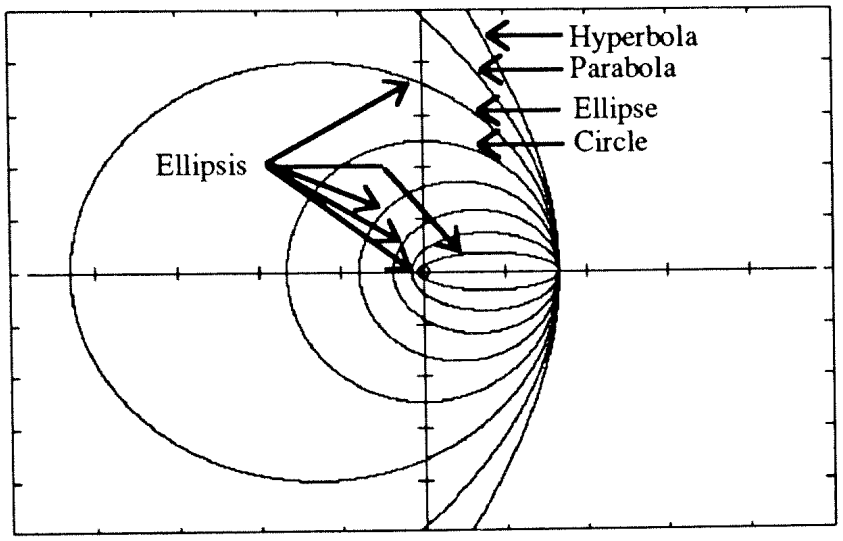


FIGURE 4.6 • A Family of Orbits with v_{y0} as Parameter. Parameters and initial conditions: *Dom* = [−75, 75], *ran* = [−50, 50], $h = 0.05$, at $t = 0$, $x = 25$, $y = 0$, $v_x = 0$, and v_y steps from 0.04 to 0.32 in steps of 0.04.

- Where are the foci of the ellipsis?
- Do all the elliptical orbits have the same focus?
- What is the significance of a parabolic orbit? If a comet were in a parabolic orbit would it return?
- What is the significance of a hyperbolic orbit?
- Under what circumstances does the satellite escape?
- Refer to Problem 4.18 below and calculate the total energy of the object for each of the initial conditions described in Figure 4.6. How does the total energy change as the orbits are changed from ellipsis to parabolas and to hyperbolas? If $GM = m = 1$, then the formula for energy is

$$E = -\frac{GMm}{r} + \frac{1}{2}mv^2 = -\frac{1}{r} + \frac{1}{2}v^2.$$

4.5 The Orbit of the Earth Around the Sun and the Orbit of an Electron Around a Proton

Our last look at the motion of a satellite involves the motion of the earth around the sun and the motion of an electron around a proton. The purpose of this section is to show that the programs can deal with real parameters as well as artificial ones, $GM = 1$ for example. Here

are the data you need to plot the orbit of the earth:

$$G = 6.672 \text{ x } 10^{-11} \text{Nm}^2\text{kg}^{-2},$$

$$M = 1.99 \text{ x } 10^{30} \text{kg},$$

$$x = 1.496 \text{ x } 10^{11} \text{m}, \; y = 0,$$

and

$$v_x = 0, v_y = 2.93 \text{ x } 10^4 \text{ms}^{-1}.$$

We used $h = 8640$ s (a tenth of a day). We have revised Algorithm 4.2 specifically for orbital motion of the earth. The modifications appear in Algorithm 4.3. A suitable choice for the graph's domain and range is [– 2.25, 2.25] x 10^{11} m and [– 1.5, 1.5] x 10^{11} m, respectively.

ALGORITHM 4.3 / An Algorithm to Model the Motion of the Earth around the Sun

STEP 1. Let the step-size $h = 8{,}640$ s.
STEP 2. Choose the initial conditions given above.
STEP 3. Let $p = GM = 1.328$ Nm^{2}kg^{-1}. Then

$$a_x = -p\frac{x}{(x^2 + y^2)^{\frac{3}{2}}}$$

and

$$a_y = -p\frac{y}{(x^2 + y^2)^{\frac{3}{2}}}.$$

STEP 4 Find the half-step velocity components with the formulas

$$v_{x\frac{1}{2}} = v_x + \frac{h}{2}a_{x0}$$

and

$$v_{y\frac{1}{2}} = v_y + \frac{h}{2}a_{y0}.$$

STEP 5. Begin a loop. Loop until the condition for ending is met. A good condition for ending is when the y coordinate changes from – to +.

STEP 6. Tabulate $x(t)$, $y(t)$, or plot the current position (x, y).
STEP 7 Calculate new coordinates with the expressions

$$x \leftarrow x + hv_x$$

and

$$y \leftarrow y + hv_y.$$

STEP 8. Calculate the x and y components of the acceleration from the equations given above.
STEP 9. Update the velocity components:

$$v_x \leftarrow v_x + ha_x$$

and

$$v_y \leftarrow v_y + ha_y.$$

STEP 10 Increment the time:

$$t \leftarrow t + h.$$

STEP 11. End of STEP 5 loop.
STEP 12. Quit.

In the Bohr model of the hydrogen atom, the electron orbits the proton in a circular orbit. To plot the orbit of this electron in a hydrogen atom, we make use of Coulomb's Law

$$F = k\frac{Q_1 Q_2}{r^2},$$

which gives the magnitude of the force of attraction between the proton and the electron in its orbit at a distance r from the proton. The quantity $k = 8.99 \times 10^9$ Nm^2C^{-2}. For simplicity we will assume a circular orbit with the proton at the origin of our coordinate system. The Coulomb force provides the centripetal force required to keep the electron in a circular orbit. For the electron in a hydrogen atom with one proton as the nucleus, both of the charges are 1.602×10^{-19} C in magnitude . Thus Newton's Second Law becomes

$$a = -\frac{ke^2}{m}\frac{1}{r^2},$$

and the parameter p in Algorithm 4.3 is replaced by the expression

$$p = \frac{ke^2}{m}.$$

You need the following information to complete the problem:

$$Q_1 = Q_2 = e = 1.60 \times 10^{-19}\ \text{C},$$
$$k = 8.99 \times 10^{9}\ \text{Nm}^2\text{C}^{-2},$$
$$m = 9.11 \times 10^{-31}\ \text{kg},$$
$$x_0 = 5.3 \times 10^{-11}\ m,\ y_0 = 0,$$

and

$$v_{x0} = 0,\ v_{y0} = 2.19 \times 10^{6}\ \text{ms}^{-1}.$$

Change the values used to plot the orbit of the earth to these values, and run the program to plot the trajectory of the electron. A suitable choice of the graph's domain and range for this problem is $[-9, 9] \times 10^{-11}$ m and $[-6, 6] \times 10^{-11}$ m, respectively, and let $h = 10^{-19}$ s. Notice the huge change in scale between the hydrogen atom problem and the problem with the earth orbiting around the sun. The ratio of the radii of the two orbits is approximately 3×10^{21} and there is a similar scale change in the step-size h.

So far we have plotted only the trajectory of a satellite, be it an electron, the earth, or an imaginary point. What about the attracting mass or charge? It also moves in an orbit. In the case of the earth orbiting around the sun, the mass of the sun is approximately 300,000 times the mass of the earth, so for all practical purposes the sun does not move and the center of mass of the system is in the center of the sun. Likewise, in the hydrogen atom the mass of the proton is approximately 2000 times that of the electron, and it may be regarded as stationary. For our purposes, neglecting the small but finite mass of the orbiting object is an acceptable approximation.

PROBLEM 4.16 Earth and Electron Orbits • Write a program to calculate and graph the orbit of the earth around the sun and the orbit of an electron around a proton.

PROBLEM 4.17 Dropping the Earth into the Sun • Run the earth orbit program described above, but use the initial condition $v_y = 0$; in other words, assume that the earth has no initial velocity. Like an apple dropping to the ground, it should drop into the sun.

4.6 Additional Problems

PROBLEM 4.18 Conservation of Energy in Projectile Motion • Modify the programs you have written for projectile motion to calculate the kinetic and potential energy of the projectile as a function of time. Kinetic energy is $(1/2)mv^2$, and potential energy is mgh. It is instructive to graph one or both of these quantities as a function of x. In other words, graph $(x, K.E.)$ or $(x, P.E.)$ or both. You will have to scale them to fit them on the same graph as the height y. The way to do this is to first print $K.E.$ and $P.E.$, and then from the printed values

you can find the appropriate scale factor. Graph the total mechanical energy, kinetic plus potential, as a function of x. In other words, as the orbit is graphed you should also plot the points (x, E) where

$$E = \frac{1}{2}mv^2 + mgh.$$

PROBLEM 4.19 Kinetic and Potential Energy in the Presence of Air Drag • Repeat Problem 4.18 for the motion of a projectile with air resistance. Is energy conserved?

PROBLEM 4.20 Graphing v versus r in Satellite Motion • Make a graph of the magnitude of **v** versus the magnitude of **r** as a satellite completes an orbit. Is an earth satellite moving most rapidly at apogee or at perigee?

PROBLEM 4.21 Other Central Forces • Experiment with different central-force laws. We used a r^n law with $n = -2$. What happens if n is 3/2, 1, or –1?

4.7 References

[1] C. Frohlich, "Aerodynamic Drag Crisis and its Possible Effect on the Flight of Baseballs," *Am. J. Phys.* 52, (April 1984): 324.

[2] R. G. Watts and S. Baroni, "Baseball-Bat Collisions and the Resulting Trajectories of Spinning Balls," *Am. J. Phys.* 57, (January 1989): 40.

[3] D. T. Kagan, "The Effects of Coefficient of Restitution Variations on Long Fly Balls," *Am. J. Phys.* 58, (February 1990): 151.

[4] W. E. Gettys, F. K. Keller, and M. J. Skove *Classical and Modern Physics* (New York: McGraw-Hill, 1989).113.

4.8 Programs

PROGRAM 4.8.1 / Projectile Motion

```
! Algorithm 4.1, Problems 4.1 - 4.5, etc.
! Try vo = 25, theta = 37, h = 0.01, r = 0.03, and m = 0.1 (an apple)
! For no drag take k = 0.
! If k = 0 take [0, 70] as domain, [0, 15] as range.
! For baseball throw take vo = 44.7, theta = 40, m = 0.145, r = 0.0367
! Use domain [0, 100], range [0, 30].
LIBRARY "ComPhys"
!.................
! Define acceleration functions here.
DEF ax = - k*sqr(vx^2 + vy^2)*vx
DEF ay = - 9.81 - k*sqr(vx^2 + vyo^2)*vy
!.................
```

```
CALL Initial_Cond(x,y,vx,vy,t,h,k)                          ! Enter the initial conditions.
!let k = 0 for no drag case. Otherwise use k from Initial_Cond.
CALL OPEN (xmin, xmax, ymin, ymax, wxmin, wxmax, wymin, wymax)
LET vx = vx + ax*h/2                                        ! Calculate half-step
LET vy = vy + ay*h/2                                        ! velocity.
DO until y < 0                                              ! This is the main loop.
   PLOT x,y                                                 ! Plot the coordinates of the
                                                              projectile.
   LET x = x + h*vx                                         ! Calculate the new x position.
   LET y = y + h*vy                                         ! Calculate the new y position.
   LET vx = vx + h*ax                                       ! Velocity calculation.
   LET vy = vy + h*ay
   LET t = t + h                                            ! Increment the time.
LOOP                                                        ! Loop until y < 0.
CALL Label (wxmin, wxmax, wymin, wymax)
END
!...........
SUB Initial_Cond(x,y,vx,vy,t,h,k)                           ! These are the initial
                                                              conditions.
    INPUT PROMPT "Enter the magnitude of the initial velocity. ":v
    INPUT PROMPT "Enter the angle (in degrees) between v and the +x axis. ": theta
    LET theta = theta*pi/180                                ! Convert to radians.
    INPUT PROMPT "Enter the small time interval, h. ":h
    INPUT PROMPT "Enter the mass m ":m
    INPUT PROMPT "Enter the radius r. ":r
    LET k = 0.25*pi*r*r*1.2/m                               ! k is a constant.
    LET t = 0                                               ! Start at t = 0.
    LET x = 0                                               ! Start projectile at (0,0)
    LET y = 0
    LET vx = v*cos(theta)                                   ! Find the x component of the
                                                              velocity.
    LET vy = v*sin(theta)                                   ! Find the y component of the
                                                              velocity.
END SUB
```

PROGRAM 4.8.2 / A Family of Trajectories with Different Initial Velocities

```
! Algorithm 4.2, Problems 4.7 - 4.10
! For baseball try domain [0, 200] range [0, 75]
LIBRARY "ComPhys"
CALL OPEN (xmin, xmax, ymin, ymax, wxmin, wxmax, wymin, wymax)
!...............
DEF ax = - k*sqr(vx^2 + vy^2)*vx                            ! Acceleration in the x
                                                              direction
DEF ay = - 9.81 - k*sqr(vx^2 + vy^2)*vy                     ! Acceleration in the y
                                                              direction.
!...............
FOR I = 20 to 80 step 10                                    ! 7 trajectories.
    CALL Initial_Cond(x,y,vx,vy,t,h,k)                      ! Enter the initial conditions.
```

```
   LET v = i                                  ! Initial velocity is
   LET vx = v*cos(40*pi/180)                  ! the family parameter.
   LET vy = v*sin(40*pi/180)                  ! Calculate components.
   LET vx = vx + ax*h/2                       ! Half-step velocity.
   LET vy = vy + ay*h/2
   DO until y < 0                             ! This is the main loop.
      PLOT x,y;                               ! Plot the coordinates of the
                                                projectile.
      LET x = x + h*vx                        ! Calculate the new x position.
      LET y = y + h*vy                        ! Calculate the new y position.
      LET vx = vx + h*ax                      ! Velocity calculation.
      LET vy = vy + h*ay
      LET t = t + h                           ! Increment the time.
   LOOP                                       ! Inside loop for one
   PLOT                                       ! trajectory.
NEXT I                                        ! Outside loop for family
                                                member.
CALL Label (wxmin, wxmax, wymin, wymax)
END
!..........
SUB Initial_Cond(x,y,vx,vy,t,h,k)             ! These are the initial
                                                conditions.
    LET theta = 40*pi/180                     ! Convert to radians.
    LET h = .01                               ! Step-size h.
    LET m = .145                              ! Mass of a baseball.
    LET r = .0732/2
    LET A = PI*r*r                            ! Cross section of baseball.
    LET k = 0.23*A*1.2/m                      ! a = -k*v^2, k is a constant.
    LET t = 0                                 ! Start at t = 0.
    LET x = 0                                 ! Start projectile at (0,0)
    LET y = 0
    LET vx = v*cos(theta)                     ! Find the x component of the
                                                velocity.
    LET vy = v*sin(theta)                     ! Find the y component of the
                                                velocity.
END SUB
```

PROGRAM 4.8.3 / Projectile with Variable Acceleration

```
! Algorithm 4.1, Problem 4.11
! domain [0, 4000]m range [1, 1.1] earth radii, v = 3100 ms-1-1h = 0.2 s
LIBRARY "ComPhys"
!.........
!Define accelerations.                       ! Acceleration in the x
DEF ax = 0                                     direction
                                              ! Acceleration in the y
DEF ay = - 9.81*(6.38e6/y)^2                   direction.
!.........
CALL Initial_Cond(x,y,vx,vy,t,h,k)            ! Enter the initial conditions.
```

```
CALL OPEN (xmin, xmax, ymin, ymax, wxmin, wxmax, wymin, wymax)
LET vx = vx + ax*h/2
LET vy = vy + ay*h/2
DO until y < 6.38e6                                    ! This is the main loop.
   PLOT x,y/6.38e6;                                    ! Plot the coordinates of the
                                                         projectile.
   LET x = x + h*vx                                    ! Calculate the new x position.
   LET y = y + h*vy                                    ! Calculate the new y position.
   LET vx = vx + h*ax                                  ! Velocity calculation.
   LET vy = vy + h*ay
   LET t = t + h                                       ! Increment the time.
LOOP
PLOT
LET x = 0                                              ! Plot theoretical curve
LET theta = 89.9*pi/180                                ! with no drag.
let vy = 3100
DO until y< 1                                          ! Plot theoretical curve.
LET y =1 +(x*tan(theta) - (9.81*x^2)/(2*(vy*cos(theta))^2))/6.38e6
PLOT x,y;
LET x = x + 10
LOOP
CALL Label (wxmin, wxmax, wymin, wymax)
END
!...........
SUB Initial_Cond(x,y,vx,vy,t,h,k)                      ! These are the initial
                                                         conditions.
 INPUT PROMPT "Enter the magnitude of the initial velocity. ":v
 INPUT PROMPT "Enter the angle (in degrees) between v and the +x axis. ": theta
 LET theta = theta*pi/180                              ! Convert to radians.
 INPUT PROMPT "Enter the small time interval, h. ":h
 LET t = 0                                             ! Start at t = 0.
 LET x = 0                                             ! Start projectile at (0,0)
 LET y = 6.38e6
 LET vx = v*cos(theta)                                 ! Find the x component of the
                                                         velocity.
 LET vy = v*sin(theta)                                 ! Find the y component of the
                                                         velocity.
END SUB
```

PROGRAM 4.8.4 / Simple Orbits

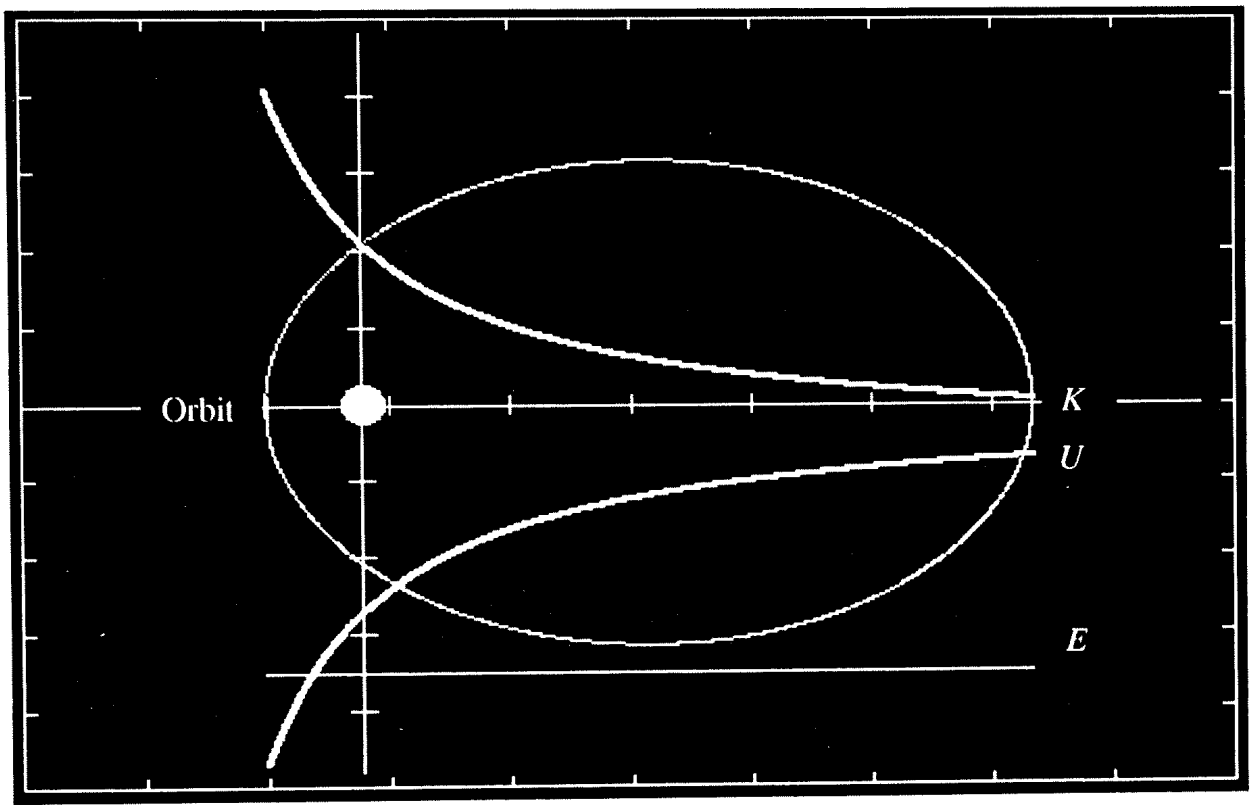
```
! Algorithm 4.1, Problems 4.12 - 4.15
! Use range [-12, 12], domain is scaled accordingly.
LIBRARY "Comphys"
!..............
!Calculate the accelerations.
DEF rth = (x^2 + y^2)^(3/2)
DEF ax = -x/rth                                        ! Acceleration in the x
                                                         direction!
```

```
DEF ay = - y/rth                                                    ! Acceleration in the y
                                                                      direction.
!...........
LET h = 0.3                                                         ! Parameter h.
CALL OPENB (xmin,xmax,ymin,ymax,wxmin,wxmax,wymin,wymax)! In ComPhys.
CALL Initial_Cond (x,y,vx,vy,t,h)                                   ! Enter the initial conditions.
BOX CIRCLE -8,8,-8,8
LET vx = vx + h*ax/2                                                ! Calculate the new velocity.
LET vy = vy + h*ay/2                                                ! Ditto for y coordinate.
DO                                                                  ! This is the main loop.
   PLOT x,y
   LET xt = x
   LET x = x + h*vx                                                 ! Calculate the new x position.
   LET y = y + h*vy                                                 ! Calculate the new y position.
   LET vx = vx + h*ax                                               ! Velocity calculation.
   LET vy = vy + h*ay
   LET t = t + h                                                    ! Increment the time.
   IF xt < 0 and x>= 0 then EXIT DO                                 ! Quit when orbit crosses
   IF x*x + y*y < 64 then EXIT DO
LOOP                                                                ! y-axis.
CALL Label (wxmin, wxmax, wymin, wymax)
END                                                                 ! End of the main program.
!...........
SUB Initial_Cond(x,y,vx,vy,t,h)                                     ! Enter the initial
   LET t = 0
   LET x = 0                                                        ! conditions.
   LET y = 10
   LET vx = 0.21
   LET vy = 0
END SUB
```

PROGRAM 4.8.5 / Earth Orbit

```
! Algorithm 4.3, Problem 4.16, 4.17, 4.21,etc.
! Choose range [-1.6 e11, 1.6e11].
LIBRARY "ComPhys"
DEF rth = (x^2 + y^2)^(3/2)
DEF ax = -GM*x/rth                                                  ! Acceleration in the x
                                                                      direction!
DEF ay = - GM*y/rth                                                 ! Acceleration n the y
                                                                      direction.
LET GM = 6.672e-11*1.99e30                                          ! G times mass of sun.
LET h = 3600                                                        ! parameter h.
CALL OPENB (xmin,xmax,ymin,ymax,wxmin,wxmax,wymin,wymax)            ! In ComPhys.
CALL Initial_Cond (x,y,vx,vy,t)                                     ! Enter the initial conditions.
LET vx = vx + h*ax/2                                                ! Calculate the new velocity.
LET vy = vy + h*ay/2                                                ! Ditto for y coordinate.
DO                                                                  ! This is the main loop.
   IF MOD (t,8640) = 0 then PLOT x,y                                ! Plot points once a day.
```

```
   LET yt = y
   LET x = x + h*vx                                  ! Calculate the new x position.
   LET y = y + h*vy                                  ! Calculate the new y position.
   LET vx = vx + h*ax                                ! Velocity calculation.
   LET vy = vy + h*ay
   LET t = t + h                                     ! Increment the time.
   IF yt < 0 and y>= 0 then EXIT DO                  ! Quit when orbit crosses
LOOP                                                 ! x-axis.
PRINT t
CALL Label (wxmin, wxmax, wymin, wymax)
END                                                  ! End of the main program.
!..........
SUB INITIal_Cond (x, y, vx, vy,t)
   LET t = 0
   LET x = 1.5201e11                                 ! Initial conditions.
   LET y = 0
   LET vx = 0
   LET vy = 29.3e3
END SUB
```



Satellite Orbit

5 / Energy

5.1 Introduction

So far we have solved physics problems using Newton's Second Law of Motion and a simple numerical technique we call the Feynman-Newton method. It should be clear that Newton's Second Law makes the concept of force of central importance to the study of dynamics. However, there are other concepts of equal importance that we have either omitted or merely touched upon in the previous chapters, namely energy and momentum. Indeed, many physics problems may be solved using only the concepts of energy and/or momentum. In this chapter we will examine the systems studied in earlier chapters, this time from the point of view of energy. The purpose of this chapter is twofold:

- To gain a deeper understanding of the concept of energy
- To see how the concept of energy may be used to check the validity of the numerical techniques we have used in previous chapters

Before beginning, you may wish to review the following topics in your textbook:

- Potential energy (gravitational and elastic), kinetic energy, and work
- The Work-Kinetic Energy Theorem (WKET)
- Conservation of energy (COE)
- Conservative and nonconservative forces

Energy is a less intuitive concept than force. We can feel many forces, but it is more difficult to sense energy. Of course, we all have some intuition about energy: We may feel energetic, the food we eat is characterized by the number of calories (of chemical energy) per serving, and we can feel when we are "out of energy." In another sense energy is simply an idea, an abstract concept, not even real. What physicists do is invent formulas for different kinds of energy—mgh, $(1/2)mv^2$, kQq/r, $(1/2)kx^2$, etc. The important thing about these formulas for energy is that when thinking about an isolated dynamical system, the sum of the appropriate energy terms remains constant. We describe this by saying energy is conserved. When we find an isolated system in which the sum of its energy terms is not constant, we will invent a new term rather than drop the concept that the sum is constant. In 1927 the noted physicist Wolfgang Pauli invented the neutrino, a completely new particle, for just this purpose. It is in this context that we will use energy in the discussion that follows, namely the sum of the energy terms must remain constant. If the sum is not constant, then our calculations are in error or perhaps the system is not isolated and energy is being exchanged with the surroundings.

5.2 Energy of an Object on a Spring

5.2.1 Calculating and Graphing K, U, and E

The system we will study is shown in Figure 5.1. It consists of an object on the end of a spring. For reference purposes, the one-dimensional equation of motion for an object on the end of a spring is

$$ma = -kx - cv,$$

where m is the mass of the object, k is the spring constant of the spring, and c is the damping coefficient. To start the system vibrating, we might stretch the spring and release the object. We assume that the surface-block interface is frictionless, but the medium in which the mass moves is viscous, providing a damping force equal to $-cv$. The minus sign indicates that the direction of the damping force is opposite to the velocity. Figure 5.1 shows the object moving toward the right, so the damping force, $-cv$, is toward the left.

In your physics textbook you will find the derivations of the formulas for various energy terms. In this section we will use formulas for kinetic energy, $(1/2)mv^2$, and elastic potential energy, $(1/2)kx^2$, where x and v refer to the position and velocity of the center of mass of the system, respectively. The origin of our coordinate system is at the center of mass of the block when the spring is not stretched. In what follows we will call the kinetic energy K, the potential energy U, and the mechanical energy E, where $E = K + U$.

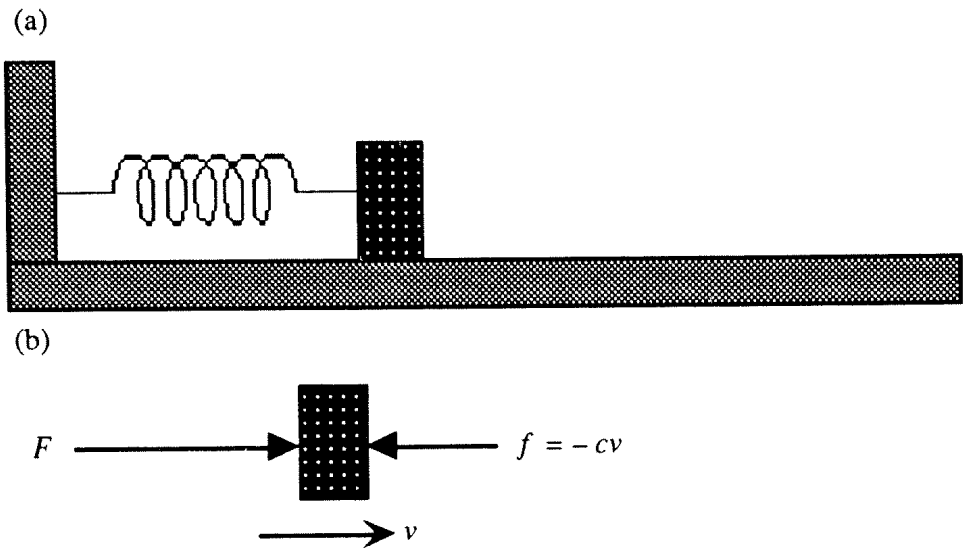


FIGURE 5.1 • (a) An Object Attached to a Spring. (b) The Horizontal Forces in the Equation of Motion of the Mass on the Spring. F is the force of the spring on the mass, and f is the damping force.

To understand the object-spring problem in terms of energy, we modified Algorithm 2.2 to include a procedure to calculate kinetic energy and elastic potential energy. The modified version is given in Algorithm 5.1, one which we shall use throughout this chapter. There is one minor problem. Recall that when using the Feynman-Newton method the positions and the velocities of the object are calculated for slightly different times. This affects the calculated value of the mechanical energy, E, but the error is not large enough to be troublesome.

ALGORITHM 5.1 / The Feynman-Newton Algorithm with Energy Calculations

STEP 1. Find the sum of the forces acting on the object and use this to solve for the acceleration,

$$\mathbf{a} = \frac{\mathbf{F}}{m}.$$

STEP 2. Determine the initial conditions $\mathbf{r}_0$, $\mathbf{v}_0$, and $\mathbf{a}_0$.
STEP 3. Start the velocity calculation:

$$\mathbf{v}_{\frac{1}{2}} = \mathbf{v}_0 + \frac{h}{2}\mathbf{a}_0.$$

STEP 4. Begin a loop. Loop until a key is pressed or $t > t_{max}$.

STEP 5. Plot points on the graphs of $\mathbf{r}(t)$, $\mathbf{v}(t)$, and/or $\mathbf{a}(t)$.
STEP 6. Calculate K, U, and their sum, E

- For a block of mass m on a spring in a one-dimensional setting:

$$E = \frac{1}{2}kx^2 + \frac{1}{2}mv^2.$$

- For a falling object or a projectile of mass m at height h (v is the magnitude of the velocity $\mathbf{v}$):

$$E = mgh + \frac{1}{2}mv^2.$$

- For a satellite of mass m orbiting around an object with a large mass, M, (r is the magnitude of the displacement vector $\mathbf{r}$ from m to M, and v is the magnitude of $\mathbf{v}$):

$$E = -\frac{GMm}{r} + \frac{1}{2}mv^2.$$

In these formulas U is the first term on the right-hand side, and K is the second.
STEP 7. Plot one or more of these points: $U(t)$, $K(t)$, or $E(t)$.
STEP 8. Update $\mathbf{r}$:

$$\mathbf{r} \leftarrow \mathbf{r} + h\mathbf{v}.$$

STEP 9. Calculate the acceleration $\mathbf{a}$ from Newton's Second Law.
STEP 10. Update $\mathbf{v}$:

$$\mathbf{v} \leftarrow \mathbf{v} + h\mathbf{a}.$$

STEP 11. Update t:

$$t \leftarrow t + h.$$

STEP 12. End of STEP 4 loop (exit on key closure or when $t > t_{max}$).
STEP 13. Quit.

To include energy calculations in your program, all you need to do is add instructions such as

```
K = 0.5*m*v^2,
U = 0.5*k*x^2,
```

and

```
E = K + U,
```

and be careful that the spring constant k and the kinetic energy K are not confused. Add plot instructions to graph these quantities versus time t. Graphs of $K(t)$ and $U(t)$ for an object-spring system are shown in Figure 5.2. Is the sum $K + U$ constant?

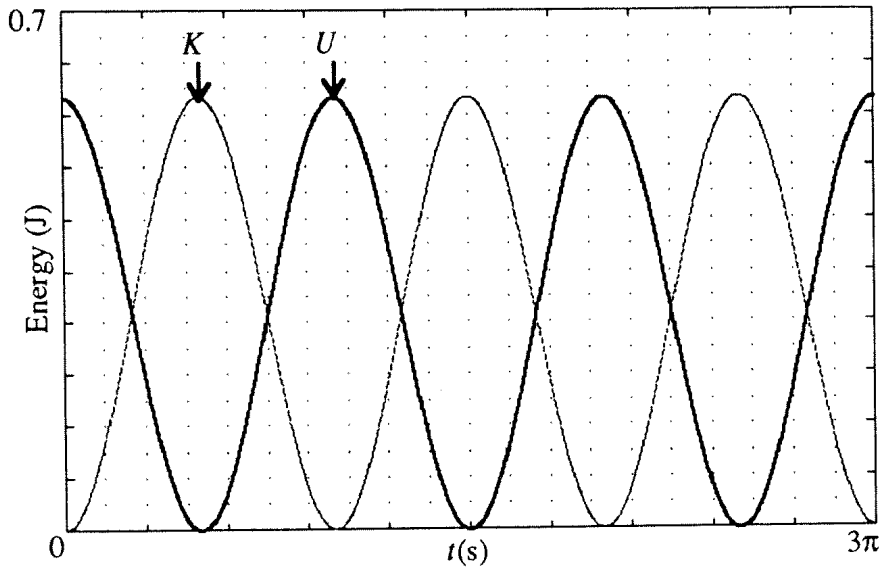


FIGURE 5.2 • Kinetic and Potential Energy in a Object-Spring System. Parameters and initial conditions: m = 1 kg, k = 1 Nm^{-1}, c = 0, x_0 = 1 m, v_0 = 0 ms^{-1}, *dom* = [0, 3π] s, and *ran* = [0, 0.6] J.

PROBLEM 5.1 Effect of the Step-Size *h* on the Total Energy • Modify the object-spring program you wrote in Chapter 2 to graph $U(t)$ and $K(t)$ on the same screen. Let *dom* = [0, 9.4] s, *ran* = [0, 0.7] J, $x_0 = 1$, $v_0 = 0$, m = 1 kg, k = 1 Nm^{-1}, and let $c = 0$. Experiment with changes in the step-size h to see how the total energy is affected by the choice of step size. Try values from $h = 0.01$ s to $h = 2$ s. You may wish to expand the time scale of the graph to longer times. What happens to $K + U$ for large h? Might the conservation of energy help us choose h?

PROBLEM 5.2 The Effect of Damping on the Energy Curves • Using $h = 0.01$ s, run the program of Problem 5.1, this time with a damping constant c = 0.1 kgs^{-1}. Plot $K(t)$ and $E(t)$. Let k = 1 Nm^{-1}, m = 1 kg, *dom* = [0, 12] s, and *ran* = [0, 0.7] J. Describe your results, noticing the terraced shape of the $E(t)$ curve. Explain this terraced shape. During what parts of the cycle does the system lose mechanical energy most rapidly? Why? At what instants is the system losing no energy? Why? Experiment with different values of the damping coefficient to discover how c affects the results.

PROBLEM 5.3 Graphing *K* and *U* Versus Displacement • Modify the program of Problem 5.1 to graph $U(x)$ and $K(x)$. Scale the energy to stay on the graph. Make the loop end only when a key is pressed. Let x_0 = 1 m, v_0 = 0, *dom* = [−1.2, 1.2] m, *ran* = [0, 0.8] J, m = 1 kg, k = 1 Nm^{-1}. First try $c = 0$, then try c = 0.1 kgs^{-1}. Write an explanation of your results. How can you tell from this graph that the oscillator loses mechanical energy most rapidly when it is near $x = 0$?

5.2.2 Calculating the Internal Thermal Energy of a Dissipative System (Very Challenging)

Of course, you know what happens when the damping coefficient c is not zero. The mechanical energy, $K + U$, of the oscillator is converted into internal thermal energy of the mass and the damping medium. The *mechanical* energy of the system is said to be dissipated. The purpose of this section is to calculate the various energy terms, including the internal thermal energy, and see if we can balance our energy checkbook. To begin, we will define the system. Next, we will use both the principle of the conservation of energy (COE) and the work-kinetic energy theorem(WKET) to identify the thermal-energy term. This is a complex calculation. After that, we will show how to compute the internal thermal energy at each step in the numerical solution of the equation of motion. Finally, we will utilize this information in our calculations as a check on the conservation of energy and to gain deeper insights into the motion of a dissipative object-spring system.

We take as our system the combination object-spring system and the medium responsible for the damping. This choice has the advantage of eliminating all external forces. To analyze this system, we begin by expressing the principle of the conservation of energy (COE) with the equation

$$\Delta E = Q + W,$$

where ΔE is the sum of the energy changes of the system, Q is the heat added to the system, and W is the work done on the system by external forces. Because there are no external forces to do work, $W = 0$. We also assume there is no heat received from the surroundings, making $Q = 0$.

With these simplifications there is only one term in the COE equation, but it is not as simple as it looks because the internal energy of our system has several forms: the kinetic energy of the mass, the potential energy of the spring, and the thermal energy of the mass and the surrounding medium. Thus

$$\Delta E = K_t - K_0 + U_t + U_0 + (E_{th})_t - (E_{th})_0 = 0.$$

The "t" subscript means "at time t" and the "0" subscript means "at $t = 0$." A term such as $(K_t - K_0)$ is the *change* in the kinetic energy of the mass. E_{th} is the internal *thermal* energy of the system, energy that resides in the motions of the molecules within the block and the surrounding medium. This COE equation simply says that for the system we have chosen, *the sum of the energy changes is zero.* An equivalent way of expressing the COE is as follows:

$$K_t + U_t + (E_{th})_t = K_0 + U_0 + (E_{th})_0.$$

Expressed in words, the total energy of the system at any time t is equal to its initial energy, which, of course, is just some number, i.e. a constant. If in our program we calculate and add K_t, U_t, and $(E_{th})_t$, the sum should remain constant. If the sum is not constant, our calculations are suspect.

It is easy to calculate K_t and U_t; they are $(1/2)mv^2$ and $(1/2)kx^2$, respectively. Calculating E_{th} is more difficult, as you will see in the next few paragraphs. (You may wish to review the

sections of your physics textbook related to the WKET.) We intend to show that

$$E_{th} = c\int_0^t [v(t)]^2\,dt.$$

We will also explain how this integral may be evaluated using a computer. Specifically, we will show that

$$E_{th} = c\int_0^t [v(t)]^2\,dt = ch\sum_{i=1}^{i=n} v_i^2 = ch[v_1^2 + v_2^2 + \cdots + v_n^2],$$

where v_i is the velocity of the object during the i_{th} time interval. If you are not interested or concerned about the details of the derivation, skip the next four paragraphs. We include them because not all textbooks include this topic.

We turn to the calculation of E_{th}. The steps are almost identical to the derivation of the WKET found in several physics textbooks. Newton's Second Law applied to the center of mass of the block is

$$F_k - f_c = ma = m\frac{dv}{dt},$$

where F_k is the force of the spring on the block whose mass is m, and f_c is the damping force of the viscous medium in which the block moves. We substitute $-kx$ for F_k and integrate this equation with respect to x to obtain the following preliminary form of the WKET:

$$-\int_{x_0}^{x_t} kx\,dx - \int_{x_0}^{x_t} f_c\,dx = \int_{x_0}^{x_t} m\frac{dv}{dt}dx.$$

We will examine each of the terms of the WKET in more detail. Carrying out the integration of the first term gives

$$-\int_{x_0}^{x_t} kx\,dx = -\frac{1}{2}kx_t^2 + \frac{1}{2}kx_0^2.$$

We recognize the terms on the right-hand side of this equation as elastic potential energy terms, $-U_t$ and U_0. Thus

$$-\int_{x_0}^{x_t} kx\,dx = -U_t - U_0.$$

The integral on the right-hand side of the WKET gives

$$\int_{x_0}^{x_t} m\frac{dv}{dt}dx = m\int_{x_0}^{x_t} \frac{dv}{dx}\frac{dx}{dt}\,dx = m\int_{x_0}^{x_t} v\frac{dv}{dx}dx.$$

To make this integration more like those you have seen in calculus, we will accent the fact that v is a function of x by writing $v(x)$, and we will use the prime notation for the derivative;

that is,

$$\frac{dv}{dx} = v'(x).$$

The last integral then assumes this form:

$$m\int_{x_0}^{x_t} v\frac{dv}{dx}dx = m\int_{x_0}^{x_t} v(x)v'(x)\,dx.$$

Next, we use an integration technique you may know as "u substitution." Let

$$u = v(x),$$

so

$$du = v'(x)dx.$$

The integration continues as follows:

$$m\int_{x_0}^{x_t} v(x)v'(x)dx = m\int_{v_0}^{v_t} u\,du = m\left[\frac{u^2}{2}\right]_{v_0}^{v_t} = \frac{1}{2}mv_t^2 - \frac{1}{2}mv_0^2 = K_t - K_f.$$

When we changed the limits of integration from x-limits to u-limits, we used the fact that $v = v_0$ when $x = x_0$, and $v = v_t$ when $x = x_t$.

Substitute these integrals into the WKET and rearrange terms to obtain this relationship:

$$-\int_{x_0}^{x_t} f_c\,dx = U_t - U_0 + K_t - K_0.$$

(Although this equation contains "energy terms," it is not a statement about conservation of energy. It is, in fact, derived from Newton's Second Law, not the COE. Moreover, the forces involved in its derivation may be either conservative or nonconservative forces. Not all textbooks are equally clear on this subject.) The COE equation and the WKET equation both contain the terms $U_t - U_0 + K_t - K_0$. We can, therefore, eliminate these terms between the two equations and find that

$$-\int_{x_0}^{x_t} f_c\,dx = (E_{th})_t - (E_{th})_0.$$

This tells us that the change in the thermal energy of our system is the negative of the integral of the damping force over x. Now we have a way of calculating E_{th}. In what follows we will assume, with no loss of generality, that $(E_{th})_0 = 0$. Thus

$$-\int_{x_0}^{x_t} f_c\,dx = E_{th}.$$

We have previously decided that

$$f_c = -cv.$$

Thus the integral we must evaluate is

$$-\int_{x_0}^{x_t} f_c\,dx = c\int_{x_0}^{x_t} v\,dx.$$

where v is the velocity of the center of mass of the block. Our calculations turn out to be simpler if we change the variable of integration from x to t. We can do that with the substitution

$$dx = x'(t)dt = v(t)dt$$

and obtain

$$E_{th} = -\int_{x_0}^{x_t} f_c\,dx = c\int_{x_0}^{x_t} v\,dx = c\int_0^t v(t)^2\,dt.$$

At long last we have the relationship we want:

$$E_{th} = c\int_0^t [v(t)]^2\,dt.$$

This is the integral we must evaluate to calculate E_{th}. Note that this integral is always positive. Hence the thermal energy increases until the mass comes to rest at its unstretched position. How shall we evaluate this integral? It cannot be evaluated in the usual way with the fundamental theorem of calculus (FTC), which requires that we know the antiderivative of $v(t)$. We do not even have an analytic function for $v(t)$, let alone for its antiderivative. We must use *numerical integration techniques* to evaluate this integral.

Recall from calculus that a definite integral may be defined as the limit of a sum. (If you have forgotten the definition of a definite integral, you may wish to review the appropriate chapters in your calculus textbook.) Here is one form that the definition may take:

$$\int_a^b f(x)dx = \lim_{\Delta x \to 0} \sum_{i=1}^{i=n} f(x_i)\Delta x,$$

where

$$\Delta x = \frac{b-a}{n}.$$

For our purposes, x becomes t and Δx becomes h. With that in mind, the integral definition above gives

$$\int_0^t [v(t)]^2\,dt = \lim_{h \to 0} \sum_{i=1}^{i=n} [v(t_i)]^2 h,$$

where we have replaced $f(x)$ with $[v(t)]^2$. Next, we make the assumption that this sum will be a good approximation to the integral if we keep h small. In that case,

$$E_{th} = c\int_0^t [v(t)]^2\,dt \cong c\sum_{i=1}^{i=n} [v(t_i)]^2 h = ch\sum_{i=1}^{i=n} [v(t_i)]^2,$$

where we have assumed that the sum is over all of the intervals (or steps) of duration h from time zero to the current time, t. The last equation is, perhaps, more simple when expressed in subscript notation. Thus

$$E_{th} = c\int_0^t [v(t)]^2\,dt \cong c\sum_{i=1}^{i=n}[v(t_i)]^2 h = ch\sum_{i=1}^{i=n} v_i^2 = ch[v_1^2 + v_2^2 + \cdots + v_n^2].$$

The sum on the right-hand side of this equation is easily evaluated in either a computer program or on a spreadsheet. The generic instruction

```
Eth = Eth + c*h*v^2,
```

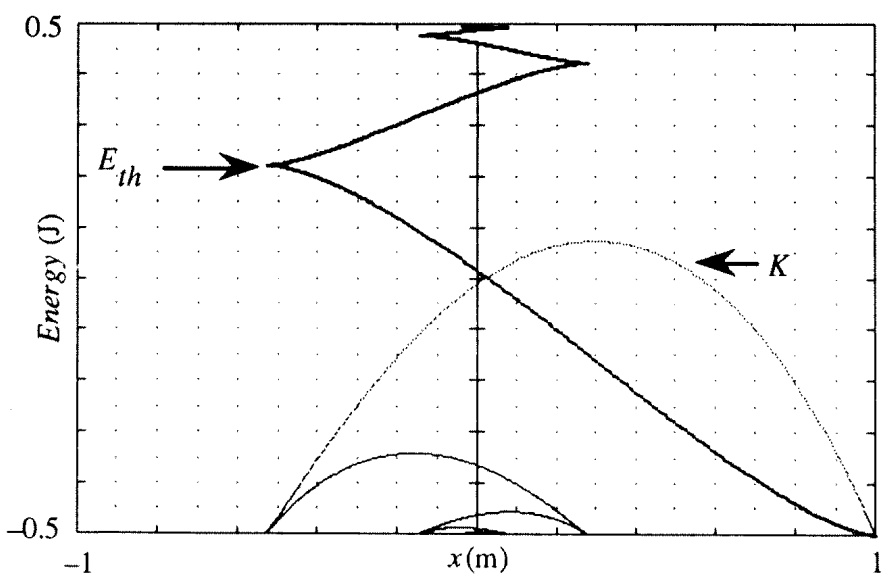
when placed in a loop, will accomplish the task. With a spreadsheet, you can simply sum the contents of the preceding cell with a new chv^2 term.

PROBLEM 5.4 Calculating the Mechanical Energy Lost in a Damped Oscillator • Modify the programs of Problem 5.1 - 5.3 to calculate the mechanical energy converted into thermal energy. In other words, find the sum of all the chv^2 terms described above. Make use of the fact that

$$E_{th} = c\int_0^t [v(t)]^2\,dt \cong ch\sum_{i=1}^{i=n} v_i^2 = ch[v_1^2 + v_2^2 + \cdots + v_n^2].$$

Have the program calculate and plot $K + U + E_{th}$ versus time. Choose $k = 1\ \text{Nm}^{-1}$, $m = 1$ kg, $c = 0.15\ \text{kgs}^{-1}$, $h = 0.01$, $dom = [0, 12]$ s, and $ran = [0, 0.7]$ J. Choose initial conditions $x_0 = 1$ m and $v_0 = 0$. Does $K + U + E_{th}$ remain constant? Next graph E_{th} versus x. Where in the cycle does E_{th} increase most rapidly? Why? Interpret the graphs in Figure 5.3?

FIGURE 5.3 • A Graph of E_{th} and K versus Displacement x for an Object on a Spring. Try to give a conceptual explanation of this figure. Parameters: $m = 1$ kg, $k = 1\ \text{Nm}^{-1}$, $h = 0.01$ s, $x_0 = 1$ m, $v_0 = 0$, $c = 0.4.\text{kgs}^{-1}$, $dom = [-1, 1]$ m and $ran = [0, 0.5]$ J. The action begins in the lower-right-hand corner with initial conditions $x_0 = 1$ m and $v_0 = 0$, a result of the fact that the object is released when the spring is stretched.



5.3 Energy in Projectile Motion

In projectile motion there are also two forms of energy—kinetic energy and gravitational potential energy. We must choose where the potential energy is zero. If we choose the zero level to be the surface of the earth, then, assuming g is constant, the formula for the gravitational potential energy is mgh. With drag, some of the mechanical energy is converted to the internal energy of the surroundings.

PROBLEM 5.5 Conservation of Energy in Projectile Motion • Modify your falling-object or projectile-motion programs to calculate and graph one or more of the following quantities as a function of time: kinetic energy, potential energy, mechanical energy, or thermal energy. Using the methods of the previous section, we can show that the thermal energy term is

$$E_{th} = \frac{1}{2}CA\rho\int_0^t |v(t)|^3\,dt \cong \frac{1}{2}CA\rho\sum_{i=1}^{i=n}|v(t_i)|^3 h.$$

In this equation C is the drag coefficient, A is the cross sectional area of the projectile, ρ is the density of air (1.2 kg m^{-3}) , and v is the speed of the object. In subscript notation this formula is

$$E_{th} \cong \frac{1}{2}CA\rho h\sum_{i=1}^{i=n}|v_i^3|.$$

Run the program to answer the following questions:

- Is $K + U$ constant if $C = 0$?
- What happens to the total energy if the step-size h is increased from 0.01 s to 2.0 s? Try this with $C = 0$.
- How is the total energy affected if the drag coefficient is not zero? To find out, choose parameters that may be characteristic of a baseball: $h = 0.01$ s, $C = 0.5$, $m = 0.145$ kg, $r = 0.036$ m, $\theta = 45°$, $r_0 = 50$ ms^{-1}, $x_0 = y_0 = 0$, $dom = [0, 10]$ s, and $ran = [0, 50]$ m.
- At what points is the rate of energy loss the greatest? Why?
- Is $K + U + E_{th}$ constant?

5.4 Energy in Satellite Motion

5.4.1 Gravitational Potential Energy: Background

When an object is near the surface of the earth, we give the gravitational potential energy the value mgh. This is a result of the definition of potential energy difference and the assumption that the force of gravity is constant near the earth's surface. The potential energy difference, $U_b - U_a$, is defined as the work you must do (or the negative of the work done by gravity) in

moving a mass m from height h_a to height h_b, that is

$$U_b - U_a = \int_a^b F_y\, dy,$$

where F_y is the force you must apply to lift the object of mass m from point a to point b. Near the surface of the earth we assume $F_y = mg$, where g is constant. This gives

$$U_b - U_a = mg(y_b - y_a).$$

Because we are only interested in *changes* in potential energy, we may choose the point at which U is zero. We choose $U_a = 0$ when $y_a = 0$. Substitution of these values into the preceding equation gives $U_b = mgy_b$. Typically we set $y_b = h$ and write $U = mgh$.

In reality the force of gravity is

$$F_g = -\frac{GMm}{r^2}.$$

In this equation $r = R_e + a$, where R_e is the radius of the earth and a is the altitude of the object. The approximation $F = mg$ is good only when a is small compared to R_e.

In the study of satellite motion, the assumption that F_g is constant will not be valid. Furthermore, it is not convenient to take the zero point for the potential energy at the surface of the earth. We must introduce the complications described in the next section.

5.4.2 Potential Energy in the Vicinity of a Single Mass

We keep the definition that potential energy is the work you must do to move the object, but now the gravitational force is variable. As a result, the formula for gravitational potential energy is

$$U_b - U_a = GMm\int_{r_a}^{r_b} \frac{1}{r^2}dr.$$

In this equation r_a and r_b are distances from the mass center of mass M. (Remember, the force we apply is opposite to the force on the object, hence the change in signs from $-GMm$ to GMm.) Our choice of the zero for potential energy will be obvious from the next equation,

$$U_b - 0 = \int_\infty^{r_b} \frac{1}{r^2}dr = -\frac{1}{r_b}.$$

Thus the potential energy may be taken as zero where the distance r between the two objects becomes infinite. Or, we may think of the potential energy as the work required to move the mass from infinity to r_b. Thus the formula for the gravitational potential energy, U_g, of a mass m at a distance r from a mass M is

$$U_g = -\frac{GMm}{r}.$$

The minus sign in the expression indicates that the gravitational potential energy of m increases as the distance r between m and M becomes larger, quite like a stone thrown upward.

5.4.3 Total Energy in Orbital Motion

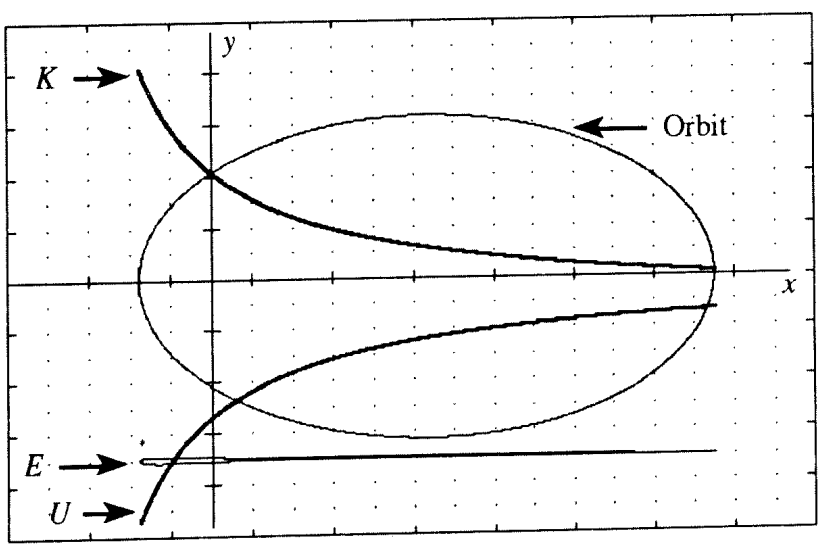
From the preceding discussion it should be clear that the total mechanical energy of an object with mass m orbiting with velocity $\mathbf{v}$ another object whose mass is M will be given by the following expression:

$$E = -\frac{GMm}{r} + \frac{1}{2}mv^2.$$

We have assumed that the mass m of the satellite is much smaller than mass M. This assumption means that the center of mass of the system is at the mass center of M. Thus in our coordinate system whose origin is at the mass center, M is at rest, it has no kinetic energy, and $\mathbf{r}$ is simply a vector from the origin to the much smaller mass m. This approximation is very good in the case of a satellite orbiting the earth, the earth orbiting the sun, or an electron orbiting a proton.

To study the various energy changes that take place in orbital motion, we need to add to our programs only a few simple calculations to find U and K. The output of our program is shown in Figure 5.4, which was made with a very small step size, $h = 0.2$. Figure 5.4 includes the trajectory of the satellite and *scaled* graphs of its kinetic, potential, and total energy as a function of the x-coordinate. Notice that both the kinetic energy and the potential energy depend on the position of the object in its orbit.

FIGURE 5.4 • Kinetic, Potential, and Total Energy in Orbital Motion. The energy graphs are scaled to fit the coordinate system, and the total energy E is magnified. Initial conditions: $x_0 = 25$, $y_0 = 0$, $v_x = 0$, $v_y = 0.1$. Parameter values: $GM = 1$, $h = 0.1$, $dom = [-10, 30]$, and $ran = [-15, 15]$.



You can see that the total energy E is *almost* constant. In this case, the departure from being perfectly constant is strictly a result of the fact that in the Feynman-Newton method the potential energy and kinetic energy are computed at times separated in by $h/2$. If this apparent non-constant E is troublesome, calculate the components of the velocity in two steps, first from the beginning of the h-interval to the middle of the interval, then from the middle of the h-interval to its end. Sandwich the calculation of K between these two steps, as follows:

```
vx = vx + ax*h/2
vy = vy + ay*h/2
KE = 0.5*m*(vx^2 + vy^2)
PLOT x, KE
vx = vx + ax*h/2
vy = vy + ay*h/2.
```

Assume that we are studying the motion of a satellite around the earth with the earth at the origin. The point in the orbit that is nearest the earth is called perigee, while the point on the orbit that is most distant is called apogee.

- At what point in the orbit, perigee or apogee, is the kinetic energy a maximum?
- At what point in the orbit is the potential energy a maximum?
- Are there orbits for which the total energy is zero or positive? We will ask this question again, but what do you think *now*?
- What happens to the speed of the satellite as it approaches perigee? Apogee?

PROBLEM 5.6 Effect of the Step-Size *h* on the Total Energy in Orbital Motion • Study the effect of the step-size h on the conservation of energy. Use the parameters from Figure 5.4. Try step sizes of 0.05, 0.5, 1 and 5. How does the step size affect the total energy? Where in the orbit is the energy most sensitive to step size? How can conservation of energy be used to determine the validity of the calculations? Since we know that the energy of this system must be conserved, if the total energy E varies then something is incorrect in the calculations. If h is sufficiently small, there should be no problem. If h is too large or there is a mistake in the calculations, the effect will generally appear in the energy calculations as a non-constant total energy E.

PROBLEM 5.7 Highly Elliptical Orbits • Run the program of Problem 5.6 with a step size of 0.1 keeping all of the initial conditions the same except v_y, which you should give the value 0.01 rather than 0.1. The smaller the initial velocity v_y, the more elliptical is the orbit. First describe what happens, then try to explain the results.

PROBLEM 5.8 The Relationship of Total Energy to the Type of Orbit • Choosing the initial conditions for x, y, v_x, and v_y is the same as making a choice of the mechanical energy E because

$$E = -\frac{GMm}{\sqrt{x^2 + y^2}} + \frac{1}{2}m(v_x^2 + v_y^2),$$

and E is constant. Thus the initial conditions fix the total energy E. We wish to observe the effect of different initial energies on the motion of the satellite. With this in mind, let $h = 0.1$, $x = 25$, $y = 0$, $v_x = 0$ and try the following initial values for v_y: 0.1, 0.2, 0.2829 and 0.32. Choose $dom = [-75, 75]$ and $ran = [-50, 50]$ for the domain and range, respectively. Calculate the total energy in each case. Compare your results with the orbits in Figure 4.6. What does total energy have to do with the shape of the orbit? Under what conditions will a particle escape rather than be a satellite? Try an initial v_y of 0.2828 once again. Does the satellite return? How long do you need to wait to see if the satellite returns?

5.5 Angular Momentum in Orbital Motion

Of equal importance to the conservation of energy is the conservation of momentum. In the case of orbital motion we are interested in the angular momentum **L**. The angular momentum about the origin of a single particle with displacement **r** and velocity **v** is a vector quantity defined as the cross product

$$\mathbf{L} = m\mathbf{r} \times \mathbf{v}.$$

In two dimensions this becomes

$$\mathbf{L} = m(x\mathbf{i} + y\mathbf{j}) \times \left(v_x\mathbf{i} + v_y\mathbf{j}\right) = (xv_y - yv_x)\mathbf{k}.$$

The coefficient of the unit vector **k** is quite easy to calculate; we can simply modify our programs to calculate the magnitude of the momentum rather than (or in addition to) the energy. In a computer program the instruction will be of the form:

```
L = x*vy - y*vx
PLOT x, L.
```

The direction of **L** is out of the plane of the orbit; only the **k** component is non-zero.

PROBLEM 5.9 Angular Momentum in Orbital Motion • Modify your orbit program to calculate and graph the magnitude of the angular momentum. You may wish to experiment with various types of orbits. Is angular momentum conserved for parabolic and hyperbolic orbits as well as elliptical orbits? What effect does the step-size h have on the calculated value of the angular momentum? Try a highly elliptical orbit. Is angular momentum conserved?

5.6 Additional Problems

PROBLEM 5.10 Energy of a Damped and Driven Oscillator (Challenging) • Repeat Problems 5.1 - 5.4 when there is a driving force present. Use

$$ma = -kx - cv + F_d \cos(2\pi ft).$$

Start with $F_d = 1$ and $f = 1$, but try different driving amplitudes and frequencies for the driving force, graphing and studying the changes in the various energy terms. Describe and explain the results of your experiments in a few paragraphs. In particular, note what happens to the energy when the object-spring system is driven at its resonance frequency,

$$f_r = \frac{1}{2\pi}\sqrt{\frac{k}{m}}.$$

PROBLEM 5.11 Work done by the Driving Force (Challenging) • Repeat the experiments in Problem 5.10, this time calculating and graphing the work done by the driving force, W_d, where

$$W_d = \int_{x_0}^{x_t} F_d \cos(2\pi f t)\, dx = F_d \int_0^t v(t)\cos(2\pi f t)\, dt.$$

You will have to approximate the integral with the following sum:

$$W_d \cong hF_d \sum_{i=1}^{i=n} v(t_i)\cos(2\pi f t_i) = hF_d \sum_{i=1}^{i=n} v_i \cos(2\pi f t_i).$$

In this sum, i is simply the number of integrations performed (or h-steps taken), and neither i nor n enter the calculations; simply perform a running sum of the product $v_i\cos(2\pi t_i)$. Under what circumstances is the work positive? Negative?

PROBLEM 5.12 Changing the Energy of a Satellite in Orbit (Challenge) • Write a program that allows you to change the energy of a satellite in orbit by applying forward or reverse thrust. Forward thrust is in the direction of v and it increases the total energy; reverse thrust decreases it. For example, reverse thrust simulates air drag slowing the satellite down. Write your program so that pressing the "cursor right" key produces forward thrust, pressing the "cursor left" key produces reverse thrust. Monitor both the position of the satellite and its total energy.

Experiment with this program. Start with $h = 0.2$, $x = 25$, $y = 0$, $v_x = 0$, and $v_y = 0.15$. Simulate a satellite spiraling into the ground. How would you dock your satellite with one in an orbit with a larger radius? Smaller radius? For a real challenge, add instructions to the program that will have two satellites in motion, one with the thrust capability. Then try to dock one satellite with the other.

5.7 Programs

PROGRAM 5.7.1 / Energy in the Object-Spring Problem

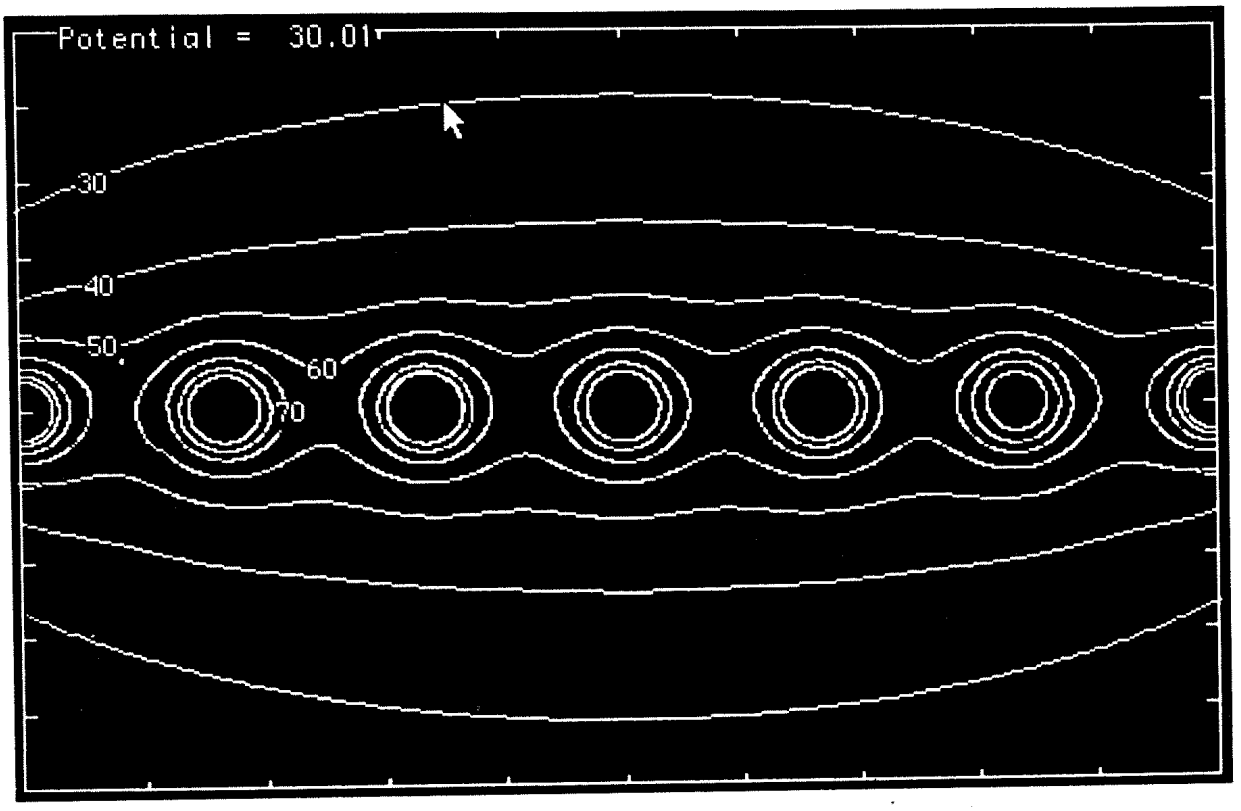
```
! Algorithm 5.1, Problems 5.1 - 5.3.
! Begin with k = m = 1 and c = d = 0. Then try c = 0.2, d=0. Next try c = 0.2, d = 1, and w = 3.
! Use domain [0, 20],range [0, 10].
```

```
!.............
! Acceleration from Newton's Second Law is defined here.
DEF a(x,v,t) = -k*x/m - c*v/m + d*cos(w*t)
!.............
LIBRARY "Comphys*"
CALL GET_INFO (x,v,k,m,c,d)                                  ! Fetch the parameters.
CALL Open(xmin, xmax, ymin, ymax, wxmin, wxmax, wymin, wymax)
CALL Save (wxmin,wxmax,wymin,wymax,keep$)                    ! Save the coordinates.
LET w = 2*sqr(k/m)                                           ! Frequency of driving force.
LET h = 0.05                                                 ! Make the small time interval
                                                               0.05 seconds.
LET v = v + a(x,v,t)*h/2                                     ! Starting value for v using
                                                               Feynman-Newton.
DO until KEY INPUT                                           ! This is the main loop.
   LET x = x + h*v
   LET U = 0.5*k*x*x                                         ! Calculate potential energy.
   LET KE = 0.5*m*v*v                                        ! Calculate kinetic energy.
   PLOT tp, KE
   LET E = U + KE
   LET Eth = Eth + c*v*v*h                                   ! Calculate thermal energy.
   PLOT tp, Eth                                              ! Graph the total energy.
   LET v = v + a(x,v,t)*h
   LET t = t + h                                             ! Clock time.
   LET tp = tp + h                                           ! Plotting time.
   IF tp > xmax then
      LET tp = xmin                                          ! Start a new frame?
      CLEAR                                                  ! New screen.
      CALL Show (wxmin,wymin, keep$)                         ! Bring the axes back.
   END IF
LOOP
GET KEY z                                                    ! Clear keyboard input.
CALL LABEL (wxmin, wxmax, wymin, wymax)                      ! Label graph.
END
!............
SUB GET_INFO (x,v,k,m,c,d)
    INPUT PROMPT "What are the initial conditions? Input xo and vo. ":x,v
    INPUT PROMPT "Please enter k and m. ":k,m
    INPUT PROMPT "Please enter the damping coefficient c. ":c
    INPUT Prompt "Please enter the driving force coefficient d. ":d
END SUB
```

PROGRAM 5.7.2 / Energy of a Satellite

```
! Algorithm 5.1, Problems 5.6 - 5.8.
! choose x,y,vx,vy = 25,0,0, 0.1 choose h = 0.1
! choose a domain of [-10, 30],and choose a range of [-15,15]
Library "ComPhys"
!................
DEF ax = -x/((x^2 + y^2)^(3/2))                              ! Acceleration in the x
                                                               direction.
```

```
DEF ay = - y/((x^2 + y^2)^(3/2))                          ! Acceleration in the y
                                                            direction.
!..........
CALL GET_INFO (h)                                         ! Get parameters R and h.
CALL OPEN (xmin, xmax, ymin, ymax, wxmin, wxmax, wymin, wymax)
CALL Initial_Cond (x,y,vx,vy,t,h)                         ! Enter the initial conditions.
LET vy = 0.1
LET vx = vx + h*ax/2                                      ! Velocity calculation when
LET vy = vy + h*ay/2                                      ! t is  zero.
DO
   LET yt = y
   LET x = x + h*vx                                       ! Calculate the new x position.
   LET y = y + h*vy                                       ! Calculate the new y position.
   LET vx = vx + h*ax                                     ! Velocity calculation.
   LET vy = vy + h*ay
   LET t = t + h                                          ! Increment the time.
   LET r = SQR(x^2 + y^2)                                 ! Calculate r.
   LET vsq = vx^2 + vy^2                                  ! Calculate the square of the
                                                            velocity.
   LET KE = 0.5*vsq                                       ! Calculate the kinetic energy.
   LET PE = -1/r                                          ! Calculate the potential
                                                            energy.
   LET E = KE + PE                                        ! Find the total energy.
   PLOT x, 75*KE                                          ! Graph KE vs.. x.
   PLOT x, 75*PE                                          ! Graph PE vs.. x.
   PLOT x, 300*E                                          ! Graph E vs.. x.
   PLOT x, y                                              ! Graph the orbit.
   IF yt <0 and y => 0 THEN EXIT DO                       ! Crosses x-axis from - to +.
LOOP
CALL Label (wxmin, wxmax, wymin, wymax)
END                                                       ! End of the main program.
!..........
SUB GET_INFO (h)                                          ! Get parameters of the problem.
   INPUT PROMPT "Input the small time interval, h. ":h
END SUB
!..........
SUB Initial_Cond(x,y,vx,vy,t,h)                           ! Enter the initial conditions.
   LET t = 0
   LET x = 25                                             ! Start the projectile at (0,10)
                                                            and
   LET y = 0                                              ! Shoot it horizontally.
   LET vx = 0
END SUB
```



Equipotential Lines

6 / Electric Fields and Potential

6.1 Introduction

In this chapter we will begin working with the forces and fields of charged particles. We will show how the computer may be used to visualize the electric field and graph equipotential lines. In a subsequent chapter we will study the motion of a charged particle in an electric field. Reviewing the following topics may be helpful:

- Coulomb's Law
- Definition of the electric field intensity, **E**
- Potential difference and potential
- Equipotential surfaces and lines

We quickly summarize the important facts and definitions. The force of charge Q on charge q is given by Coulomb's Law

$$\mathbf{F} = k\frac{Qq}{|\mathbf{r}_q - \mathbf{r}_Q|^3}(\mathbf{r}_q - \mathbf{r}_Q),$$

where $(\mathbf{r}_q - \mathbf{r}_Q)$ is a vector from Q to q, $k = 8.988 \times 10^9\ \mathrm{Nm^2C^{-2}}$, and the charges Q and q are measured in Coulombs. The notation

$$|\mathbf{r}_q - \mathbf{r}_Q|$$

means "*the magnitude of* $\mathbf{r}_q - \mathbf{r}_Q$." Notice that the vector $\mathbf{F}$ will point in a direction from q opposite Q if the charges have the same sign; otherwise the force will point from q toward Q. Refer to Figure 6.1.

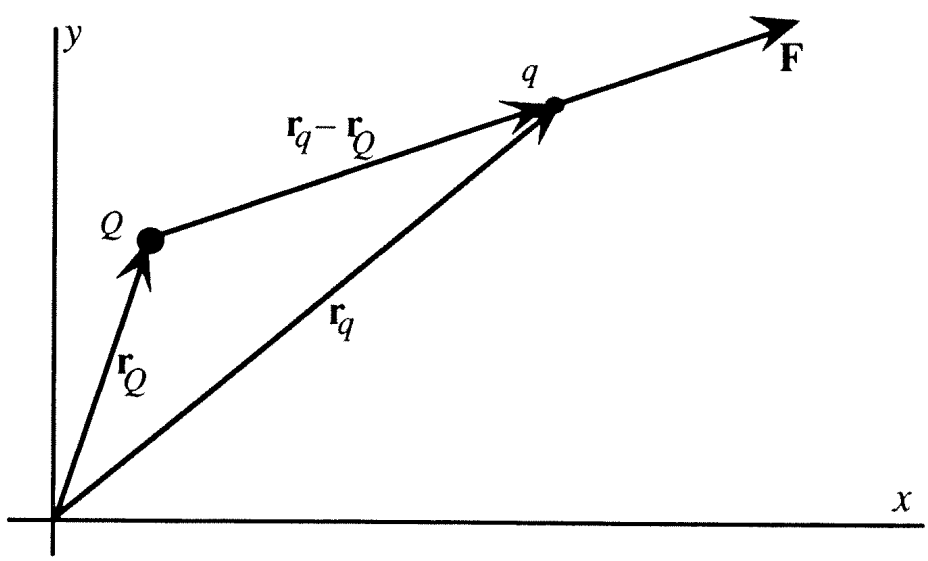
The electric field at a point is defined as the ratio of $\mathbf{F}$ to q in the limit as q, a positive charge, approaches zero. Symbolically,

$$\mathbf{E} = \lim_{q \to 0} \frac{\mathbf{F}}{q}.$$

If Q is an isolated point charge, then the two preceding equations allow us to calculate the electric field in the vicinity of the charge Q, at a point identified by the displacement vector $\mathbf{r}$. In that case $\mathbf{E}$ is given by the expression

$$\mathbf{E} = k\frac{Q}{|\mathbf{r} - \mathbf{r}_Q|^3}(\mathbf{r} - \mathbf{r}_Q).$$

FIGURE 6.1 • Force **F** of Charge Q on Charge q.



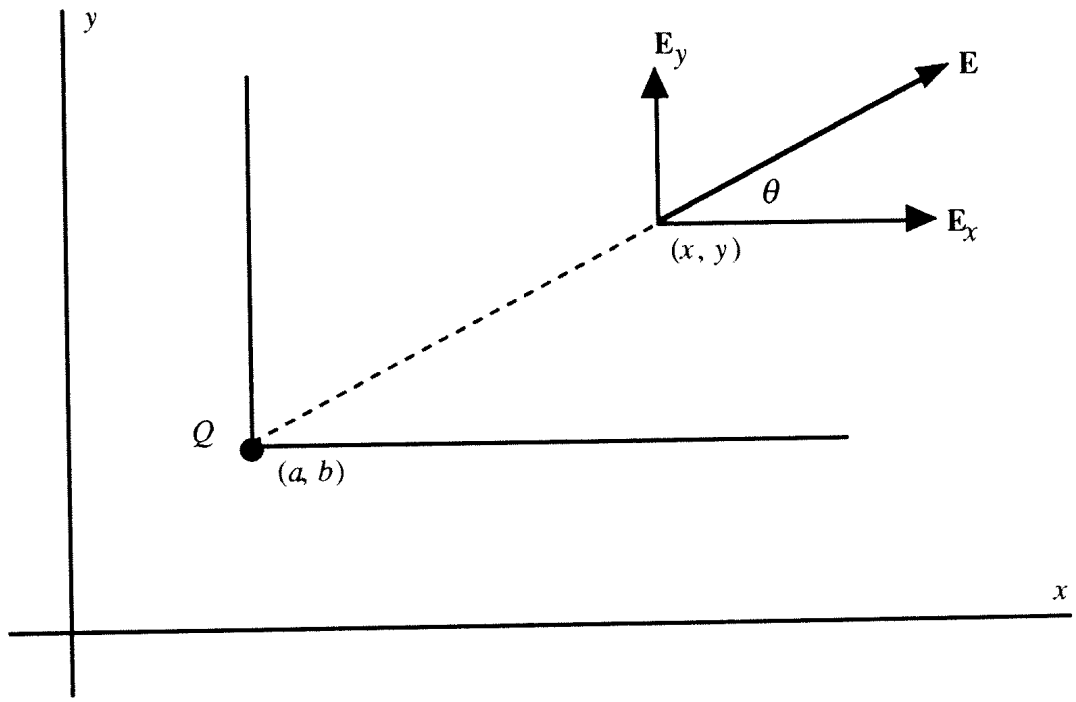


FIGURE 6.2 • Components of the Electric Field at (x, y) Due to a Charge Q at (a, b).

If we place Q at the origin and let E be the magnitude of $\mathbf{E}$, then

$$E = k\frac{Q}{r^2},$$

and r is the distance from the origin to the point where the field is being calculated.

If there is more than one point charge, then the electric field is the vector sum of the electric fields of the individual charges. A vector sum is usually carried out by finding the x, y, and z components of the electric field due to each charge and adding the components associated with each coordinate axis. Consider a two-dimensional case, and suppose charge Q is located at the point with coordinates (a, b). We wish to find the x and y components of the electric field at point (x, y) due to Q alone. The geometry is illustrated in Figure 6.2.

You should verify the following calculations to find the x and y components of E at the point (x, y). Let

$$x_d = x - a$$

and let

$$y_d = y - b.$$

Also let

$$r = \sqrt{x_d^2 + y_d^2},$$

to obtain

$$E_x = E\cos\theta = k\frac{Q}{r^2}\cos\theta = k\frac{Q}{r^2}\frac{(x-a)}{r} = kQ\frac{x_d}{(x_d^2+y_d^2)^{\frac{3}{2}}}$$

and

$$E_y = E\sin\theta = k\frac{Q}{r^2}\sin\theta = k\frac{Q}{r^2}\frac{(y-b)}{r} = kQ\frac{y_d}{(x_d^2+y_d^2)^{\frac{3}{2}}}.$$

If there are N charges Q_i ($i = 1$ to N), then we calculate the x and y components of the **E**-field due to each charge, as we have above, and add them to find the x component and the y component of the resultant. Thus, we can find the **E**-field of N charges at any point (x, y).

It is useful to be able to visualize the electric field, and to do this it is customary to draw *field lines*. These are lines that originate on + charges and terminate on – charges and have **E** as a tangent at each point in space. The number of such lines through a unit cross section perpendicular to **E** is proportional to the magnitude of **E**. In your textbook you will find pictures of **E**-field lines for several different charge configurations. It is important to realize that **E**-field lines are not real; they are simply a conceptual aid for human beings. In the next few paragraphs we will show how the computer can be used to plot two-dimensional information about the electric field.

6.2 Two-Dimensional Representations of the Electric Field

There are several ways to sketch field lines. If you are working with pencil and paper, the procedure is to start a field line at a positive charge and continue the curve, always keeping the **E**-field tangent to the field line, making calculations of the **E**-field as necessary. This procedure is arduous, even for simple distributions of charge. At the other extreme there are sophisticated computer programs that make detailed maps of the electric field. We choose a middle of the road approach, keeping both the concepts and the programs quite simple.

Think of the graphics window as a coordinate system. N charges are placed at different points on the coordinate system. Using the equations given above, the x and y components of the electric field due to each charge may be calculated for any point, (x, y), on the coordinate system. These components are added to find the x and y components of the resultant **E**-field, and this vector may be drawn on the screen. Suppose that the resultant has components E_x and E_y. At the point with coordinates (x, y) we use the line-drawing capability of the programming language to draw a line between the points

$$(x - \frac{1}{2}k\frac{E_x}{E}, y - \frac{1}{2}k\frac{E_y}{E}) \text{ and } (x + \frac{1}{2}k\frac{E_x}{E}, y + \frac{1}{2}k\frac{E_y}{E}).$$

What are the properties of this line segment? It is centered on the point (x, y), it has the same direction as the **E**-field at point (x, y), and its length is proportional to E. Thus the segment

is much like the vector representing **E**. The quantity k is a constant that determines the length of the line segment that represents the vector **E**. We choose k small enough to keep the line segment on the screen, and large enough to see it, a minimum of several pixels.

Thus we can draw a vector representing the **E**-field for any point on the screen. If we did this for each point on the screen, the result would be chaos. Instead, we think of the screen as an *array* of small *regions*, we calculate the **E**-field components at the center of each region, and in each region we plot a scaled version of **E**. The result will look like an array of compasses placed in a magnetic field, but ours are "electric compasses." The procedure is outlined in Algorithm 6.1 and one example, for two unlike charges, is shown in Figure 6.3. Of course, there is a problem if the distance between a point and a charge is so small that we get an overflow or a divide-by-zero error. This problem is solved by eliminating from consideration all points in the region of the array in which a charge lies.

You should also realize that our electric field map gives no information about the magnitude of the electric field, only its direction. To draw actual field lines, pick a positive charge and let the number of field lines that originate on it be proportional to its charge. Sketch these lines on the electric field map, always keeping the **E**-field vectors tangent to the field line. If you continue this process for all the charges, the density of the lines will give you some idea of the magnitude of the field, but remember that almost all two-dimensional pictures of three-dimensional fields are at best suggestive of the magnitude of **E** at a point.

ALGORITHM 6.1 / Graphing the Electric Field Due to N Point Charges

STEP 1. Provide the positions and magnitudes of the N charges, x_i, y_i, Q_i, where i varies from 1 to N.

STEP 2. Choose the domain and range of the graph, $[x_{min}, x_{max}]$ and $[y_{min}, y_{max}]$, respectively. This defines the area in which we will show the electric field.

STEP 3. Choose m, where m x m is the number of small regions that subdivide the area. We kept m between 15 and 40. A large m produces lengthy calculations and small vectors, a small m results in a loss of resolution.

STEP 4. Calculate the length and width of a region:

$$dx = \frac{x_{max} - x_{min}}{m},$$

and

$$dy = \frac{y_{max} - y_{min}}{m}.$$

STEP 5. Let (j, k) be a pair of whole number indices that identify the location of the region on the screen. Let j be the column number and let k be the row number. Both indices are zero in the upper-left-hand corner of the screen.

STEP 6. Begin a loop over k. Start with $k = 1$ and loop m times.

STEP 7. Calculate the y coordinate of the region:

$$y = y_{min} + (k - \frac{1}{2})dy.$$

STEP 8. Begin a loop over j.

STEP 9. Calculate the x coordinate of the region:

$$x = x_{min} + (j - \frac{1}{2})dx.$$

STEP 10. Begin loop over the N charges.

STEP 11. Calculate r, the distance of the point (x, y) from the charge Q_i at point (x_i, y_i):

$$r = \sqrt{(x - x_i)^2 + (y - y_i)^2}.$$

STEP 12. If

$$r \geq \sqrt{dx^2 + dy^2}.$$

then calculate the components of the **E**-field in STEPS 13 through 16 and continue the algorithm with STEP 18. Otherwise proceed directly to STEP 17.

STEP 13. Calculate E_x from the equation

$$E_x = kQ\frac{x_d}{(x_d^2 + y_d^2)^{\frac{3}{2}}}.$$

STEP 14. Update the sum of the x-components:

$$E_{rx} \leftarrow E_{rx} + E_x.$$

STEP 15. Calculate E_y from the equation

$$E_y = kQ\frac{y_d}{(x_d^2 + y_d^2)^{\frac{3}{2}}}.$$

STEP 16. Keep a running sum of the y components:

$$E_{ry} \leftarrow E_{ry} + E_y.$$

STEP 17. Set a flag (FLAG = 1), informing the computer not to calculate the field.

STEP 18. End of the IF in STEP 12.

STEP 19. End of STEP 10 loop.

STEP 20. If FLAG = 0, calculate the components, v_x and v_y, of the vector to be drawn:

$$v_x = \frac{E_{rx}}{\sqrt{E_{rx}^2 + E_{ry}^2}}dx \quad \text{and} \quad v_y = \frac{E_{ry}}{\sqrt{E_{rx}^2 + E_{ry}^2}}dy,$$

and plot a line between

$$(x-\frac{1}{2}v_x, y-\frac{1}{2}v_y) \text{ and } (x+\frac{1}{2}v_x, y+\frac{1}{2}v_y).$$

STEP 21. If FLAG = 1, reset the flag to zero (FLAG = 0).
STEP 22. Let

$$j \leftarrow j+1.$$

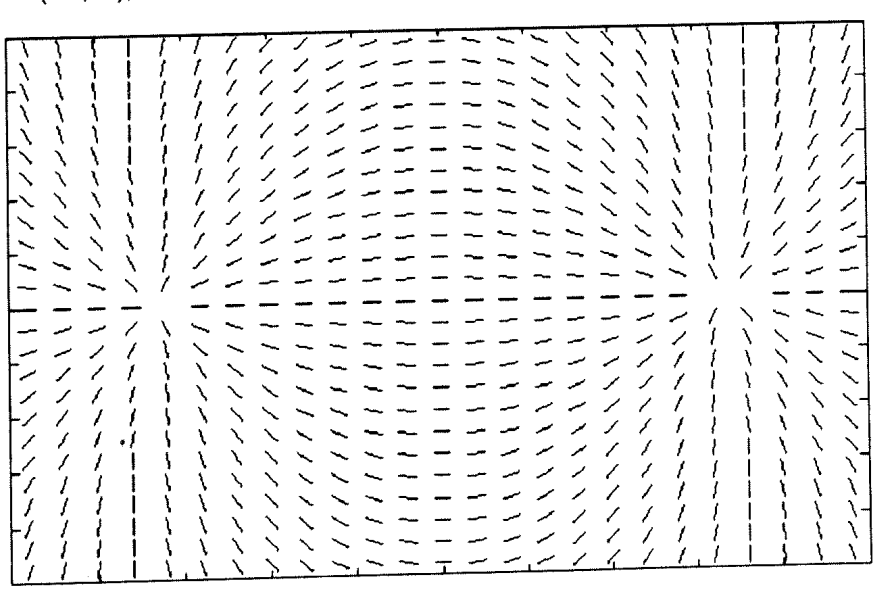
STEP 23. End of STEP 8 loop over j.
STEP 24. Let

$$k \leftarrow k+1.$$

STEP 25. End of STEP 6 loop over k.
STEP 26. Quit.

PROBLEM 6.1 Graphing Electric Fields • Write a program to implement Algorithm 6.1. Begin by using the parameters in Figure 6.3. Experiment with different distributions of charges, for example a line of 21 equal charges on the x-axis at unit intervals from −10 to 10, or two parallel rows of 21 equal charges. Compare the **E**-fields of these two charge distributions to the **E**-field in the vicinity of a long charged rod and the **E**-field between two oppositely charged parallel plates.

FIGURE 6.3 • The Electric Field in the Vicinity of Two Unlike Charges. A 50 μC charge is at (−2, 0), and a −50 μC charge is at (2,0). *Dom* = [−3, 3] m and *ran* = [−2, 2] m.



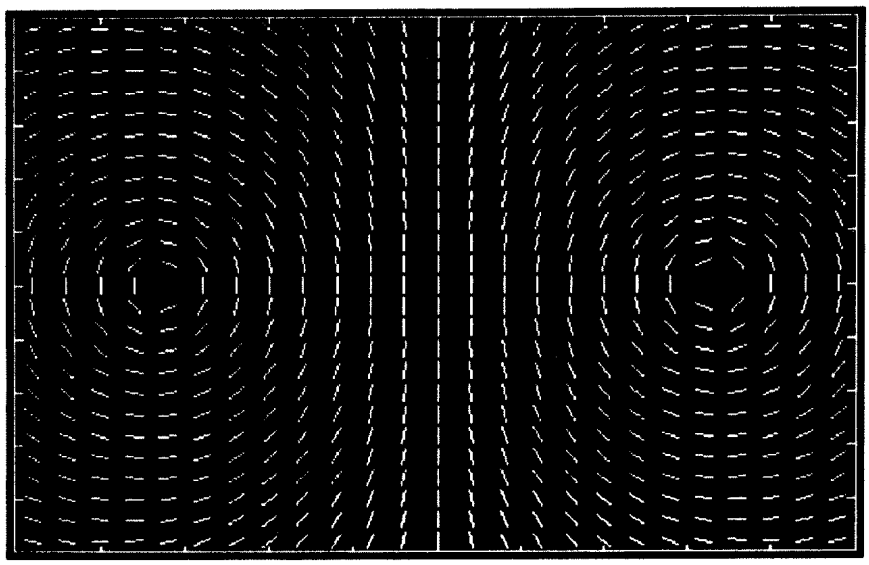


FIGURE 6.4 • Equipotential Line Segments in the Same Region as Figure 6.3.

6.3 Graphing Equipotential Curves

6.3.1 What Are Equipotential Curves?

Potential difference is another concept that is encountered almost immediately in the study of electrical phenomena. The potential difference between two points, a and b, is defined as the work per unit charge that you must do to move the charge from a to b. Thus

$$V_b - V_a = \frac{1}{q}\int_a^b \mathbf{F} \cdot \mathbf{ds},$$

where **F** is the force you must apply. Since this force is opposite to the force produced by the electric field, we can express the potential difference in another way, namely

$$V_b - V_a = -\int_a^b \mathbf{E} \cdot \mathbf{ds}.$$

If the **E**-field is known, then this equation may be used to find the potential difference $V_b - V_a$. This equation also tells us that if there is an **E**-field in a region, then there will be potential differences in that region, and conversely, that a potential difference always implies an **E**-field.

We know the **E**-field in the vicinity of a single point charge Q. If we apply the previous equation to calculate the potential difference between two points in the same region, we find

$$V_b - V_a = kQ\left(\frac{1}{r_b} - \frac{1}{r_a}\right),$$

where r_b and r_a are distances measured radially outward from Q to points a and b. Just as we did for gravitational potential energy, we may take the zero for potential wherever we wish.

We choose to let $V_a = 0$ in the limit as $r_a \to \infty$. This gives

$$V = k\frac{Q}{r},$$

where we have dropped the b subscript.

Unlike the electric field, potential is a scalar. To find the potential at a point due to a set of n point charges, we merely add, algebraically, the potential of the individual charges. Thus

$$V = k\sum_{i=1}^{i=n}\frac{Q_i}{r_i} = k(\frac{Q_1}{r_1} + \frac{Q_2}{r_2} + \ldots + \frac{Q_n}{r_n}).$$

It is important to realize that the *electrical potential energy* of a charge q is Vq. For example, in the planetary model of the hydrogen atom, a two-charge system with an electron orbiting a proton, the potential energy U_e of the electron is simply

$$U_e = k\frac{Q}{r}q = -k\frac{e^2}{r},$$

where $e = 1.602 \times 10^{-19}$ C.

Suppose that in a particular region of space there is an electric field **E**. Then we know there are also potential differences in that region. It is useful to think of the set of points that have the same potential as an *equipotential surface*. Such surfaces are difficult to imagine unless the charge configuration is very simple. For example, in the vicinity of a point charge the equipotential surfaces are spheres. In a region where the electric field is constant, the equipotential surfaces are planes. It is simpler to deal with two dimensions, in which case we are dealing with *equipotential curves*. An equipotential curve is the set of all points on the graph that are at the same potential. In two dimensions the equipotential surfaces we just mentioned become circles and straight lines, respectively.

If you hike, backpack, or camp, you may be familiar with a simple gravitational analog of equipotential curves, namely the contours of constant elevation on a topographical map. The contour curves are gravitational equipotential curves. The elevation differences give the hiker an estimate of the energy required to climb a given hill, and the steepness of the hill is easily inferred from the spacing of the gravitational equipotential curves. If the curves are close, the hill is steep. A small mass sliding around on these (frictionless) hills is the analog of a free charge in a region where there is an **E**-field.

6.3.2 A Simple Scheme for Visualizing Equipotential Curves

The computer is a useful tool to deal with more complex charge distributions. Figure 6.4 is a graph of segments of equipotential curves for the field shown in Figure 6.3. A minor modification of Algorithm 6.1 allows us to make the diagram in Figure 6.4. The electric field **E** and the potential difference are related by the integral expression

$$V_b - V_a = -\int_a^b \mathbf{E}\cdot d\mathbf{s}.$$

If **E** is *perpendicular* to **ds**, then the dot product is zero. Thus points on a tiny line segment perpendicular to **E** are at the same potential, and we may use this fact to draw equipotential line segments in each of the m-by-m regions described above. The following result is drawn from analytic geometry: tangent lines and normal lines have slopes that are the negative reciprocals of each other. Thus if the **E** line-segment has x and y components E_x and E_y, respectively, with slope E_y/E_x, then the equipotential line-segment has x and y components $-E_y$ and E_x, respectively, with slope $-E_x/E_y$. Sketch a picture to convince yourself that this is true. This conclusion was used to produce the equipotential line segments in Figure 6.4. Compare the equipotential segments with the field segments of Figure 6.3. Thus a very simple modification of Algorithm 6.1 may be used to make graphs of equipotential lines.

PROBLEM 6.2 Graphing Equipotential Lines • Modify the program written in Problem 6.1 to graph equipotential line segments rather than **E**-field segments. Use the parameters in Figure 6.3. Repeat the calculations for the two charge distributions mentioned in Problem 6.1.

6.3.3 A More Sophisticated Program for Drawing Equipotential Curves

A useful approach to plotting equipotential curves is an interactive one in which the programmer and the computer both play a role. This idea is described in more detail in Algorithm 6.2, which implements the following operations. N charges are installed on the coordinate system on the screen of the video monitor. The computer program loops, calculating and displaying the potential at the point at which the cursor is pointed, either with cursor keys or a mouse. When the user moves the cursor to find a potential at which an equipotential curve is desired, the mouse is clicked (or a key depressed), and the computer program enters a new loop. In this latter loop the electric field **E** is calculated, and a very small line segment is drawn perpendicular to the **E**-field, to a new equipotential point. A succession of small line segments drawn in this way produce an equipotential curve. The program repeats this process over and over, drawing the curve as it loops. It stops drawing the curve if the curve goes out of the domain or the range of the graph, or until the mouse is clicked (or a key depressed). If the space bar is pressed at this point the program stops; otherwise it loops back to do another equipotential line. The figure at the beginning of this chapter was drawn in this manner, where we had six equal charges on a line. Algorithm 6.2 describes the procedure more concisely.

ALGORITHM 6.2 / An Interactive Approach to Plotting Equipotential Curves

STEP 1. Position a set of N charges on a coordinate system. (Refer to Algorithm 6.1.)
STEP 2. Loop until the space bar is pressed.
  STEP 3. Loop until mouse is clicked or key is pressed. Record the coordinates, x, y, of this point.
    STEP 4. Position the cursor (with keys or mouse) on the coordinate system.
    STEP 5. Calculate and display the potential at this point.

STEP 6. End of STEP 3 loop.
STEP 7. Loop until the point (x, y) is not within the domain and range of the graph, or until a mouse or key is pressed.
STEP 8. Calculate the x and y components of the **E**-field at the point (x, y). (See Algorithm 6.1.)
STEP 9. Graph a very small displacement, **ds**, *normal* to **E**. The components of **ds** are

$$ds_x = -\frac{E_y}{E} ds \quad \text{and} \quad ds_y = \frac{E_x}{E} ds,$$

where ds is the magnitude of **ds**. Like "h," ds must be "sufficiently small" to make a precise graph and "sufficiently large" to sketch the curve quickly. Experiment with various values of ds.
STEP 10. Draw a line from the point (x, y) to the new point with coordinates:

$$x_n = x + ds_x \quad \text{and} \quad y_n = y + ds_y.$$

STEP 11. Update the point (x, y):

$$x \leftarrow x + ds_x \quad \text{and} \quad y \leftarrow y + ds_y.$$

STEP 12. End of STEP 7 loop.
STEP 13. End of the STEP 2 loop.
STEP 14. Quit.

One option you may wish to include is a step-size ds that depends on the magnitude of the electric field. This will make the plotting go more rapidly in regions where the field is small, and more slowly, giving greater precision, in regions where the field is large. We choose $ds = ran/1000$, but with this new modification you might choose

$$ds = \frac{c}{\sqrt{E_x^2 + E_y^2}},$$

where c is a constant chosen to give good results. If equipotential circles close, then you have chosen c small enough.

PROBLEM 6.3 Interactive Equipotential-Curve Sketching • Write a program to implement Algorithm 6.2. Use it to graph equipotential lines for the charge distributions described in Problem 6.1. See Program 6.4.2. What are the effects of changing the size of the displacement, **ds**?

PROBLEM 6.4 Graphing Electric Field Lines (Challenging) • Modify the program of Problem 6.3 to draw field lines. Once again, study the charge distributions described in Problem 6.1. Identify the strengths and weaknesses of this algorithm.

PROBLEM 6.5 Calculating E • Write a program to calculate the electric field **E** at a point (x, y) in the vicinity of several point charges. Input for the program should include the number of charges, the magnitude of each charge, the coordinates of each charge, and the coordinates of the position where the electric field is to be calculated. Remember, **E** is a vector quantity. Your program should have, as output, the magnitude of **E**, its x and y components, and the angle **E** makes with the $+x$-axis.

PROBLEM 6.6 The Electric Field and Graphics (Challenging) • Modify the program of Problem 6.5 to draw the **E**-vector at any point on the screen where the mouse or the cursor is set.

6.4 Programs

PROGRAM 6.4.1 / Plotting the E-Field of N Charges

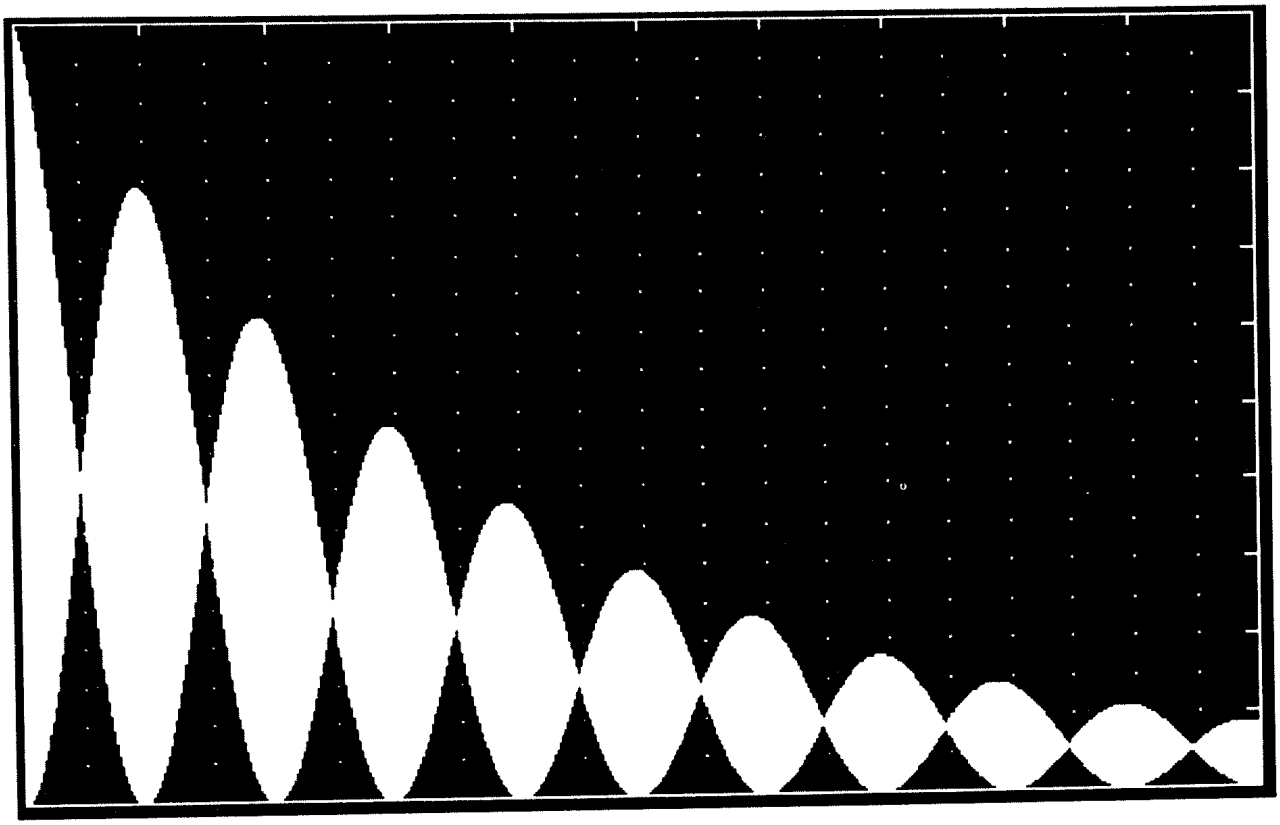
```
! Try Q's of 50 and - 50 at (-2, 0) and (2, 0). Use range [-2,2].
! Algorithm 6.1 and Problem 6.1
DIM Q(10), XC(10), YC(10)                                  ! Store charge data.
LIBRARY "Comphys"
PRINT "How many charges?"
INPUT N                                                    ! Number of charges.
FOR I = 1 to N
    PRINT "Type the size of the charge, [-10, 10] Coulombs,"
    PRINT "and its x and y coordinates. x in [xmin, xmax], y in [ymin, ymax]"
    PRINT "separated by commas."
    INPUT Q(I), XC(I), YC(I)
    PRINT q(i),xc(i),yc(i)
NEXT I
INPUT prompt "What mesh size (m by m)?": m                 ! How many regions?
CALL openB (xmin, xmax, ymin, ymax, wxmin, wxmax, wymin, wymax)
LET dy = (ymax - ymin)/m                                   ! Height of a region.
LET dx = (xmax - xmin)/m                                   ! Width of a region.
FOR k = 1 to m                                             ! Loop over columns of the
                                                             array.
    LET y = ymin + dy*(k -1/2)
    FOR j = 1 to m                                         ! Loop over rows of the array.
        LET x = xmin + dx*(j-1/2)
        LET Ex = 0
        LET Ey = 0
        FOR i = 1 to N                                     ! Find fields due to each of the
                                                             N charges.
            LET xd = x - XC(i)                             ! x distance to charge
            LET yd = y - YC(i)                             ! y distance to charge.
            LET R = sqr(xd*xd + yd*yd)
            IF R > dy then                                 ! Don't get too close.
               LET E = Q(i)/(R*R*R)
               LET Ex = Ex + E*xd                          ! Sum field components.
               LET Ey = Ey + E*yd
```

```
            ELSE
               LET flg = 1
            END IF
         NEXT i
         IF Flg = 0 then
            LET E = sqr(Ex*Ex + Ey*Ey)
            LET xcomp = Ex/E*dy*.75                          ! Graph tiny vectors.
            LET ycomp = Ey/E*dy*.75
            PLOT x-xcomp/2,y-ycomp/2;x+xcomp/2, y+ycomp/2
         ELSE
            LET flg = 0
         END IF
      NEXT j
   NEXT k
   END
```

PROGRAM 6.4.2 / Interactive Equipotential Line Plotter

```
! Algorithm 6.2 and Problem 6.2.
! Try two 5 uC plus charges at (-2,0) and (2, 0) with range [-2, 2] (program
! picks domain.
DIM Q(10), XC(10), YC(10)                                    ! Arrays for charge and
LIBRARY "Comphys"                                            ! positions.
!......................
! Input the values of the charges and their coordinates.
PRINT "How many charges?"                                    ! Get the charges and
INPUT N                                                      ! their positions.
FOR I = 1 to N
    PRINT "Type the size of the charge in microcoulombs "
    PRINT "and it's x and y coordinates. x in [xmin, xmax], y in [ymin, ymax]"
    PRINT "separated by commas."
    INPUT Q(I), XC(I), YC(I)
    PRINT q(i),xc(i),yc(i)                                   ! Check input.
NEXT I
CALL openB (xmin, xmax, ymin, ymax, wxmin, wxmax, wymin, wymax)
LET ds = (ymax-ymin)/1000                                    ! Step size. Try different
                                                               values.
!.......................
! Calculate (and display) the potential V at the position of the cursor.
DO                                                           ! This is the big loop.
   DO                                                        ! This is an inside loop
      GET MOUSE x,y,s                                        ! to get potential at point
      LET V = 0                                              ! (x, y).
      FOR i = 1 to N                                         ! Find potential due to each of
                                                               the charges.
         LET xd = x - XC(i)                                  ! x distance to charge
         LET yd = y - YC(i)                                  ! y distance to charge.
         LET R = sqr(xd*xd + yd*yd)
         IF R > (ymax-ymin)/100 then                         ! Don't get too close.
```

```
            LET V = V + Q(i)/R
         ELSE
            LET V = 0
         END IF
      NEXT i
      SET CURSOR 1,5                                        ! Print the potential.
      PRINT using "Potential = ###.##":V
   LOOP until s = 2                                         ! Loop until mouse is clicked,
!.........................
! Calculate the components of the electric field at the point (x, y)
   DO                                                       ! then go to this loop
      LET flg = 0                                           ! to calculate step size
      LET Ex = 0                                            ! and direction.
      LET Ey = 0
      FOR i = 1 to N                                        ! Find field due to each of the
                                                              charges.
         LET xd = x - XC(i)                                 ! x distance to charge
         LET yd = y - YC(i)                                 ! y distance to charge.
         LET R = sqr(xd*xd + yd*yd)
         IF R > (ymax-ymin)/100 then
            LET E = Q(i)/(R*R*R)
            LET Ex = Ex + E*xd                              ! Calculate field components.
            LET Ey = Ey + E*yd
         ELSE
            LET flg = 1                                     ! Too close! set flag.
         END IF
      NEXT i
      IF flg = 0 then
         LET E = sqr(Ex*Ex + Ey*Ey)                         ! Calculate steps
!.........................
! Calculate the tiny displacement normal to E, and its endpoint (x, y).
         LET xcomp = -Ey/E*ds                               ! in x and y directions.
         LET ycomp = Ex/E*ds
         PLOT x,y;x+xcomp,y+ycomp                           ! Plot the segment.
         LET x = x + xcomp                                  ! Increment the coordinates.
         LET y = y + ycomp
      ELSE
      END IF
      IF x > xmax or x < xmin or y > ymax or y < ymin or flg = 1 then
         EXIT DO                                            ! Quit for a variety of reasons.
      ELSE
      END IF
      GET MOUSE xx, yy, s                                   ! Check mouse.
      IF s = 2 then EXIT DO                                 ! Out of this loop.
   LOOP                                                     ! Draw more of the curve.
!.......................
! Escape from the big loop.
   IF key input then                                        ! Into this one. Press space bar
      GET KEY z                                             ! before pressing mouse.
      IF z = 32 then EXIT DO
   END IF
LOOP
END
```



Oscillations in an *LRC* Circuit

7 / *RC*, *LR*, and *LRC* Circuits

7.1 Introduction

A review of the following topics may be helpful in this chapter:

- Electric current, potential difference, resistance
- Direct-current circuits, Ohm's Law, and Kirchhoff's Laws
- Capacitance and, later in the chapter, self-inductance

In physics textbooks the topic of self-inductance does not immediately follow the topic of capacitance; therefore you may wish to delay Section 7.3 of this chapter until you have studied the appropriate sections in your textbook. In some textbooks the subject of *LRC* circuits may also be postponed until the chapter on ac circuits and electromagnetic radiation. As a consequence, you may also choose to delay the study of Section 7.4. We gather these topics into one chapter because they are similar circuits, and the techniques of analysis are almost identical for all three circuits. One of the things you should notice in this chapter is

that the numerical techniques we developed as we studied mechanical systems are just as applicable to electrical systems. We will be solving the same kinds of equations. Many of the systems studied in this chapter may also be studied with analytic techniques, but the numerical approach is easier to understand than the analytic techniques.

7.2 The *RC* Circuit

7.2.1 Charging and Discharging a Capacitor

An *RC* circuit consists of a battery having a constant voltage V across its terminals in series with a resistor with resistance R and a capacitor whose capacitance is C. Such a circuit is shown in Figure 7.1. At $t = 0$ the switched is closed. We label the current in the circuit $i(t)$, and the charge on the capacitor, $q(t)$. How does the current in the circuit change with time thereafter? How does the charge on C vary with time?

Kirchhoff's Loop Rule requires that the sum of the potential changes around a closed loop be zero. With the switch closed, the loop rule gives

$$V - iR - \frac{q}{C} = 0.$$

The voltage drop across the resistor is, according to Ohm's Law, iR, and the voltage drop across the capacitor is q/C.

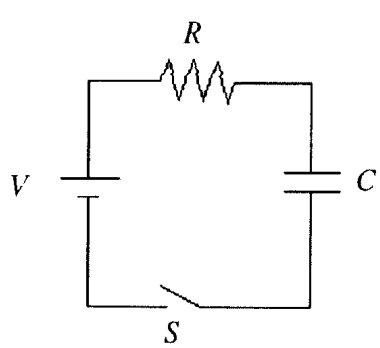
The charge $q(t)$ and the current $i(t)$ are related. The charge on the capacitor increases because there are charges flowing through the wire to C. Specifically,

$$\frac{dq}{dt} = i.$$

Clearly, the larger the current, the more rapidly q increases. That is the significance of this equation. Substituting this relationship into the first equation and rearranging terms gives

$$\frac{dq}{dt} = \frac{V}{R} - \frac{q}{RC}.$$

FIGURE 7.1 • An *RC* Circuit.



In symbolizing the derivative it is convenient to use both the dq/dt notation and the prime notation, $q'(t)$. Thus the last equation may also be written

$$q'(t) = \frac{V}{R} - \frac{q(t)}{RC}.$$

At $t = 0$ the switch is closed. Prior to the switch closure, the current i was zero, so no charge accumulated on the capacitor. Assuming that there was no charge at $t = 0$, our initial conditions are

$$\text{at } t = 0,\ q = 0, \text{ and } i = \frac{V}{R}.$$

We would like to pause for a moment to point out the similarities between the situation we are studying and the problem addressed in Chapter 1. Refer to Table 7.1 where it should be clear that i is the analog of a and q is the analog of v. Do you think there we will find a terminal q just as we found a terminal v?

TABLE 7.1 • A Comparison of the Falling Object Problem with the *RC* Circuit Problem.

Speed of a Falling Object	**Current in an RC Circuit**
Definition of Acceleration	Definition of Current
$$a(t) = v'(t) = \lim_{h\to 0} \frac{v(t+h) - v(t)}{h}.$$	$$i(t) = q'(t) = \lim_{h\to 0} \frac{q(t+h) - q(t)}{h}.$$
To solve a problem, find a from Newton's Second Law. Thus	To solve a problem, find i from Kirchhoff's Loop Rule. Thus
$$a = g - \frac{1}{2} C\rho \frac{A}{m} v^2.$$	$$i = \frac{V}{R} - \frac{q}{RC}.$$
Use the derivative definition of acceleration to make the approximations	Use the derivative definition of current to make the approximations
$$a(t) \cong \frac{v(t+h) - v(t)}{h},$$	$$i(t) \cong \frac{q(t+h) - q(t)}{h},$$
and	and
$$v(t+h) \cong v(t) + ha(t).$$	$$q(t+h) \cong q(t) + hi(t).$$
With Euler's method, use the last approximation to solve the problem in time steps of length h. The initial conditions are:	With Euler's method, use the last approximation to solve the problem in time steps of length h. The initial conditions are:
at $t = 0$, $v = 0$, and $a = g$.	at $t = 0$, $q = 0$, and $i = V/R$.

Our intent is to solve the RC circuit problem in the same way we handled the system of the falling object. The definition of the derivative is

$$q'(t) = \frac{dq}{dt} = \lim_{h \to 0} \frac{q(t+h) - q(t)}{h} \cong \frac{q(t+h) - q(t)}{h}.$$

The approximation indicated by $\cong$ is a good one for sufficiently small values of h; therefore we replace $\cong$ with $=$. Rewrite the approximation this way:

$$q(t+h) = q(t) + hq'(t).$$

Earlier we showed that

$$q'(t) = \frac{V}{R} - \frac{q(t)}{RC}.$$

By writing the charge as $q(t)$, we explicitly indicate that it is a function of time. You are reminded once again that $i(t) = q'(t)$.

The last two equations allow us to find a numerical solution using Euler's method. Let's change to subscript notation and proceed to outline the numerical solution in steps:

$$q_1 = q_0 + hi_0 \quad \text{or} \quad q_1 = q_0 + hq'_0,$$

$$q_2 = q_1 + hi_1 \quad \text{or} \quad q_2 = q_1 + hq'_1,$$

$$q_3 = q_2 + hi_2 \quad \text{or} \quad q_3 = q_2 + hq'_2,$$

which when generalized is

$$q_{n+1} = q_n + hi_n \quad \text{or} \quad q_{n+1} = q_n + hq'_n.$$

For reference purposes, we also give the general equation using function notation. Thus

$$q[(n+1)h] = q(nh) + ha(nh).$$

A procedure to calculate and graph $q(t)$ for the RC circuit is described with Algorithm 7.1, where, once again, we have chosen to emphasize the fact that $i(t) = q'(t)$ by using both in the formulas.

ALGORITHM 7.1 / The Charge on a Capacitor in an *RC* Circuit

STEP 1. Specify the values of R and C. Choose a value for the time interval h.
STEP 2. Choose the initial conditions: $i_0 = V/R$, and $q_0 = 0$.
STEP 3. Loop until $t > t_{max}$.

STEP 4. Calculate $q'(t)$ or $i(t)$ from the formulas

$$i(t) = q'(t) = \frac{1}{R}\left[V(t) - \frac{q(t)}{C}\right].$$

STEP 5. Plot a point on the graph of $q(t)$ and/or $i(t)$.
STEP 6. Update the charge:

$$q \leftarrow q + hi \quad \text{or} \quad q \leftarrow q + hq'.$$

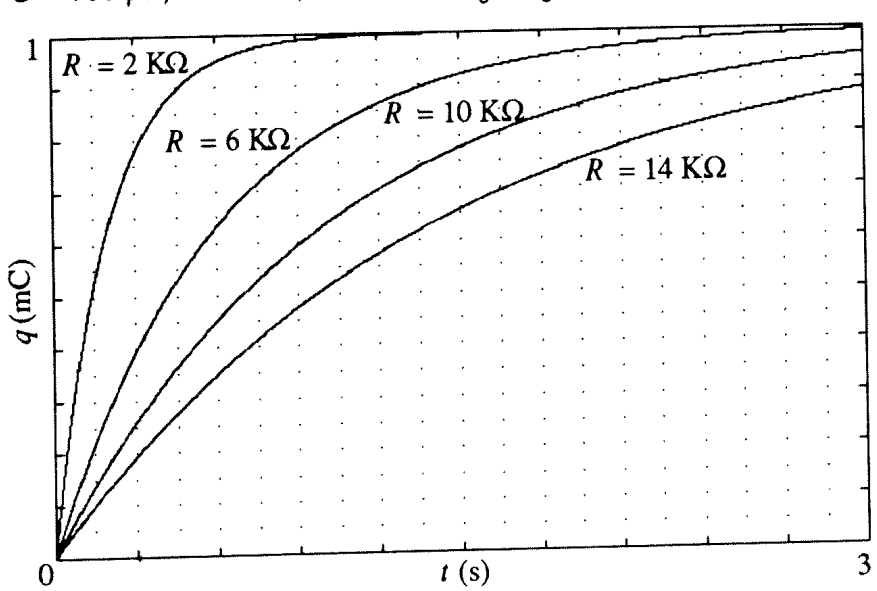
STEP 7. Increment the time:

$$t \leftarrow t + h.$$

STEP 8. End of the STEP 3 loop.
STEP 9. Quit.

PROBLEM 7.1 Finding $q(t)$ for an *RC* Circuit • Write a program to implement Algorithm 7.1 and make a graph of $q(t)$. Choose $V = 10$ V, $R = 1000\ \Omega$, $C = 100\ \mu$F, $h = 0.5$ ms, $i_0 = q_0 = 0$, $dom = [0, 0.5]$ s, and $ran = [0, 0.001]$ C. Modify your program to make a family of curves for different values of R. Our results are shown in Figure 7.2. Repeat this process with C as the parameter. How do the curves change as C increases? Can you give a conceptual explanation for this behavior? Does q approach some terminal value?

FIGURE 7.2 • A Family of Graphs of $q(t)$ for Various Values of R. Other parameters were $C = 100\ \mu$F, $V = 10$ V, $h = 0.5$ ms, $i_0 = q_0 = 0$, $dom = [0, 3]$ s, and $ran = [0, 0.001]$ C.



PROBLEM 7.2 Numerical Versus Analytic Solution • Compare your numerical solution for $q(t)$ with the analytic solution,

$$q(t) = VC\left(1 - e^{-\frac{t}{RC}}\right)$$

derived in your textbook. With the help of the computer, construct a graph of this equation and compare it with the graph you obtained in Problem 7.1.

PROBLEM 7.3 Finding $i(t)$ for an RC Circuit • Since

$$i(t) = \frac{dq(t)}{dt} = q'(t),$$

we can see how the current in the circuit varies with time. Add instructions to the program of Problem 7.1 to graph $i(t)$. Scale it so both $q(t)$ and $i(t)$ are graphed. Use the parameters in Problem 7.1. Do the two graphs suggest that $i = dq/dt$? Why does $i(t)$ approach zero for large t? What variable approaches zero in the falling-object problem?

PROBLEM 7.4 Discharging a Capacitor • Algorithm 7.1 may also be used to study the discharge of a capacitor. We take the battery out of the circuit ($V = 0$) leaving only the resistor, the capacitor, and the switch. Assume that the capacitor is charged and at $t = 0$ the switch is closed. These are the initial conditions: $i_0 = 0$, $q_0 = 10^{-3}$ C. The parameters are $R = 1000\ \Omega$, $C = 100\ \mu$F, $h = 1$ ms, $dom = [0, 0.5]$ s, and $ran = [0, 0.001]$ C. Make graphs of $i(t)$ and $q(t)$. Study your graphs and describe the physical principles that cause the current and charge on the capacitor to behave in this way. Describe how the graphs show that $q'(t) = i(t)$. It may be useful to make families of curves, first with R as the parameter, then with C as the parameter.

7.2.2 Energy in the *RC* Circuit (Challenging)

In this section we study the RC circuit from an energy perspective. This perspective gives us a way to check our calculations of $q(t)$ and $i(t)$, for if energy is not conserved at each step we make, then it is very likely that our calculations are incorrect. Studying the RC circuit from an energy perspective also gives fresh insights into the behavior of this circuit. The relationship we will derive from the conservation of energy principle is

$$\frac{1}{2}\frac{q_n^2}{C} + Rh[i_1^2 + i_2^2 + \cdots + i_n^2] = Vh[i_1 + i_2 + \cdots + i_n],$$

where the subscripts identify each step in the solution for $q(t)$. In this expression the left-most term is the energy stored in the capacitor, the next term is the thermal energy of the resistor, and the right-hand side is the energy supplied by the battery. The paragraphs that follow are not simple. If you wish to skip them, start with Problem 7.5, where we summarize the terms in the conservation of energy (COE) equation.

We begin with the COE equation, first seen in Chapter 5. The COE principle says that the change of energy, ΔE, in the system is equal to the heat, Q, added to the system and the work, W, done on the system. In symbols the COE equation is,

$$\Delta E = Q + W.$$

We choose the system to be the resistor and the capacitor. While we allow the resistor to acquire thermal energy, we will assume no heat is transferred in or out of the system while the capacitor is charging. Thus

$$Q = 0.$$

(We could isolate the resistor and the capacitor in a vacuum if necessary, so they are insulated from the surroundings.)

In the charging process the chemical forces in the battery do work on the system. This is the W term in the equation, and the definition of potential difference gives

$$W = \int_0^{q_f} V\, dq,$$

where V is the potential difference maintained across the terminals of the battery, and q_f is the total amount of charge supplied to the resistor-capacitor system. A "u-substitution," namely

$$dq = q'(t)\, dt$$

turns the integral into one over time. Thus

$$W = \int_0^{q_f} V\, dq = \int_0^{t_f} V q'(t) dt = V \int_0^{t_f} i(t) dt,$$

where we have made use of the fact that V is constant and

$$i = q'(t).$$

Next we will deal with the left-hand side of the COE. In the resistor, electrical energy is converted into thermal energy, E_{th}, at the rate

$$P = i^2 R.$$

Thus

$$\Delta E_{th} = \int P(t)\, dt = R \int_0^{t_f} [i(t)]^2\, dt.$$

The change in the electrical potential energy of the charge on the capacitor is

$$\Delta U_c = \int_0^{q_f} V_c(t) dq.$$

Recall that potential difference is potential energy difference per unit charge. Thus you may think of the battery as taking a plus charge from the negative side of the capacitor and raising it through the potential difference V_c, increasing its potential energy by qV_c (quite like lifting

a stone from the ground to increase its gravitational potential energy). Two substitutions will simplify the last integral; they are

$$V_c = \frac{q}{C}$$

and

$$dq = q'(t)dt,$$

giving

$$\Delta U_c = \frac{1}{C}\int_0^{t_f} q(t)q'(t)\,dt = \frac{1}{2C}\left\{[q(t_f)]^2 - [q(0)]^2\right\},$$

independent of the time variation of $q(t)$.

We assemble all of the terms in the COE equation to obtain

$$\frac{1}{2C}\left[[q(t_f)]^2 - \ [q(0)]^2\right] \ + R\int_0^{t_f}[i(t)]^2\,dt = V\int_0^{t_f} i(t)\,dt.$$

We approximate the two integrals numerically with calculated sums, exactly as we did in Chapter 5. With that approximation, the COE equation becomes

$$\frac{1}{2C}[q(t_n)]^2 + Rh\sum_{j=1}^{j=n}[i(t_j)]^2 = Vh\sum_{j=1}^{j=n} i(t_j).$$

We have assumed $q_0 = i_0 = 0$. In this equation n is the number of steps we have made in the solving the differential equation using Euler's method.

This equation looks much less formidable in subscript notation. Thus

$$\frac{1}{2C}q_n^2 + Rh\sum_{j=1}^{j=n} i_j^2 = Vh\sum_{j=1}^{j=n} i_j,$$

where

$$Rh\sum_{j=1}^{j=n} i_j^2 = Rh[i_1^2 + i_2^2 + \cdots + i_n^2],$$

and

$$Vh\sum_{j=1}^{j=n} i_j = Vh[i_1 + i_2 + \cdots + i_n].$$

Each of these sums may be calculated as a running sum. A computer program would read

```
SUM = SUM + I^2
```

for the instruction that performs the "running sum" of the i^2 terms.

PROBLEM 7.5 Energy Changes in an *RC* Circuit • Add instructions to the program you used in Problem 7.1 to calculate the following:

- The energy supplied by the battery (right-hand side of the COE equation)

$$W = Vh\sum_{j=1}^{j=n} i_j = Vh[i_1 + i_2 + \cdots + i_n]$$

- The change (increase) in the electrical potential energy of the charge on the capacitor (assuming $q_0 = 0$)

$$\Delta U_c = \frac{1}{2}\frac{q_n^2}{C}$$

- The internal thermal energy generated in the resistor

$$\Delta E_{th} = Rh\sum_{j=1}^{j=n} i_j^2 = Rh[i_1^2 + i_2^2 + \cdots + i_n^2],$$

where the last two terms represent the change in the system's internal energy, ΔE. If energy is conserved, then these terms are related by the equation

$$\frac{1}{2}\frac{q_n^2}{C} + Rh[i_1^2 + i_2^2 + \cdots + i_n^2] = Vh[i_1 + i_2 + \cdots + i_n].$$

Graph each energy term as a function of time. Let V = 10 V, C = 100 μF, R = 1 kΩ, h = 0.5 ms, $i_0 = q_0 = 0$, *dom* = [0, 0.5] s and *ran* = [0, 0.005] J.

- How does the electrical potential energy compare with the total energy supplied by the battery after the capacitor is almost completely charged ($i \cong 0$)?
- Compare the electrical potential energy with the thermal energy acquired by the resistor after the charging is essentially complete. Does changing the resistance (or the capacitance) affect this result?
- Recall that power is the rate at which energy is acquired (or lost). Using the energy curves just graphed, can you predict how the power supplied (or lost) by each component will vary with time? Modify the program to graph power instead of energy.

7.3 The *LR* Circuit

7.3.1 Current in an *LR* Circuit

An *LR* circuit, consisting of a switch, a battery, a resistor, and an inductor, is shown in Figure 7.3. Our objective is to find how the current varies with time after the switch is closed. We assume that the battery maintains a potential difference of V across its terminals, the resis-

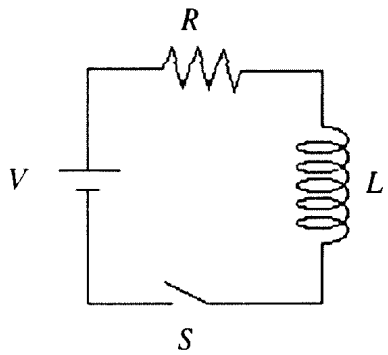


FIGURE 7.3 • The *LR* Circuit.

tance is R, and the self-inductance of the coil is L. Recall that the voltage, V_L, across the inductor is given by the expression

$$V_L = -L\frac{di}{dt}.$$

This relationship gives us a clue about the initial conditions; the current must change smoothly or continuously from zero. If the current changes abruptly from zero to some value i_0 when the switch is closed, then di/dt would be infinite. Nature abhors the infinite as much as a vacuum. In short, no pun intended, the current will increase smoothly from an initial value of zero. Thus $i_0 = 0$.

We apply Kirchhoff's Loop Rule to the circuit when the switch is closed. This gives

$$V - iR - L\frac{di}{dt} = 0.$$

We rearrange this equation to have di/dt on the left-hand side. Thus

$$i'(t) = \frac{di}{dt} = \frac{V}{L} - i\frac{R}{L}.$$

For purposes of comparison, here is Kirchhoff's Loop Rule for the RC circuit:

$$q'(t) = \frac{dq}{dt} = \frac{V}{R} - \frac{q(t)}{RC}.$$

It should be clear that the algorithms and programs we developed for the RC circuit in Section 7.2 may, with some parameter changes, also be used for the LR circuit. We could also use the algorithms for the falling-object problem of Chapter 1.

To obtain a numerical solution for $i(t)$, begin by finding i'_0 given the initial condition i_0. Next find i_1 from the expression

$$i_1 = i_0 + hi'_0.$$

Continue the solution with these expressions:

$$i_2 = i_1 + hi_1',$$

$$i_3 = i_2 + hi_2',$$

and so on, until you have the numerical solution for $i(t)$. The general formula for this sequence of calculations is

$$i_{n+1} = i_n + hi_n',$$

with

$$i_n' = \frac{V}{L} - \frac{R}{L} i_n.$$

Expressed in function notation, the general form is

$$i[(n+1)h] = i(nh) + hi'(nh).$$

Other time-varying quantities such as the voltage across the resistor and the voltage across the inductor may be calculated from the information generated in the solution. Thus

$$V_R = iR,$$

and

$$V_L = -L\frac{di}{dt} = -Li'(t).$$

PROBLEM 7.6 Current in an *LR* Circuit • Modify Algorithm 7.1 to work for the *LR* circuit. Use the algorithm to write a computer program to find a numerical solution for $i(t)$. Find and graph $i(t)$ using the following parameters: $L = 100$ mH, $R = 200\ \Omega$, $V = 10$ V. Choose $h = 0.01$ ms, $dom = [0, 0.005]$ s, and $ran = [0, 0.05]$ A. How does the graph change when R is changed? What effect does the choice of L have on the shape of the curves?

Make an analogy with the *LR* circuit and the falling-object problem of Chapter 1. What variable in the *LR* circuit is analogous to the velocity of the falling object? What is the analog of acceleration? Does any variable have a terminal value? What parameter of the *LR* circuit is analogous to the mass of the falling object? Is the drag coefficient an analog of R?

PROBLEM 7.7 Voltage Across an Inductor in an *LR* Circuit • The voltage, V_L, across the inductor is given with the following equation:

$$V_L = -L\frac{di}{dt} = -Li'(t).$$

Graph the voltage across the inductor as a function of time. Use the parameters in Problem 7.6. You need only to change the range to [0, 10] V. Compare the graphs obtained in Problems

7.6 and 7.7. Do you see the derivative relationships? If you do, then you should be able to answer the following question. Assume that the current in an *LR* circuit varies like the "triangle" function, first starting at zero and increasing *linearly* with time to some peak value, i_p, then decreasing *linearly* back to zero. How will the voltage across L vary with time?

PROBLEM 7.8 Analytic Solution for the *LR* Circuit • Compare the numerical solution of Problem 7.6 with the analytic solution

$$i(t) = \frac{V}{R}(1 - e^{-\frac{R}{L}t}).$$

Graph this equation and then compare it with the graph from Problem 7.6.

7.3.2 Power in an *LR* Circuit

The conservation of energy (COE) equation is

$$\Delta E = Q + W,$$

where ΔE is the change in the internal energy of the system, Q is the heat gained (or lost if Q is negative) by the system, and W is the work done on (or "by" if W is negative) the system. We choose our system to be the resistor and the inductor, while the battery is an outside agent, doing work on the system.

ΔE is made up of the change in the thermal energy of the resistor and the change in the magnetic potential energy associated with the inductor. If we take the derivative of this energy relationship, we obtain a power relationship, namely

$$P_R + P_L = P_V,$$

where the terms are (check your physics textbook):

- The power dissipated in the resistor,

$$P_R = i^2 R$$

- The power going into the magnetic field,

$$P_L = Li\frac{di}{dt}$$

- The power supplied by the battery,

$$P_V = Vi$$

PROBLEM 7.9 Power Relationships in an *LR* Circuit • Modify the program written for Problem 7.6 to graph each of the preceding three power terms as a function of time. Use $L = 0.1$ H, $R = 200\,\Omega$, $V = 10$ V, $h = 0.002$ s, *dom* = [0, 0.002] s, and *ran* = [0, 0.5] W. Our results

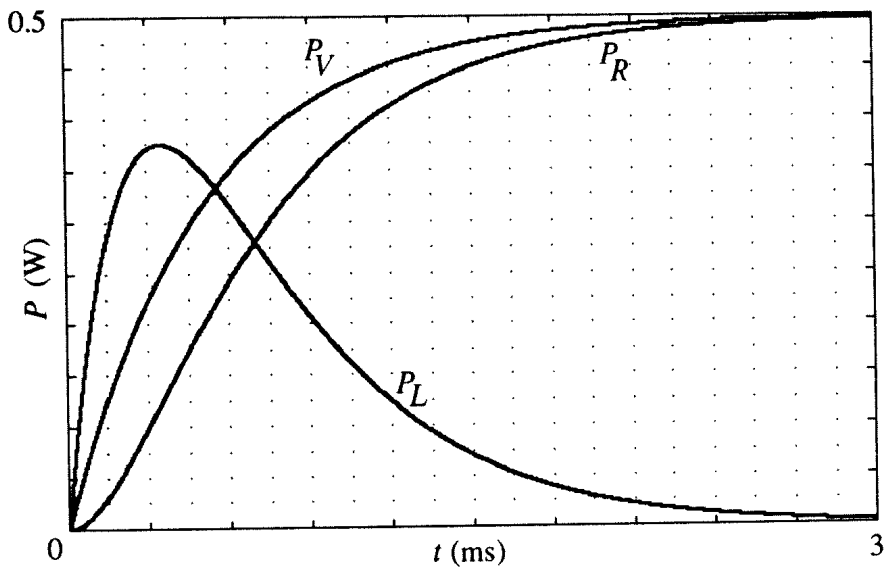


FIGURE 7.4 • Power Changes in an *LR* Circuit. P_V is the power supplied by the batter, P_R is the power dissipated in the resistor, and P_L is the power being converted to magnetic potential energy by the inductor. Parameters: $V = 10$ V, $L = 0.1$ H, $R = 200\ \Omega$, and $h = 0.002$ s, *dom* = [0, 0.003] s, and *ran* = [0, 0.5] W.

are shown in Figure 7.4. The graph of P_L was scaled upward by a factor of three. Why does P_L first increase and then decrease? Is it true that

$$P_R + P_L = P_V\ ?$$

7.3.3 Energy Relationships in an *LR* Circuit (Challenging)

Once again, we begin with the COE equation, which states that the change in energy of the system is equal to the heat gained by the system and the work done on the system by an outside agent. Thus

$$\Delta E = Q + W,$$

where the internal changes in energy are those associated with the thermal energy of the resistor and the magnetic potential energy. We assume $Q = 0$, and W is the work done on the system by the chemical forces in the battery. In Section 7.2.2 we studied the energy changes that take place in an *RC* circuit, concluding that the work done by the battery is

$$W = \int_0^{t_f} Vi(t)\,dt \cong Vh\sum_{j=1}^{j=n} i(t_j) = Vh(i_1 + i_2 + \cdots i_n) = Vh\sum_{j=1}^{j=n} i_j.$$

Once again, we are approximating an integral with a sum.

Part of the ΔE term in the COE equation is the change in the thermal energy of the resistor, assuming the inductor itself has no resistance. In Section 7.2 we found that

$$E_{th} = R\int_0^{t_f} [i(t)]^2 dt \cong Rh\sum_{j=1}^{j=n}\left[i(t_j)\right]^2 = Rh(i_1^2 + i_2^2 + \cdots i_n^2) = Rh\sum_{j=1}^{j=n} i_j^2.$$

The other part of the ΔE term in the COE equation is the (magnetic) energy stored in the vicinity of the inductor. This result is derived in most physics textbooks, so, without proof, this term is

$$\Delta U_L = \frac{1}{2}L\left\{\left[i(t_n)\right]^2 - \left[i(0)\right]^2\right\} = \frac{1}{2}L(i_n^2 - i_0^2).$$

Notice that both function and subscript notation are being used.

Assembling these in the COE equation gives

$$\frac{1}{2}L[i(t_n)]^2 + Rh\sum_{j=1}^{j=n}[i(t_j)]^2 = Vh\sum_{j=1}^{j=n} i(t_j),$$

where we have assumed that $i(0) = 0$. This relationship typically looks less forbidding in subscript notation, namely

$$\frac{1}{2}L(i_n)^2 + Rh(i_1^2 + i_2^2 + \cdots + i_n^2) = Vh(i_1 + i_2 + \cdots + i_n),$$

or, more concisely,

$$\frac{1}{2}L(i_n)^2 + Rh\sum_{j=1}^{j=n} i_j^2 = Vh\sum_{j=1}^{j=n} i_j.$$

PROBLEM 7.10 Energy in an *LR* Circuit • Modify the program of Problem 7.6 to calculate and graph the three energy terms in the COE equation. The terms include the

- Magnetic energy

$$U_L = \tfrac{1}{2}Li_n^2$$

- Internal thermal energy

$$E_{th} = Rh\sum_{j=1}^{j=n} i_j^2 = Rh(i_1^2 + i_2^2 + \cdots + i_n^2)$$

- Energy supplied by the battery

$$W = Vh\sum_{j=1}^{j=n} i_j = Vh(i_1 + i_2 + \cdots + i_n)$$

It should also be true that

$$E_{th} + U_L - W = 0.$$

Verify that this is indeed the case as the solution for $i(t)$ in the *LR* circuit unfolds. Try different values of the step-size h. The choice of parameters for this program is up to you. Starting with previously used parameters is a good idea, and printing some values before graphing is recommended.

7.4 The *LRC* Circuit

Suppose that the switch in the *LR* circuit is closed and current is flowing. What happens when the switch is opened? Opening the switch makes di/dt huge, producing a very large voltage across the inductor, and sparking frequently occurs across the switch contacts. There are several ways of avoiding this problem. One way is to put a capacitor across the switch terminals. The inductor, in the manner of all inductors, "tries" to keep the current constant. But now instead of sparking across the switch contacts, the capacitor is charged. Let us look at this process in more detail. The circuit is shown in Figure 7.5. Before $t = 0$, the switch is closed and a current $i = V/R$ is flowing in the circuit. At $t = 0$, the switch is opened.

Applying Kirchhoff's Loop Rule yields the equation

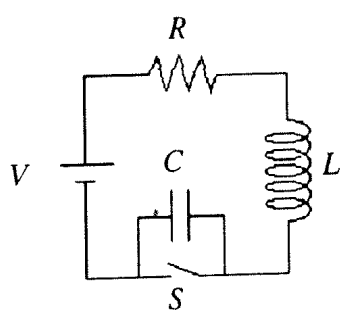
$$V - L\frac{di}{dt} - \frac{q}{C} - Ri = 0.$$

In the case we are describing the initial conditions are $i_0, \neq 0$, but $q_0 = 0$. There was a current in the circuit when the switch was opened, which is the situation that is motivating this investigation.

We know that

$$i = \frac{dq}{dt} = q'(t),$$

FIGURE 7.5 • An *LRC* Circuit.



and

$$\frac{di}{dt} = \frac{d^2q}{dt^2} = q''(t).$$

Making these substitutions into the preceding equation gives

$$q''(t) = -\frac{1}{L}\left[\frac{q}{C} + Rq'(t) - V\right].$$

Just as in the case of Newton's Second Law, Kirchhoff's Loop Rule yields an equation that we may use to find a numerical solution for $q(t)$ and $i(t)$. Having already found the solution to this problem ourselves, we assert that the V term merely shifts it up or down by a constant. For the present discussion we make $V = 0$. We summarize the procedure for finding the numerical solution with Algorithm 7.2.

ALGORITHM 7.2 / A Numerical Solution to the *LRC* Circuit Using the Feynman-Newton Method

STEP 1. Choose the parameters R, C, L, and V, choose a value for h, and specify the initial conditions q_0 and i_0. Do not lose sight of the fact that $i(t) = q'(t)$.
STEP 2. Calculate the second derivative at $t = 0$;

$$q_0'' = -\frac{1}{L}\left[\frac{q_0}{C} + Rq'\right].$$

STEP 3. From the second derivative find the first derivative at $t = h/2$:

$$q' = q_0' + \frac{h}{2}q_0'' \quad \text{or} \quad i = i_0 + \frac{h}{2}q_0''.$$

STEP 4. Begin a loop.
STEP 5. Plot a point on the graph of $q(t)$ and/or $i(t)$.
STEP 6. Update the charge:

$$q \leftarrow q + hq'.$$

STEP 7. Calculate a new value for the second derivative, q'' where

$$q'' = -\frac{1}{L}\left[\frac{q}{C} + Rq'\right].$$

STEP 8. Update the derivative of the charge:

$$q' \leftarrow q' + hq'' \quad \text{or} \quad i \leftarrow i + hq''.$$

STEP 9. Increment the time:

$$t \leftarrow t + h.$$

STEP 10. Loop back to STEP 5 until $t > t_{max}$.
STEP 11. Quit.

It is useful to compare this equation with Newton's Second Law written for the object-spring problem in Chapter 2. Table 7.2 compares a Newton's Second Law problem with our solution to the *LRC* circuit. Notice that

$$q''(t) \leftrightarrow a(t) \quad \text{and} \quad q'(t) \leftrightarrow v(t),$$

where $\leftrightarrow$ means "corresponds to." Also note that L is the analog of m, C is the analog of $1/k$, R is the analog of c, q is the analog of x, and the current i is the analog of v. Given this analogy, how do you expect the circuit to behave if the initial conditions specify that $q_0 \neq 0$ and $i_0 = 0$?

TABLE 7.2 • A Comparison of the Solution to an LRC Circuit with a Newton's Second Law Problem.

Kirchhoff's Loop Rule	**Newton's Second Law**
$$q''(t) = -\frac{1}{L}\left[\frac{q}{C} + Rq'(t)\right].$$	$$x''(t) = -\frac{1}{m}[kx + cx'(t)].$$
Specify the initial conditions q_0 and $q'_0(i_0)$.	Specify the initial conditions x_0 and $x'_0(v_0)$.
Solution: let	Solution: let
$$q'_{\frac{1}{2}} = i_{\frac{1}{2}} = i_0 + \frac{h}{2}q''_0.$$	$$x'_{\frac{1}{2}} = v_{\frac{1}{2}} = v_0 + \frac{h}{2}x''_0.$$
Loop: let	Loop: let
$$q \leftarrow q + hi \quad \text{or} \quad q \leftarrow q + hq'.$$	$$x \leftarrow x + hv \quad \text{or} \quad x \leftarrow x + hx'.$$
calculate q'' from Kirchhoff's Loop Rule.	calculate x'' from Newton's Second Law.
Let	Let
$$i \leftarrow i + hq'',$$	$$v \leftarrow v + hx'',$$
and let	and let
$$t \leftarrow t + h.$$	$$t \leftarrow t + h.$$
End the loop.	End the loop.

Recall the motivation for this effort. We assumed that there was a current flowing in a *LR* circuit when the switch was opened. When the switch opens, di/dt is huge, and arcing across the contacts of the switch may occur. To help remedy this situation, we place a capacitor across the switch. The next problem will model this set of circumstances. We want to know how the current through the circuit varies and how the voltage across the inductor changes with time. The motivation that led to our study of the *LRC* circuit is not the only motivation for studying *LRC* circuits. These circuits occur in many applications—radios, stereos, transmitters, and telephones to name a few. It is likely that you will study oscillations in *LRC* circuits when you study electromagnetic radiation. With that in mind, let us proceed with the next problem.

PROBLEM 7.11 Study of an *LRC* Circuit • Write a program to implement Algorithm 7.2. Choose these parameters: L = 1 mH, R = 5 Ω, C = 1 μF, V = 10 V, $h = 5 \times 10^{-7}$, dom = [0, 0.001] s, and ran = [–10, 10] V. For initial conditions choose $q_0 = 0$ and $i_0 = 0.2$ A. Graph

$$V_L = -L\frac{di}{dt} = -L\frac{d^2q}{dt^2}$$

as a function of time. Also graph the current in the circuit as a function of time. For graphing purposes, the current is scaled up by a factor of 20. Our results are shown in Figure 7.6. Here are some questions to be answered:

- What happens to the voltage across the inductor when the switch is opened? Describe what you observe.
- What is the approximate frequency of the oscillations? Your frequency should be approximately equal to the resonant frequency of the circuit. Your textbook will show that the resonant frequency is given by the expression

$$f = \frac{1}{2\pi\sqrt{LC}}.$$

- How does the graph of V_L versus t change if L is increased? If C is increased? If R is increased?
- Describe how the slopes of the curve in Figure 7.6 show the derivative relationship between $i(t)$ and $V_L(t)$.
- Repeat the experiment with $V = 0$. How do the oscillations change?
- Repeat the experiment with $V = 0$ and $R = 0$. How do the oscillations change?
- What mechanical system that shows similar behavior?

PROBLEM 7.12 Critical Damping • Execute the program of Problem 7.11 with L = 1 mH, C = 0.1 μF, R = 63.25 Ω, $h = 0.5 \times 10^{-5}$ s, dom = [0, 0.25] ms, and

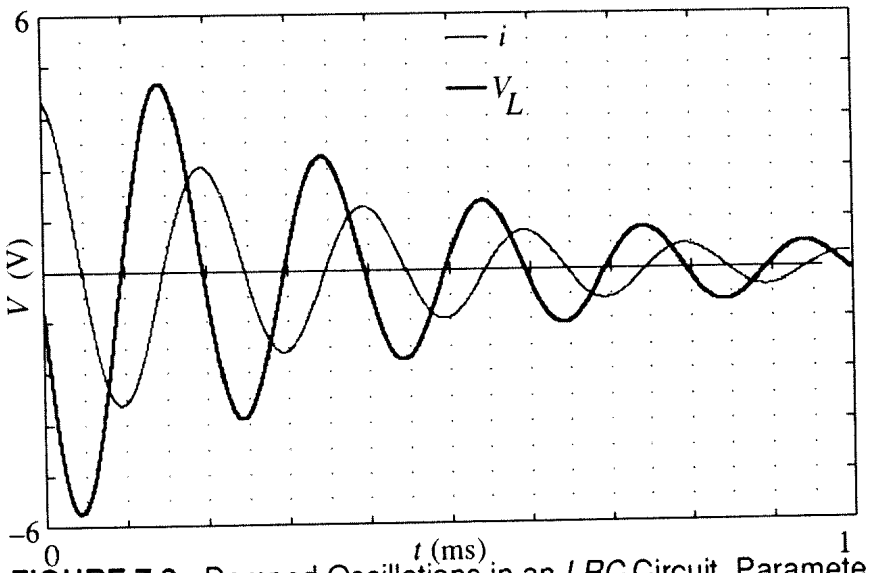


FIGURE 7.6 • Damped Oscillations in an *LRC* Circuit. Parameters: L = 1 mH, R = 5 Ω, C = 1 μF, h = 1 x 10^{-7} s, *dom* = [0, 1] ms, *ran* = [−6, 6] V, q_0 = 0, and i_0= 0.2 A.

ran = [−5, 5] V, with $q_0 = 0$ and $i_0 = 0.25$ A. In this case the circuit is said to be *critically damped*; there are no oscillations. The critical damping resistance is given by the formula

$$R = 2\sqrt{\frac{L}{C}}.$$

The system portrayed by Figure 7.6 is said to be *underdamped*. You have just examined the critically damped situation, and with R larger than the critically damped case the system is *overdamped*.

If you have not already guessed, our circuit is the analog of a mass vibrating on a spring. L is the analog of m, C is the analog of $1/k$, where k is the spring constant, and the resistance is the analog of the coefficient of the velocity term in the equation of motion of the mass on a spring. Consult your physics textbook for extensions of this analogy. Your car is an "object on a spring." Which case of damping— underdamped, critically damped, or overdamped— would you choose for your car?

We conclude our study of the *LRC* circuit by looking at it from a new perspective. Human beings are quite time conscious, and many of the graphs that describe the behavior of a system will have some variable graphed versus time, $i(t)$ in the *LRC* circuit for example. But, rather than graph $i(t)$, what do we obtain when we graph $i(t)$ versus $q(t)$? Figure 7.7 is the answer. The flow of time is, like a clock, clockwise. Do you perceive the oscillations in this figure as easily as in Figure 7.6? Can you make correspondences between points on the graph of $i(t)$ in Figure 7.6 with points on the graph of Figure 7.7? The point with coordinates $q = 0$ and $i = 0$ is called an attractor.

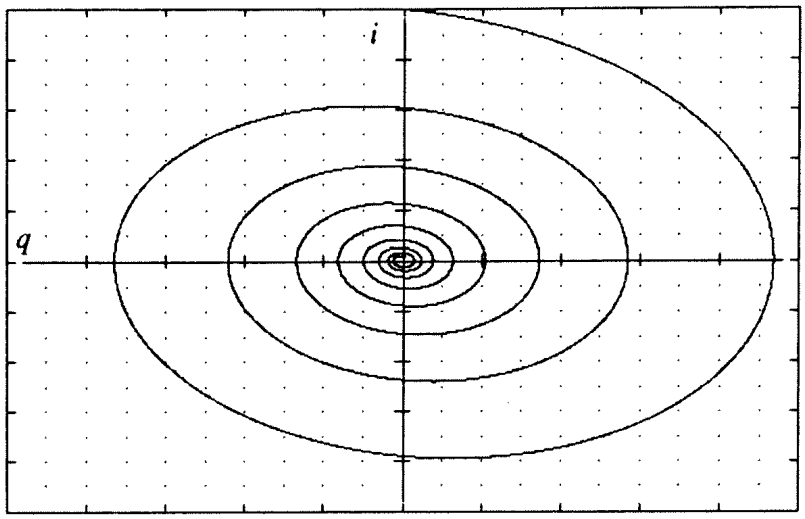


FIGURE 7.7 • A Graph of $i(t)$ vs. $q(t)$ for the *LRC* Circuit. Parameters: $L = 1$ mH, $R = 5\Omega$, $C = 1\ \mu$F, $V = 0$, $h = 2 \times 10^{-7}$ s, $q_0 = 0$, $i_0 = 0.2$ A, *dom* = [–6, 6] μC, and *ran* = [–0.2, 0.2] A.

7.5 Additional Problems

PROBLEM 7.13 Energy Study of an *LRC* Circuit (Challenging) • Use the methods described in Sections 7.2.3 and 7.3.3 to make an energy study of the *LRC* circuit. Make graphs of energy versus time for the energy dissipated by the resistor, stored in the magnetic field around the inductor, and stored in the capacitor. Remove the battery from the circuit, and for initial conditions choose $q_0 = 0$ and $i_0 = 0.2$ A. First study a circuit with a resistance R, then take the resistor out of the circuit and simply monitor energy associated with the inductor and the capacitor. The choice of parameters is yours.

PROBLEM 7.14 Graphing Energy Versus $q(t)$ and $i(t)$ • Use the methods of Problem 7.13 to graph the energy stored in the capacitor as a function of q, the charge on the capacitor. Repeat this experiment for the energy associated with the inductor, graphing both on the same axes. Then graph these two energy terms versus $i(t)$. What new insights have you gained?

7.6 Programs

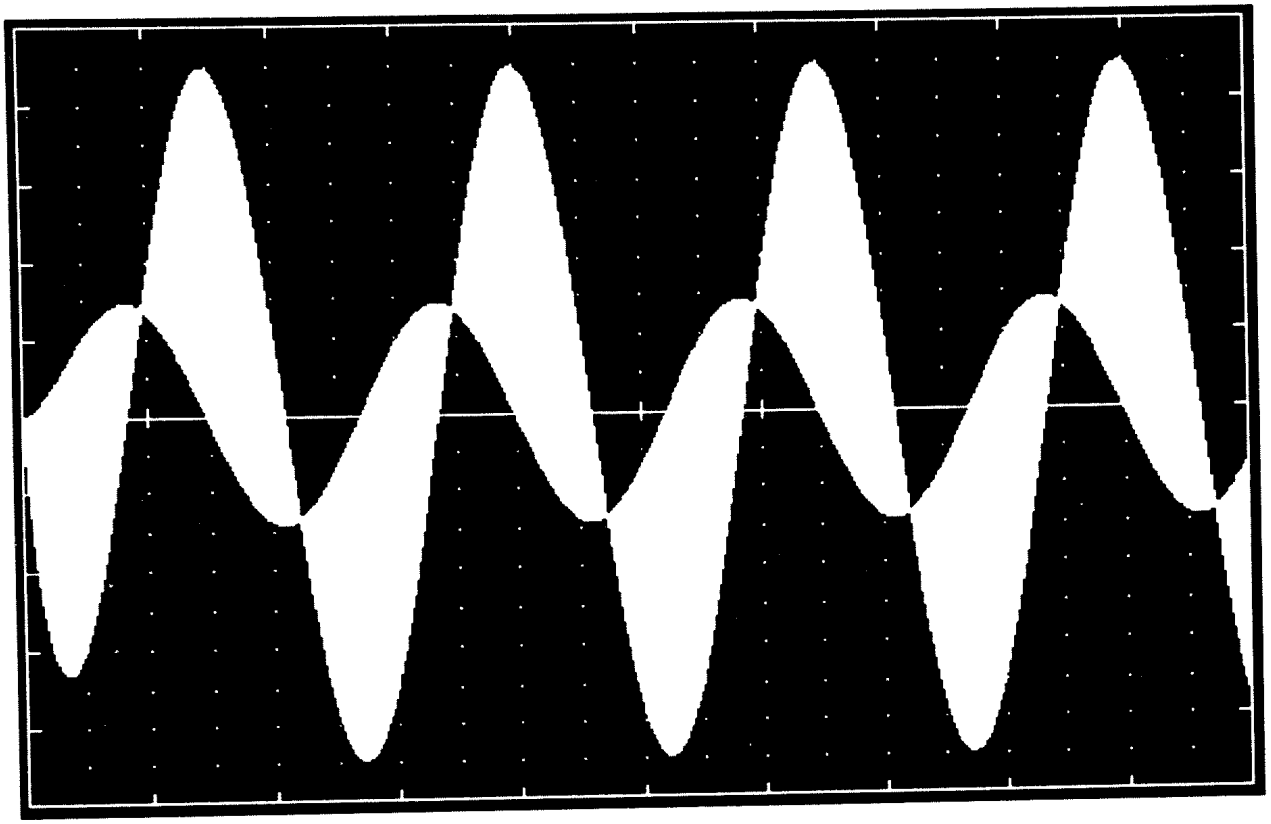
PROGRAM 7.6.1 / Computer Model of an *RC* Circuit

```
! Algorithm 7.1, Problems 7.1 to 7.10
! Choose h = 0.001, dom = [0, 0.5], ran = [0, 0.001], R = 1000 ohms, and C = 100 microfarads.
Library "Comphys"
CALL Open(xmin, xmax, ymin, ymax, wxmin, wxmax, wymin, wymax)
LET h = 0.001                                          ! Choose step-size h
LET V = 10                                             ! Voltage of the battery.
LET C= 100e-6                                          ! Capacitance.
LET R = 1000                                           ! Resistance.
```

```
Let q = 0
DO                                                    ! This is the main loop.
   PLOT t, q                                          ! Graph q(t).
   LET qp =V/R - q/(R*C)                              ! Calculate the derivative (also
                                                        i(t).
   LET q = q + h*qp                                   ! Calculate the new charge.
   LET t = t + h                                      ! Increment the time.
LOOP until t >= xmax                                  ! Loop until t exceeds xmax.
CALL LABEL (wxmin, wxmax, wymin, wymax)               ! Label graph.
END
```

PROGRAM 7.5.2 / Computer Model of an *LRC* Circuit

```
! Algorithm 7.2, Problems 7.11 and 7.12.
! Let L = 1 mH, R = 5 ohms, C = 1 microfarad, V = 0.
! Choose dom =[0, 0.001], ran = [-10, 10].
! q(0) = 0, i(0) = 0.2
!..............
! The second derivative in Kirchhoff's Loop Rule is defined here.
DEF qpp(q,i,t) = -(q/C+i*R-V)/L                       ! qpp is q"
!.............
LIBRARY "Comphys"
CALL GET_INFO (q,i,L,R,C,V)                           ! Fetch the parameters.
CALL Open(xmin, xmax, ymin, ymax, wxmin, wxmax, wymin, wymax)
CALL Save (wxmin, wxmax,wymin,wymax,keep$)            ! Save coordinate system.
LET h = .05e-5                                        ! Step-size h.
LET i = i + qpp(q,i,t)*h/2                            ! Starting value for i (or q')
                                                        using Feynman-Newton.
DO until KEY INPUT                                    ! This is the main loop.
   PLOT tp, -L*qpp(q,i,t)                             ! Plot voltage across L.
   LET q = q + h*i                                    ! Calculate new q.
   PLOT tp, 20*i                                      ! Scale i(t) and graph it.
   LET i = i + qpp(q,i,t)*h                           ! New value for i.
   LET t = t + h                                      ! Clock time.
   LET tp = tp + h                                    ! Plotting time.
   IF tp > xmax then
      LET tp = xmin                                   ! Start a new frame?
      CLEAR                                           ! Yes.
      CALL Show (wxmin,wymin, keep$)                  ! Bring the axes back.
   END IF
LOOP                                                  ! End of the main loop.
GET KEY z
CALL LABEL (wxmin, wxmax, wymin, wymax)               ! Label graph.
END
!............
SUB GET_INFO (q,i,L,R,C,V)
    INPUT PROMPT "What are the initial conditions? Input q(0) and i(0). ":q,i
    INPUT PROMPT "Please enter L, and R, and C. ":L,R,C
    INPUT PROMPT "Please enter the battery voltage, V. ":V
END SUB
```



Oscillations

8 / Driven *LR*, *RC*, and *LRC* Circuits

8.1 Introduction

A review of the following topics may be helpful in this chapter:

- Inductance and capacitance
- Kirchhoff's Loop Rule
- Phase relationships in ac circuits
- Resonance
- Numerical solutions to *RC*, *LR*, and *L RC* circuits (Chapter 7)

The essential difference in this chapter is that the constant-voltage battery in Chapter 7 is replaced by a function generator capable of producing time-varying signals. Your textbook will analyze *RC*, *LR*, and *LRC* circuits when the function generator produces a sinusoidal voltage, but when more complex functions are considered, the analysis becomes difficult. Thus the ability to analyze circuits with a computer becomes more important as the

complexity of the function driving the circuit increases, and, of course, with the increasing complexity of the circuit itself. This chapter begins with an analysis of an *LR* circuit, and you will be asked to analyze an *RC* circuit. Later in the chapter we will take a second look at an *LRC* circuit, this time with a function generator in it.

8.2 The *LR* Circuit

8.2.1 Computer Model of an *LR* Circuit Driven by a Sine Function

An *LR* circuit is shown in Figure 8.1. In Chapter 7 we used Kirchhoff's Loop Rule to find the differential equation that describes the current in the *LR* circuit. The equation is

$$i'(t) = \frac{1}{L}[V(t) - Ri(t)],$$

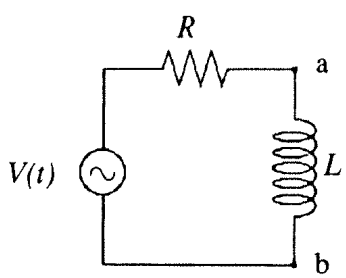
where R is the resistance and L is the self-inductance. This equation may be solved using Euler's method, a technique we described in Chapter 7. Algorithm 8.1 outlines the procedure we will use to solve this equation.

ALGORITHM 8.1 / Finding $i(t)$ in an *LR* Circuit

STEP 1. Specify the values of L and R and choose a value for the time interval h.
STEP 2. Choose the initial conditions: i_0, i'_0, and V_0.
STEP 3. Begin a loop.
  STEP 4. Plot a point on the graph of $i(t)$, and/or plot a point on the graph of $V_L(t)$.
  STEP 5. Calculate i':

$$i' = \frac{1}{L}[V - Ri].$$

FIGURE 8.1 • The *LR* Circuit Driven by a Function Generator. A function generator produces a voltage across its terminals that varies periodically with time. We identify this voltage as $V(t)$.



STEP 6. Update the current:

$$i \leftarrow i + hi'.$$

STEP 7. Increment the time:

$$t \leftarrow t + h.$$

STEP 8. Loop until $t > t_{max}$.
STEP 9. Quit.

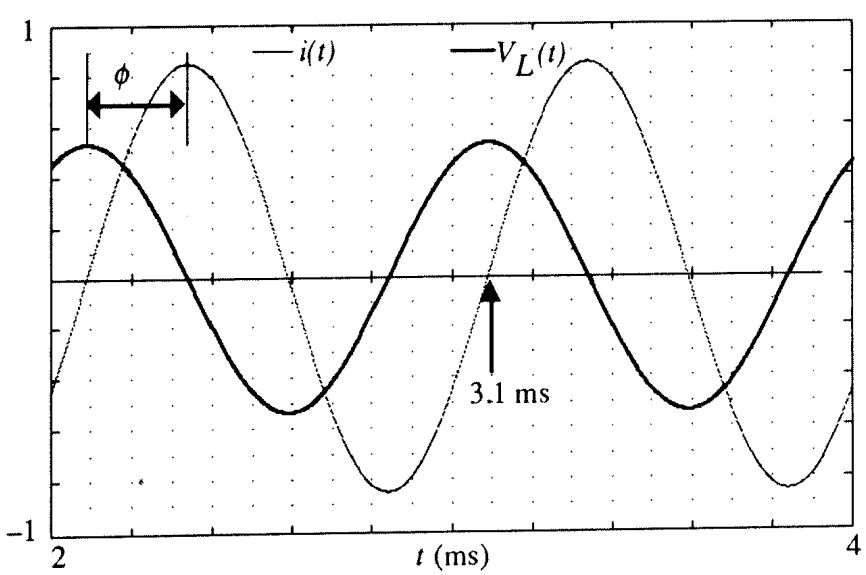
Our results are shown in Figure 8.2, where we have graphed the current through the inductor and the voltage across the inductor. This figure shows an interesting phenomenon; the current through the inductor is *out of phase* with the voltage across it. Study the graph in Figure 8.2 and you will see why the current is said to lag the voltage, meaning that the current reaches its peak value at a *later* time than the voltage. Notice that the lag, ϕ, is approximately a quarter of a complete cycle, or 90 degrees. In this case the current lags the voltage (or the voltage leads the current) by 90 degrees.

How can we explain Figure 8.2? Faraday's Law gives the relationship between the current through an inductor and the voltage across it in the situation shown in Figure 8.1. Thus

$$V_L = V_a - V_b = L\frac{di}{dt},$$

where L is the inductance and the subscripts a and b refer to points in circuit in Figure 8.1.

FIGURE 8.2 • Current in an *LR* Circuit and Voltage across the Inductor in an *LR* Circuit Driven by a Sine Function Voltage. Parameters: L = 1 mH, R = 10 Ω, f = 1000 Hz, $h = 2 \times 10^{-6}$ s, *dom* = [2, 4] ms, and *ran* = [−1, 1]. To allow transients time to die, we began the graph at t = 2 ms rather than at t = 0.



(In this case we used Lenz's Law to determine the correct sign for $V_b - V_a$. If di/dt is positive and i is clockwise in the circuit, then the self-induced emf is in the direction that makes V_a larger than V_b.) With reference to Figure 8.2, since di/dt is the slope of the $i(t)$ curve, V_L should be proportional to that slope. Indeed, a close examination of the curves will show that that is the case. For example, near $t = 3.1$ ms, at the location of the arrow and where $i = 0$, the slope of the $i(t)$ curve has its most positive value, and, indeed, that is where the voltage has a maximum.

We can give a more detailed explanation. In Figure 8.2 it is evident that the current may be expressed, at least approximately, with the formula

$$i(t) = i_0 \sin(2\pi f t),$$

where i_0 is the peak current. Implicit in this formula is the assumption that the origin of our t-i coordinate system is at one of the positive-going zero crossings for the current—for example, the zero crossing at 3.1 ms in Figure 8.2. We substitute this expression for the current into the formula for the voltage across the inductance, given above. Thus

$$V_L = L\frac{di}{dt} = L\frac{d}{dt} i_0 \sin(2\pi f t) = 2\pi f i_0 L \cos(2\pi f t).$$

Next we make use of the identity

$$\sin(\theta + \frac{\pi}{2}) = \cos\theta$$

to rewrite the previous equation in the form

$$V_L = 2\pi f i_o L \sin(2\pi f t + \frac{\pi}{2}),$$

proving that the voltage leads the current by a phase angle of $\pi/2$.

There is another phase relationship that is discussed in physics textbooks, namely the phase difference between the applied voltage, $V(t)$, and the current in the circuit, $i(t)$. These quantities are graphed in Figure 8.3, where the parameters are identical to those specified in Figure 8.2. Note that in an inductive circuit such as this, the current always lags the voltage. How does the phase difference change with R and/or L?

PROBLEM 8.1 Phase Relationships in an *RC* Circuit • Modify Algorithm 8.1 to find a numerical solution for $q(t)$, the charge on the capacitor, and $i(t)$, the current through the circuit, in an RC circuit driven by a 1000 Hz sine function. Recall from Chapter 7 that

$$q'(t) = \frac{1}{R}\left[V - \frac{q(t)}{C}\right].$$

Let

$$V(t) = \sin(2\pi f t).$$

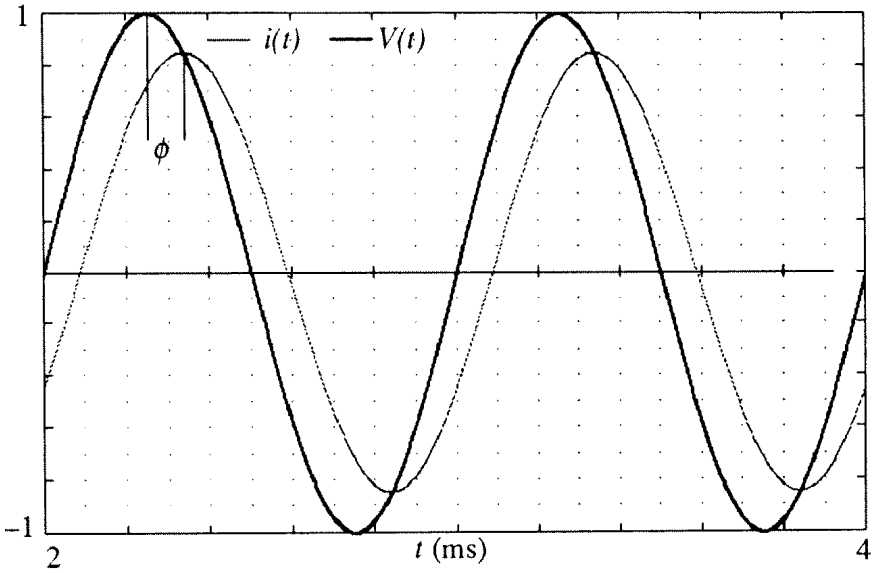


FIGURE 8.3 • Phase Relationship between *V(t)* and *i(t)* in the *LR* Circuit. Parameters are those given in Figure 8.2. The phase angle is ϕ, which is approximately 1/10 of a cycle, or nine degrees, and the voltage leads the current.

Graph $i(t)$ and the voltage across the capacitor, $V_C(t)$, where

$$V_C(t) = \frac{q(t)}{C}.$$

What is the phase relationship between $i(t)$ and $V_C(t)$? Does the current lead or lag the voltage? Use these parameters: $C = 3.18 \times 10^{-5}$ F, $R = 10\ \Omega$, $f = 1000$ Hz, $h = 1 \times 10^{-5}$ s, *dom* = [0, 2] ms, and *ran* = [−1, 1]. Scale $i(t)$ and $V_C(t)$ so they both fit on the coordinate system. Explain the behavior of the circuit, making use of the fact that

$$i(t) = \frac{dq}{dt}$$

or

$$q(t) = \int i(t)\,dt.$$

Repeat the experiment, this time graphing both $V(t)$ and $i(t)$. Does the current in the circuit lead or lag the driving voltage, $V(t)$? Why?

8.2.2 An *LR* Circuit Driven by a Square Function

In this section we will change the driving voltage to be a square (or step) function of time. This function periodically switches between −1 V and +1 V. It is illustrated by the thin line in Figure 8.4. On a computer the square function may be generated with the following expression:

$$\text{If } \sin(2\pi f t) \geq 0 \text{ then } V(t) = 1, \text{ else } V(t) = -1.$$

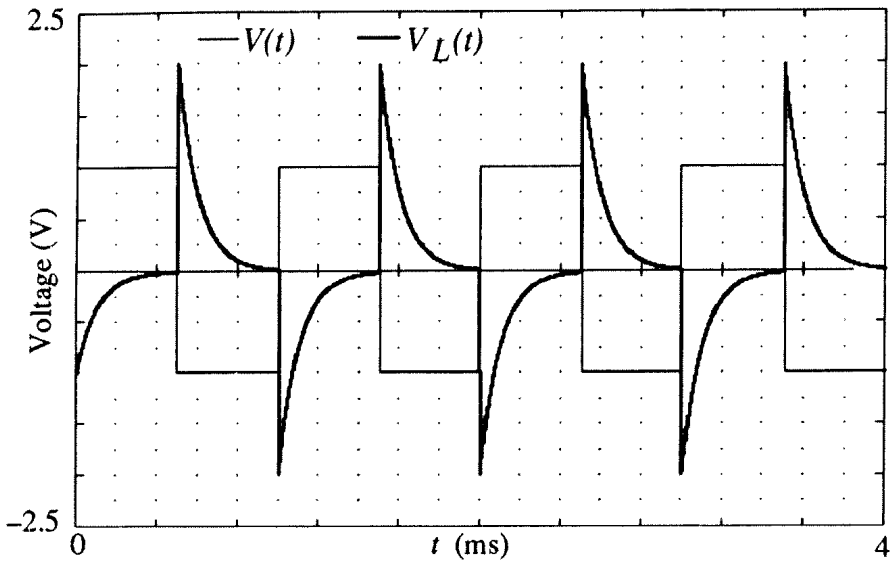


FIGURE 8.4 • Voltage Across an Inductor in an *LR* Circuit Driven by a Square Function. Parameters: $L = 1$ mH, $R = 10\ \Omega$, $h = 1 \times 10^{-6}$ s, *dom* = [0, 4] ms, and *ran* = [−2.5, 2.5] V.

The square function is periodic because the sine function is periodic. The response of the *LR* circuit to the square function is the heavy line in Figure 8.4. Why does the voltage across the inductor reach its largest value when the square function steps from 1 V to −1 V? Predict how the voltage across the resistor will vary with time, then use your computer program to graph it.

How can we explain the appearance of Figure 8.4? When the voltage across the *LR* combination switches from −1 to 1, for example, the current is forced to "try" to reverse itself. The inductor "opposes" this change, so its voltage switches to the opposite direction. Since $i(t)$ is going positive, $V_L(t)$ will go negative. When di/dt is a maximum, V_L is a minimum, just as Faraday's Law predicts.

PROBLEM 8.2 An *RC* Circuit Driven by a Square Function • Analyze the *RC* circuit when $V(t)$ is a square (or step) function of time. Choose $C = 3.18 \times 10^{-5}$ F, $R = 10\ \Omega$, $f = 1000$ Hz, $h = 1 \times 10^{-5}$ s, *dom* = [0, 4] ms, and *ran* = [−1, 1] V. Make graphs of the voltage, V_C, across the capacitor and the current, $i(t)$ in the circuit. Give an explanation of the behavior of these two voltages in time, making use of the derivative relationships between $i(t)$ and $q(t)$ for charging and discharging capacitors.

8.3 Computer Model of an *LRC* Circuit

8.3.1 An *LRC* Circuit Driven by a Sine Function

Figure 8.5 shows an *LRC* circuit with a function generator attached. In the previous chapter we applied Kirchhoff's Loop Rule to this circuit to obtain the differential equation

$$V(t) - L\frac{di}{dt} - \frac{q}{C} - Ri = 0.$$

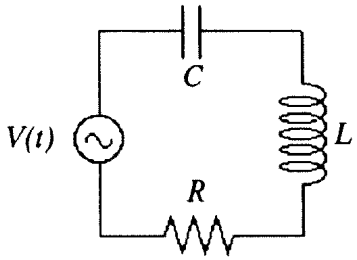


FIGURE 8.5 • The *LRC* Circuit Driven by a Function Generator with Voltage $V(t)$.

With some mathematical work, we transformed this equation into an expression for $q''(t)$, with the following result:

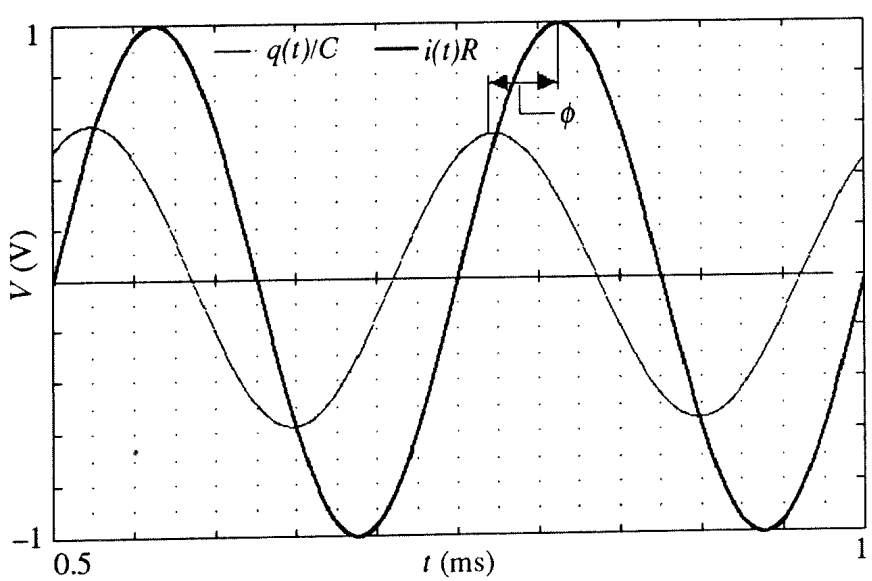
$$q''(t) = -\frac{1}{L}\left[\frac{q(t)}{C} + Rq'(t) - V(t)\right].$$

We found numerical solutions to this equation using the Feynman-Newton method. Algorithm 7.2 in the previous chapter outlines the procedure. To study the behavior of the *LRC* circuit being driven by a function generator with voltage $V(t)$, we need only replace the dc voltage V in Algorithm 7.2 by the voltage of the function generator. Since Algorithm 7.2 is essentially unchanged, we will not repeat it here.

The graphs in Figure 8.6 are scaled versions of the current, $i(t)$, in the circuit and the charge, $q(t)$, on the capacitor of Figure 8.5. Can you tell from the slopes that

$$i(t) = \frac{dq(t)}{dt}?$$

FIGURE 8.6 • Scaled Versions of $i(t)$ and $q(t)$ for a Driven *LRC* Circuit. Parameters: $L = 1$ mH, $R = 10\ \Omega$, $C = 1\ \mu$F, $h = 1$ ms, $f = 4000$ Hz, $dom = [0.5, 1]$ ms, and $ran = [-1,1]$ V. $V(t)$ is sinusoidal. The graph starts at 0.5 ms to eliminate the transient behavior near $t = 0$.



What is the phase difference ϕ between $i(t)$ and $q(t)$? The current leads the charge; try to give a conceptual explanation of this.

PROBLEM 8.3 Phase Relationships in an *LRC* Circuit • Make graphs of $i(t)$ and $V(t)$ for the *LRC* circuit driven with a 4000 Hz source. Use $L = 1$ mH, $R = 5\ \Omega$, $C = 1\ \mu$F, $h = 1\ \mu$s, $dom = [0, 1]$ ms, $ran = [-1, 1]$ V. What is the phase angle between $V(t)$ and $i(t)$? Your textbook will show that the theoretical phase angle is given by

$$\tan\phi = \frac{X_L - X_C}{R},$$

where

$$X_L = 2\pi f L$$

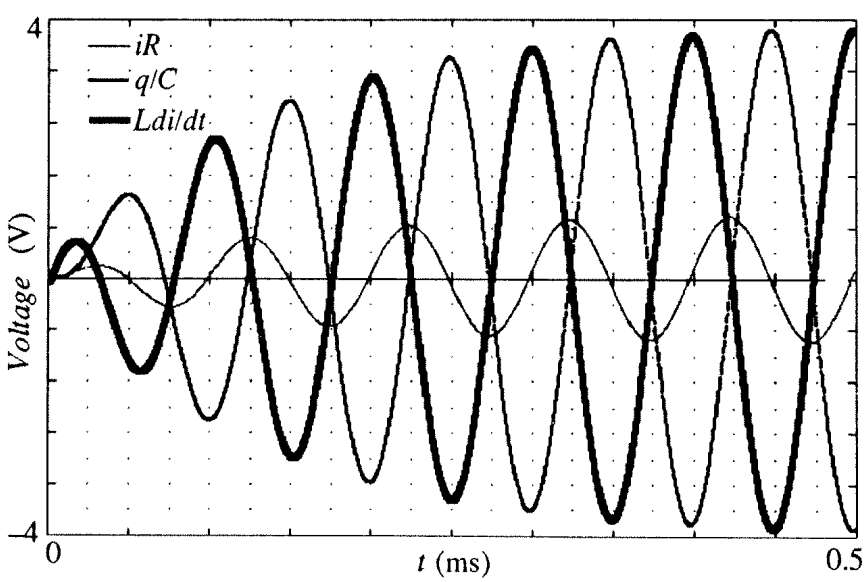
and

$$X_C = \frac{1}{2\pi f C}.$$

X_L and X_C are the inductive and capacitive reactances, respectively. How well does your measured value for ϕ agree with the theoretical value? Repeat your measurements for the situation in which $f = 6000$ Hz.

PROBLEM 8.4 What's Happening Here? • Study Figure 8.7 and try to explain what is happening. Notice that we are graphing the voltages across the resistor, the capacitor, and the inductor. Modify your program to graph these three voltages, then observe what happens

FIGURE 8.7 • Voltages in an *LRC* Circuit. Parameters: $L = 1$ mH, $R = 10\ \Omega$, $C = 1\ \mu$F, $h = 0.5\ \mu$s, $dom = [0, 0.5]$ ms, and $ran = [-2, 2]$ V.



when $f = 4000$ Hz, 5033 Hz, and 6000 Hz. Observe how the voltages change with time, but eventually reach steady-state values.

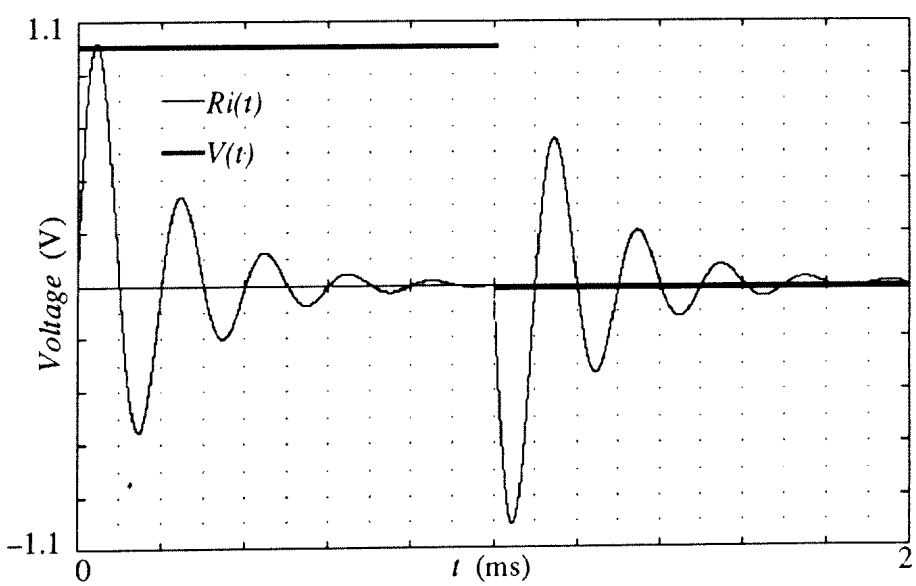
8.3.2 An *LRC* Circuit Driven by a Square Function

A *transient* voltage or current is an abrupt and temporary electrical signal, usually undesirable, that occurs when an electrical circuit is started, stopped, or switched from one state to another. For example, when an electric motor is started, there is an initial current surge, and when the same motor is stopped, the voltage across the on-off switch may become large enough to cause sparking. You can, in fact, purchase devices to insert between a computer and its power source to eliminate damaging current or voltage transients. Perhaps a bolt of lightning is the outstanding example of a transient.

The circuits and situations we have studied thus far in this chapter yield their secrets to skillful analys—algebra, trigonometry, calculus, and differential equations—as well as computer simulations. These circuits may also be studied in the laboratory with VOMs and oscilloscopes. The name "oscilloscope" suggests its capacity for observing oscillations. Indeed, you should become familiar with these measurements in the physics laboratory. One advantage of using a computer to simulate these circuits is the ability to monitor variables anywhere in the circuit, including places where VOMs and oscilloscopes sometimes cannot go. It is also the case that transients are sometimes difficult to display and analyze on an oscilloscope, and it may be easier and quicker to use a computer to simulate the response of a circuit to changing conditions.

Consider the transients that are produced when a square function voltage is applied to a *LRC* circuit. One example is shown in Figure 8.8, in which case the driving voltage

FIGURE 8.8 • Transients in an *LRC* Circuit. Parameters: $L = 1$ mH, $C = 1$ μF, $R = 10\ \Omega$, $h = 0.2$ μs, $f = 500$ Hz, *dom* = [0, 2] ms, and *ran* = [−1.1, 1.1] V. The quantity *Ri(t)* in the figure is scaled up by a factor of two. The phenomenon shown is called *ringing*.



switched between zero and one, as illustrated with the heavy line in the figure. Recall from the previous chapter that, given an appropriate set of initial conditions, an *LRC* circuit may oscillate. That is exactly what is pictured in Figure 8.8. A switch in the driving voltage is equivalent to a new set of initial conditions, and when $V(t)$ switches the circuit oscillates. Because there is resistance in the circuit, the oscillations eventually die, only to start anew when once again the voltage switches.

In Chapter 7 we pointed out that a *LRC* circuit has a mechanical analog in a system consisting of a mass on the end of a spring. L is the analog of the mass, C is the analog of $1/k$ where k is the spring constant, and R is the analog of the damping factor, c. We may think of the charge on the capacitor as the analog of the displacement of the mass. Both the charge on the capacitor and the displacement of the mass oscillate back and forth when given an appropriate set of initial conditions. With that in mind, describe the vertical motion of a car that would result from driving a car without shock absorbers over a series of rectangular bumps, and compare this behavior with that illustrated in Figure 8.8.

It may be worth noting that the logic circuitry found in computers and a host of other devices makes use of voltages that are identical to the square-function voltage shown in Figure 8.8, and the kind of transient behavior we observe here is highly undesirable in those circuits. Look inside a computer and notice all of the printed circuit wiring; there are a considerable number of "loops" made by the wiring, and each of these has some inductance. The proximity of these conductors to each other also introduces capacitance into the circuit. Thus the ingredients for producing the transients shown in Figure 8.8 are built into the circuitry of almost every electronic device.

PROBLEM 8.5 Studying Transients in an *LRC* Circuit • The results in Figure 8.8 were obtained by making $V(t)$ a square function that switched back and forth between zero and one volt. How do the results change if $V(t)$ switches back and forth between minus one and plus one volt? Keep the parameters the same as the ones given in Figure 8.8. How do the oscillations change if R is increased? Describe this with a sentence or two. How do the oscillations change if C is increased? If L is increased? Give a conceptual explanation of the behavior you observe.

8.3.3 Phase Space for *LRC* Circuits

Recall the phase-space graph we made in Chapter 2 for the object-spring system. Our variables in that case were the displacement x and the velocity v. For a *LRC* circuit, the corresponding variables are the charge q on the capacitor and the current i through the circuit. It is easy to modify the programs in this chapter to graph the trajectory of a driven *LRC* circuit in phase space. Simply plot i versus q and omit all references to time in the plotting instructions. An example of a phase portrait is shown in Figure 8.9.

PROBLEM 8.6 Phase Space Portrait of an *LRC* Circuit Driven by a Sine Function • Modify your program to graph q versus i. Exit the main loop with a key press. Initiate the program with the parameters given in Figure 8.9, using a sine function for

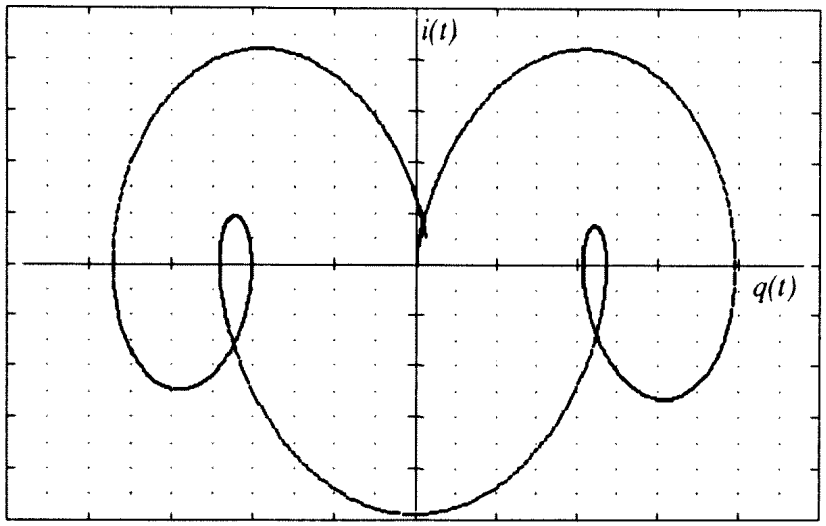


FIGURE 8.9 • Trajectory in Phase Space of a Driven *LRC* Circuit. Parameters: L = 1 mH, C = 1 μF, R = 1 Ω, h = 0. 2 μs, f = 20,000 Hz, *dom* = [−0.4 , 0.4] x 10^{-6} C, and *ran* = [−1.7, 1.7] x 10^{-2} A. $V(t) = \sin 2\pi f t$.

$V(t)$, namely

$$V(t) = \sin 2\pi f t.$$

Make phase diagrams for f = 15,000, 10,000, 5,000, and 2,000 Hz. You may have to change the domain and range to keep the trajectory on the coordinate system. Cite and explain the differences in these phase diagrams. How does the phase diagram change with increasing R?

8.4 Additional Problems

PROBLEM 8.7 Mechanical and Electrical Analogs (Challenging) • Try to find an exact *mechanical* analog of the phase diagram in Figure 8.9. Use the algorithms in Chapter 2 for an object-spring system. Both systems should give identical phase diagrams. Is the mechanical analog realistic?

PROBLEM 8.8 Another Technique for Measuring the Phase Angle Φ (Challenging) • Write a program that generates two signals, one called i and the other called V, where

$$i = A\sin(2\pi ft)$$

and

$$V = B\sin(2\pi ft - \phi).$$

The angle ϕ is restricted as follows:

$$-\frac{\pi}{2} \le \phi \le \frac{\pi}{2}.$$

Place the instructions that calculate i and V in a loop in which t is incremented by a small time step h. Assume a frequency of 1000 Hz, and plot the points (i, V) generated by these calculations. Begin with $A = B = 1$ but experiment with other values and try various values of ϕ. Describe your results. You should see an ellipse on the screen similar to the ellipse shown in Figure 8.10.

Use the equations for i and V given above to show that

$$\sin\phi = \pm\frac{b}{B}.$$

You will need the trigonometric identity

$$\sin(\alpha + \beta) = \sin\alpha\cos\beta + \cos\alpha\sin\beta,$$

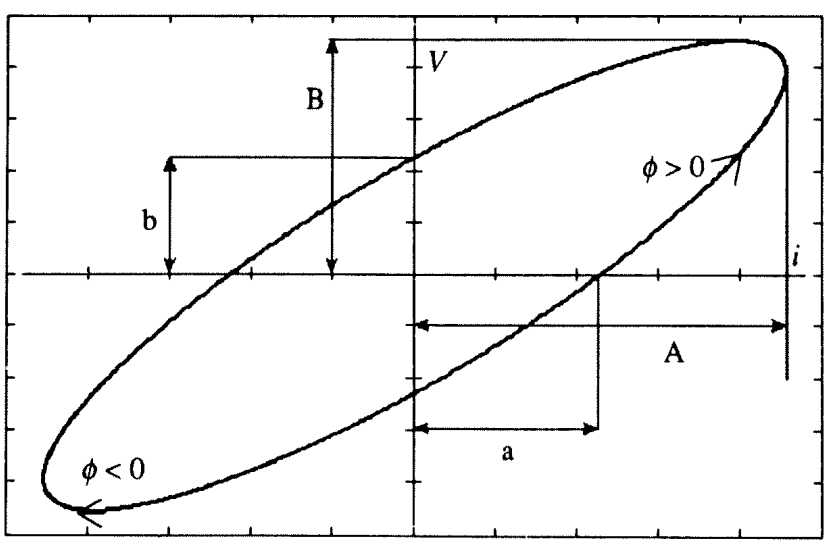
and you must make use of the fact that $i = 0$ when the curve crosses the V-axis. The requirement that $i = 0$ means that

$$2\pi ft = 0,\ \pm\pi,\ \pm 2\pi,\ \cdots,\ \pm n\pi,$$

for that is when

$$\sin 2\pi ft = 0,$$

FIGURE 8.10 • Phase Measurement. The measurements of A and a or B and b are frequently made by connecting an oscilloscope to the circuit, with the horizontal deflection proportional to $i(t)$ and the vertical deflection proportional to $V(t)$. Time increases counterclockwise if ϕ is positive, clockwise if ϕ is negative.



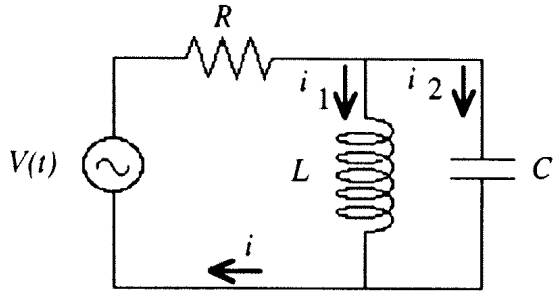


FIGURE 8.11 • A Parallel *LC* Circuit.

and

$$\cos 2\pi ft = \pm 1.$$

Thus the phase angle ϕ may be measured from the graph of $i(t)$ versus $V(t)$ shown in Figure 8.10 and the relationship

$$\sin\phi = \pm\frac{b}{B}.$$

Repeat Problem 8.3, this time graphing $V(t)$ versus $i(t)$. Measure the phase angle using the ellipse and compare it to the theoretical value (see Problem 8.3). Try the phase measurement technique developed in this problem on *LR* and *RC* circuits. For what phase angles is the technique most precise? Least precise? Set up actual circuits in the laboratory and compare your experimental work with theoretical work and computational work.

PROBLEM 8.9 Numerical Solution to a Parallel *LC* Circuit (Challenging) • A circuit diagram for a parallel *LC* circuit is given in Figure 8.11. This circuit is not usually solved in introductory physics texts, in spite of the fact that it is one of the most widely used circuits in entertainment and communication electronics systems. This circuit will require that we simultaneously solve two first-order differential equations.

The first of the differential equations is found by applying Kirchhoff's Loop Rule to the loop containing *L*. Starting at the function generator, we find that

$$V(t) - (i_1 + i_2)R - L\frac{di_1}{dt} = 0.$$

Note that i_1 is the current in the *L*-branch and i_2 is the current in the *C*-branch of the circuit. The second differential equation is obtained by applying Kirchhoff's Loop Rule to the loop containing *C*. Thus if q is the charge on the capacitor, we may write

$$V(t) - (i_1 + i_2)R - \frac{q}{C} = 0.$$

This is not yet a differential equation, but when we make use of the fact that

$$i_2 = \frac{dq}{dt}$$

we find that

$$V(t) - \frac{dq}{dt}R - i_2 R - \frac{q}{C} = 0.$$

We also know that $i = i_1 + i_2$.

We rewrite the differential equations in the following way:

$$\frac{di_1}{dt} = \frac{R}{L}\left[V(t) - (i_1 + i_2)\right], \qquad \text{I}$$

and

$$\frac{dq}{dt} = \frac{1}{R}\left[V(t) - Ri_1 - \frac{q}{C}\right]. \qquad \text{II}$$

We have identified these differential equations as I and II. Now we are ready to describe the process of solving these two equations simultaneously.

Choose some parameters. Try $L = 1$ mH, $R = 10\ \Omega$, $C = 1\ \mu$F, and $h = 1\ \mu$s. To begin let

$$V(t) = \sin 2\pi f t,$$

with $f = 4000$ Hz. Choose some initial conditions. You might make $i_1 = i_2 = q = 0$. Calculate dq/dt from equation II and apply it to this formula to update q:

$$q \leftarrow q + h\left(\frac{dq}{dt}\right).$$

Make use of the fact that

$$i_2 = \frac{dq}{dt}$$

to find di_1/dt from equation I. Apply that derivative to this formula to update i_1:

$$i_1 \leftarrow i_1 + h\left(\frac{di_1}{dt}\right).$$

One step in the iteration process is finished when we let

$$i = i_1 + i_2,$$

and update t:

$$t \leftarrow t + h.$$

Loop back to equation I and continue the process until you have as much of the solution as you wish. Variables to graph include $i(t)$, $V_L(t)$ and/or $V_C(t)$. Try the parameters suggested above with *dom* = [0, 1] ms and *ran* = [−1, 1] V. Investigate what happens to the ratio V_C/i at resonance, when f = 5033 Hz. Study the behavior of the circuit when R is very large, $10^6\ \Omega$ for example, given an initial charge on the capacitor of 1 μC, with *no* driving voltage [setting $f = 0$ makes $V(t) = 0$]. You might also wish to study the phase relationships for this circuit.

8.5 Programs

PROGRAM 8.5.1 / Computer Model of an *LR* Circuit

```
! Algorithm 8.1, Problem 8.1, 8.2, etc.
! Study of a LR circuit.
! Try R= 10 ohms, L=1 mH, f = 1000, dom=[0,.002], ran=[-1,1], h=2e-6.
!.......................
DECLARE DEF V
! Define derivative here. "ip" stands for "i prime,"
DEF ip(t) = (V(t,f)-i*R)/L                                  ! the derivative of i.
!.......................
! Define function generator here.
DEF V(t,f)= sin(2*pi*f*t)
!.......................
LIBRARY "Comphys"
CALL GET_INFO (L,R)                                         ! Subroutine to define
                                                              parameters.
CALL Open (xmin,xmax,ymin,ymax,wxmin,wxmax,wymin,wymax)
LET h = 2e-6                                                ! Choose the step size, h.
LET f = 1000                                                ! Frequency is 1000 Hz.
CALL SAVE (wxmin, wxmax, wymin, wymax, Keep$)               ! Save the coordinate system.
DO                                                          ! Begin loop.
   PLOT tp,-L*ip(t)                                         ! Graph voltage across L.
   PLOT tp, R*i                                             ! Graph scaled up version of
                                                              current.
   LET i= i + h*ip(t)                                       ! Calculate the new i.
   LET t = t + h                                            ! Increment the time.
   LET tp = tp + h
   IF tp > xmax then                                        ! Increment "plotting time."
      LET tp = tp - xmax                                    ! Time for a new frame?
      CLEAR                                                 ! Yes.
      CALL SHOW (wxmin, wymin,Keep$)                        ! Bring back axes.
      IF KEY INPUT then                                     ! Time to quit?
         GET KEY key                                        ! Then press a key
         EXIT DO                                            ! to exit loop.
      END IF
   END IF                                                   ! Not ready for new frame.
LOOP                                                        ! Keep looping.
CALL Label(wxmin, wxmax, wymin, wymax)                      ! Label graph.
END
```

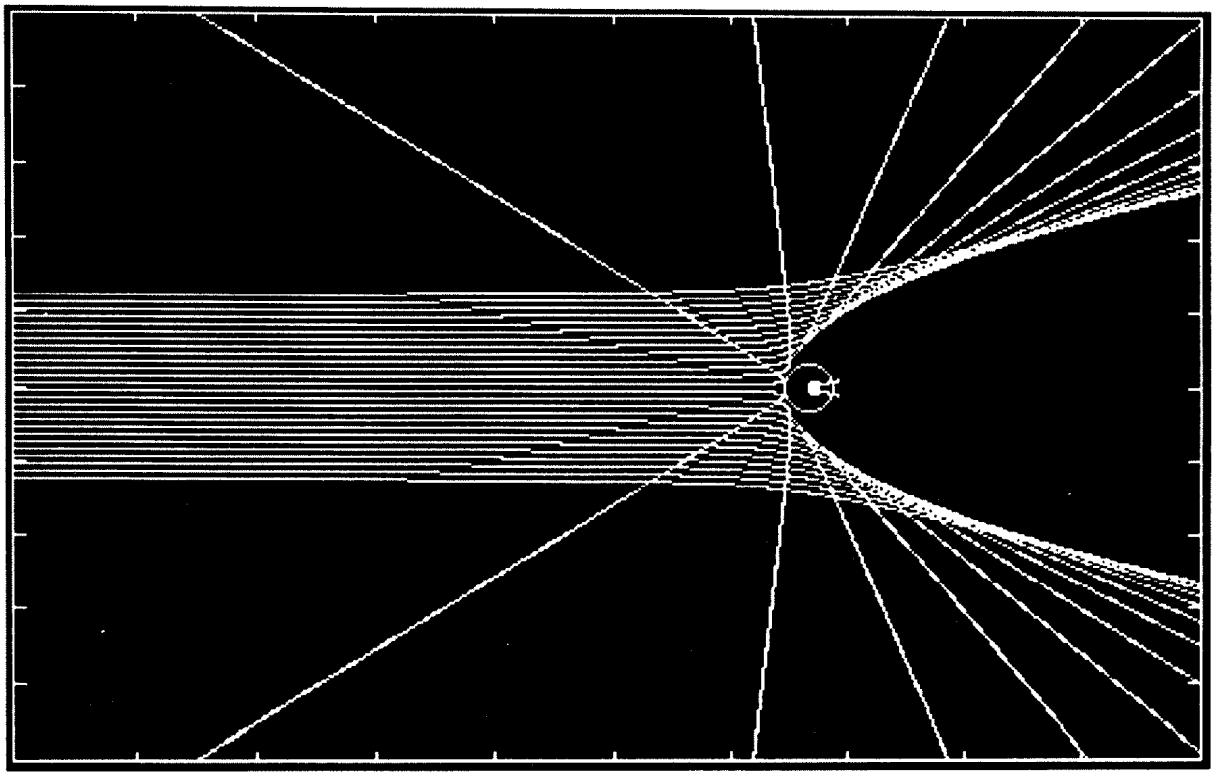
```
!............
SUB GET_INFO(L,R)
    INPUT PROMPT "Please enter L & R. ": L, R                ! Enter the parameters of the
                                                               problem.
END SUB
!............................
! Use this segment of code to generate a square function, but remove it for other functions.
! Define function generator here.
DEF V(t,f)
    LET y = sin(2*pi*f*t)
    IF y>0 then LET V=1
    IF y<=0 then LET V=-1
END DEF
```

PROGRAM 8.5.2 / Graphing Voltages and Currents for an *LRC* Circuit

```
! Algorithm 7.2 (previous chapter), Problem 8.3.
! Try L = 1 mH, C = 1 microfarad, R = 10 ohms, dom [0, 0.5] ms, and ran = [-2, 2] volt, and f =
                                                            3000 Hz.
!....................
DEF V(t,f)
LET V=sin(2*pi*f*t)
! ...................
! The second derivative in Kirchhoff's Loop Rule for an LRC circuit is defined here.
DEF qpp(q,i,t) = -(q/C+i*R-V(t,f))/L
!................
LIBRARY "Comphys"
CALL GET_INFO (L,R,C,f)                                     ! Get parameters L, R, C, and f.
CALL Open(xmin, xmax, ymin, ymax, wxmin, wxmax, wymin, wymax)
CALL Save (wxmin, wxmax,wymin,wymax,keep$)                  ! Save the screen.
LET h = 1e-6                                                ! Set h.
LET i = i + qpp(q,i,t)*h/2                                  ! The current i is the first
                                                              derivative of q. qpp is the
                                                              second derivative.
DO until KEY INPUT                                          ! Press a key to exit.
   PLOT tp, i*R                                             ! Plot voltage across R (which
                                                              is proportional to i).
   PLOT tp, q/C                                             ! Plot voltage across C (which
                                                              is proportional to q).
   PLOT tp, L*qpp(q, i, t)                                  ! Plot voltage across L (which
                                                              is proportional to q").
   LET q = q + h*i                                          ! Increment q.
   LET i = i + qpp(q,i,t)*h                                 ! Increment i.
   LET t = t + h                                            ! Clock time.
   LET tp = tp + h                                          ! Plotting time.
   IF tp > xmax then                                        ! Start a new frame when tp
                                                              exceeds xmax.
      LET tp = xmin                                         ! Start a new frame?
      CLEAR                                                 ! Yes. Then clear the screen.
```

```
      CALL Show (wxmin,wymin, keep$)                    ! Bring the axes back.
   END IF
LOOP                                                    ! Loop back for another point on
                                                          the graph.
GET KEY z                                               ! Clear the keyboard.
CALL LABEL (wxmin, wxmax, wymin, wymax)                 ! Label graph.
END
!............
SUB GET_INFO (L,R,C,f)
    INPUT PROMPT "Please enter L, and R, and C. ":L,R,C
    INPUT PROMPT "Please enter the frequency of the driving voltage. ":f
END SUB
```

●●●



Rutherford Scattering

9 / Rutherford Scattering and Other Collisions

9.1 Introduction

A review of the following topics may be helpful in this chapter:

- Coulomb's Law
- The electric field, **E**
- Force on a charge in an electric field
- Atomic structure

We have used numerical integration to solve problems on the scale of the solar system where our dimensions of displacement were as large as 10^{11} m and our time intervals were as large as 10^7 s. We have also worked at the scale of projectiles and satellites, bound to the earth, with dimensions ranging from 10^7 m to a few meters, and our time intervals varied from 10^6 s to a few seconds. Our study of the H_2^+ molecule in the next chapter will put us at the

atomic scale of physics with dimensions characterized by the factors 10^{-10} m and 10^{-16} s. In this chapter we are going to study the nucleus of an atom characterized by dimensions of 10^{-15} m in space and approximately 10^{-22} s in time. It is worth noting that we can handle these problems with relatively few concepts of physics: Newton's Second Law, Newton's Law of Universal Gravitation, Coulomb's Law, and the concepts of work, kinetic energy, and potential energy. That a discipline like physics can be unified with just a few concepts is remarkable.

We turn now to one of the most fundamental experiments of this century, an experiment that was suggested by physicist Lord Rutherford (1871–1937) and carried out by H. Geiger and E. Marsden. The time was close to the turn of the century and the structure of atoms was still a complete mystery. The problem was one of size; you cannot see atoms through microscopes. But seeing something is only one way of obtaining information about an object. Another, not very different technique is to shoot things at it, and perhaps deduce some of the properties of the target from what happens. During most of this century, that is what many nuclear physicists have been doing.

By shooting x-rays at substances, physicists learned that atoms contained electrons. Of course, because the net charge of an atom is zero, there must also be some positive charges in it. One late nineteenth century model of an atom was J. J. Thomson's "plum-pudding" model in which the electrons (plums) were distributed through a more or less uniform mass of positive charge (pudding). In this model, atoms emit light when their electrons vibrate back and forth from their equilibrium positions in the atom, whose size is approximately 10^{-10} m. The electrons, being several powers of 10 less massive than the atom itself, would be relatively easy to accelerate and capable, therefore, of scattering x-rays.

At the turn of the century, radioactivity had already been discovered—that is, it was known that some atoms spontaneously emit alpha particles, an ion as massive as the helium atom with a charge equal to but opposite in sign to the charge of two electrons. With a stroke of genius, Rutherford proposed to study the distribution of mass and charge in the atom by shooting alpha particles at atoms and observing the results. The gravitational interaction of alpha particles and atoms is negligible, a fact that can be verified with a simple calculation. The operative force law is not gravitational, it is electrostatic, a situation that calls for Coulomb's Law. If the entire plum-pudding atom is electrically neutral, then the alpha particle should whisk right on through the atom. If there are discrete charges, then something different will happen, but since the mass of an alpha particle is more than 7000 times the mass of an electron, the electrons will not inhibit the motion of an alpha particle. Electrons will be swept out of the way by an incoming alpha particle. In this chapter we focus our attention on the interaction of the alpha particle with the pudding containing the + charges.

9.2 The Physics of the Rutherford Scattering Experiment

The physics needed to deal with this problem is quite simple. Find the electric field strength **E** in the vicinity of a sphere (the atom) of + charge, and also find **E** inside the sphere, assuming that the charge is uniformly distributed. Next, calculate the force, $\mathbf{F} = q\mathbf{E}$, on the alpha particle as it approaches the atom and/or enters the atom. Find the acceleration from Newton's Second Law, and put it in our Feynman-Newton algorithm to generate a numerical solution. That's

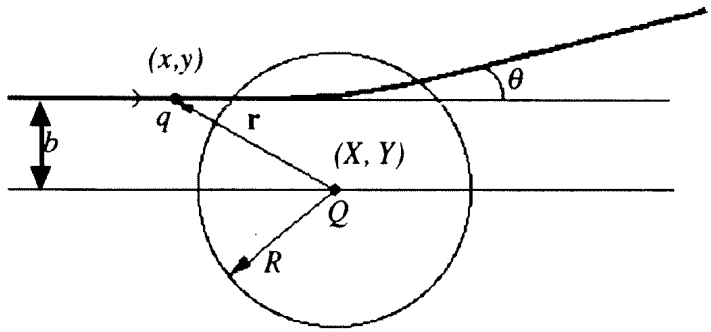


FIGURE 9.1 • An Alpha Particle Encounters a Gold Atom. The center of the gold atom is at (X, Y), and has a charge Q, uniformly distributed in a sphere of radius R. The alpha particle, with charge q, is at (x, y).

it. Although the principles are simple, the problem is made more complex than previous Newton's Law problems because we will allow the atom to accelerate and move about. Thus we have four sets of motion equations, one for each coordinate of both the alpha particle and the atom. This makes the algorithm more complex. In addition, there are more parameters in this problem than in any previous problem with the possible exception of the accelerating car we studied in Chapter 3.

The geometry of the problem is shown in Figure 9.1. We idealize the atom as a sphere of radius R. The plum pudding of J. J. Thomson's model is inside the sphere. In the Rutherford experiment the collisions between alpha particles and gold atoms were obtained by shooting a pencil-like beam of alpha particles at a very thin gold foil. Figure 9.1 illustrates one possible trajectory for an alpha particle. The gold nucleus, having a positive charge like the alpha particle, repels it, and it is bent away from a straight line through a scattering angle θ. The quantity labelled b in Figure 9.1 is called the impact parameter. It is the distance from an axis through the center of the nucleus to the parallel trajectory of an incoming alpha particle. Once again, because of their low mass the electrons in the atom do not have a significant effect on the trajectory.

To do the calculations, we need an expression for the **E**-field due to the atom. With this in hand, the force on the alpha particle is just $q\mathbf{E}$, where q is the charge of the alpha particle. We assume the atom has a net positive charge Q located at (X, Y) and it has a radius R. Refer to Figures 9.1 and 9.2. We locate q at the point (x, y). The vector **r** is from Q to q—that is, from point (X, Y) to point (x, y).

Outside the atom, $r > R$, the electric field will be simply that due to a point charge Q; therefore

$$\mathbf{E} = k\frac{Q}{r^2}\frac{\mathbf{r}}{r}.$$

Inside the atom, $r \leq R$, at a distance r from its center, the magnitude of the electric field is

$$\mathbf{E} = \frac{r^3}{R^3}k\frac{Q}{r^2}\frac{\mathbf{r}}{r},$$

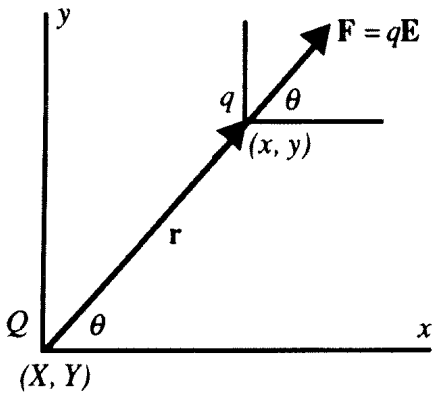


FIGURE 9.2 • Geometry of the Locations of the Target Atom (Q) and the Alpha Particle (q).

where the factor r^3/R^3 is a result of the fact that the alpha particle is only affected by the charge *inside* a sphere of radius r, and for a uniform distribution of charge the fraction of the total charge inside a sphere of radius r is r^3/R^3. We find the force on the alpha particle from the equation $\mathbf{F} = q\mathbf{E}$, and we find its acceleration from the equation $q\mathbf{E} = m\mathbf{a}$. Refer again to Figure 9.2, and notice that $\mathbf{F}$ will be along $\mathbf{r}$.

The geometry of the problem allows us to resolve the acceleration into x and y components. Thus

$$a_x = \frac{F_x}{m} = \frac{k}{m}\frac{Qq}{r^2}\cos\theta = \frac{k}{m}\frac{Qq}{r^2}\frac{x-X}{r} = \frac{k}{m}Qq\frac{x-X}{[(x-X)^2+(y-Y)^2]^{\frac{3}{2}}},$$

and

$$a_y = \frac{F_y}{m} = \frac{k}{m}\frac{Qq}{r^2}\sin\theta = \frac{k}{m}\frac{Qq}{r^2}\frac{y-Y}{r} = \frac{k}{m}Qq\frac{y-Y}{[(x-X)^2+(y-Y)^2]^{\frac{3}{2}}},$$

where m is the mass of the alpha particle. Newton's Third Law allows us to find A_x and A_y, the acceleration of the atom. Thus

$$MA_x = -ma_x$$

gives

$$A_x = -\frac{m}{M}a_x.$$

Likewise,

$$A_y = -\frac{m}{M}a_y.$$

Next let us assemble a list of the parameters we will need: the mass, M, and charge, Q, of the target atom; the mass, m, and charge, q, of the alpha particle; the radius, R, of the target atom; the energy (velocity) of the incoming alpha particle; the constant k in Coulomb's Law; and the impact parameter b. The variables are as follows: for the alpha particle x, y, v_x, v_y, a_x, and a_y; for the target atom X, Y, V_x, V_y, A_x, and A_y. Now we are ready to formulate the problem in terms of an algorithm. Algorithm 9.1 outlines the calculations we will make to study the motion of the alpha particle and the target atom.

ALGORITHM 9.1 / A Model of the Rutherford Scattering Experiment

STEP 1. Supply values for the following parameters:

- The radius of the atom, R
- The mass of the alpha particle, $m = 6.64 \times 10^{-27}$ kg
- The mass M of the target atom (mass number A times 1.66×10^{-27} kg amu^{-1})
- The charge on the electron, $e = 1.60 \times 10^{-19}$ C
- The ratio of the charge of the alpha particle to the charge of the electron (2)
- The ratio of the charge on the target atom to the charge of the electron (the atomic number Z)
- The constant k in Coulomb's Law, 8.99×10^9 Nm^{2}C^{-2}
- The impact parameter, b (see Figure 9.1)
- The initial velocity of the alpha particle; For 7 Mev alphas, $v = 2 \times 10^7$ ms^{-1}
- The step-size h, approximately 1×10^{-24} s

STEP 2. Determine the initial conditions for both the alpha particle and the target atom

- For the alpha particle: x, y, v_x, v_y
- For the target atom: X, Y, V_x, V_y

STEP 4. Define a constant N, where

$$N = \frac{2kQe}{m}.$$

STEP 3. Calculate the accelerations of the alpha particle and the atom.

- First, calculate the distance (or its square) between the alpha particle and the atom,

$$r = \sqrt{(x-X)^2 + (y-Y)^2}.$$

- Is $r > R$?
  - Yes. Then find the acceleration of the alpha particle from the **E**-field *outside* the atom and Newton's Second Law,

$$a_x = N\frac{x-X}{r^3},$$

and

$$a_y = N\frac{y-Y}{r^3}.$$

- No. Then find the acceleration of the alpha particle from the **E**-field inside the atom and Newton's Second Law,

$$a_x = N\frac{x-X}{r^3}\frac{r^3}{R^3} = N\frac{(x-X)}{R^3},$$

and

$$a_y = N\frac{y-Y}{r^3}\frac{r^3}{R^3} = N\frac{(y-Y)}{R^3}.$$

- Find the acceleration of the atom:

$$A_x = -\frac{m}{M}a_x,$$

and

$$A_y = -\frac{m}{M}a_y.$$

STEP 4. Calculate the half-step initial velocities:

$$v_x = v_{x0} + \frac{h}{2}a_x,$$

$$v_y = v_{y0} + \frac{h}{2}a_y,$$

$$V_x = V_{x0} + \frac{h}{2}A_x,$$

and

$$V_y = V_{y0} + \frac{h}{2}A_y.$$

STEP 5. Loop until alpha particle leaves the screen (while it is on the screen).

STEP 6. Tabulate or plot the positions of both the alpha particle and the atom.

STEP 7. Update the positions:

$$x \leftarrow x + hv_x,$$

$$y \leftarrow y + hv_y,$$

$$X \leftarrow X + hV_x,$$

and

$$Y \leftarrow Y + hV_y.$$

STEP 8. Update the accelerations using the appropriate formulas for acceleration in STEP 3.
STEP 9. Update the velocities:

$$v_x \leftarrow v_x + ha_x,$$

$$v_y \leftarrow v_y + ha_y,$$

$$V_x \leftarrow V_x + hA_x,$$

and

$$V_y \leftarrow V_y + hA_y.$$

STEP 10. Update the time:

$$t \leftarrow t + h.$$

STEP 11. End of loop.
STEP 12. Quit.

PROBLEM 9.1 Simulation of Alpha Particle Scattering by an Atom • Write a program to implement Algorithm 9.1. Let the target atom be gold with a positive charge of 79 x (1.60 x 10^{-19}) C, a mass of 197 x (1.66 x 10^{-27}) kg, and a radius of 1 x 10^{-13} m. Try impact parameters larger and smaller than the radius. Assume that the kinetic energy of the alpha particles is 7 MeV, which means the initial velocity is approximately 2 x 10^7 ms^{-1}. Let *dom* = [−6, 6] x 10^{-13} m, and *ran* = [−4, 4] x 10^{-13} m. For the initial conditions of the atom, assume that $X = Y = 0$ and $V_x = V_y = 0$. For the initial conditions of the alpha particle, assume that $x = -6 \times 10^{-13}$ m, y is the impact parameter, $v_x = 2 \times 10^7$ ms^{-1}, and $v_y = 0$. Run the program for various values of the impact parameter and radii of the atom.

The output we obtained from our program is shown in Figure 9.3. The beam of alpha particles comes from the left, each line representing a single encounter with a gold atom that has a charge 79 times that of the electron and has a mass that is 197 amu. The recoil of the

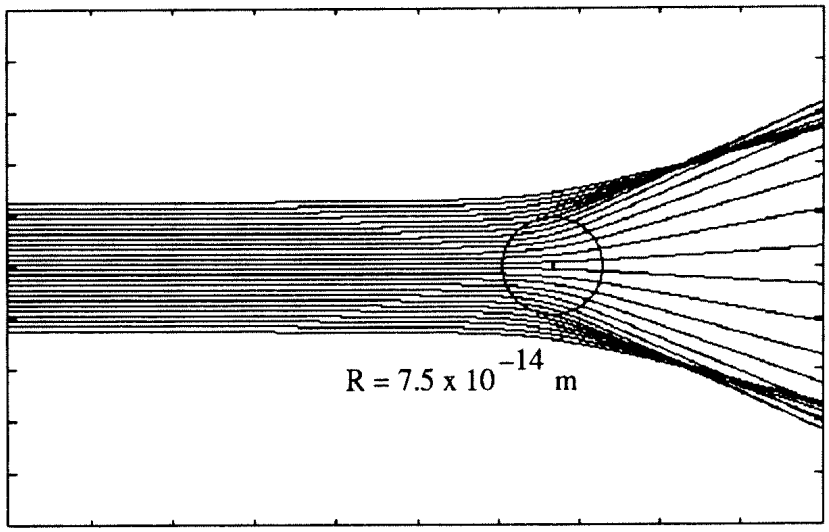


FIGURE 9.3 • Scattering of an Alpha Particle by a Gold (Au) Atom (Nucleus). Parameters: Gold Target, $Z = 79$, $A = 197$, $R = 7.5 \times 10^{-14}$ m, $h = 1 \times 10^{-22}$ s, *dom* = $[-6, 6] \times 10^{-13}$ m, and *ran* = $[-4, 4] \times 10^{-13}$ m.

gold atom is extremely small because the mass of the gold atom is much larger than the mass of the alpha particle, and in this case is not observable. The recoil may be seen in the picture at the beginning of the chapter. The alpha particle has the same energy as the ones in the original Rutherford experiments, namely 7 MeV.

9.3 Doing a Rutherford Scattering Experiment

In the original Rutherford scattering experiment a beam of alpha particles was shot at a gold foil. Geiger and Marsden performed the actual experiment, which consisted of looking through a microscope to count small flashes of light when, after passing though the gold foil, an alpha particle would strike a zinc-sulfide screen. (Rutherford's older eyes prevented him from doing the counting; his counts were usually too high.[1]) Today we would use a photomultiplier tube and a digital frequency counter instead of eyesight.

Before the experiment was carried out, Rutherford used his mathematical analysis skills and intuition to gain some idea of what to expect from his alpha particle experiments. He knew what a relatively large and massive sphere of positive charge would do to a pencil-like beam of incoming alpha particles, and that is precisely what we can discover with the computer. We are going to vary the size of the sphere of positive charge, and see how the scattering depends on this size.

PROBLEM 9.2 Computer Simulation of Rutherford Scattering • Run the program of Problem 9.1 with a gold atom as the target. Gold has an atomic number (Z) of 79 and an atomic weight (A) of 197. Use the parameters in the table below. Each set of parameters in this table

should be used with approximately 10 different values of the impact parameter, from approximately 10^{-13} m to 10^{-14} m. Each row in the table will produce a figure that shows how the alpha particle scattering depends on the impact parameter for that particular radius of the atom. You can save time by making use of the symmetry of the problem; if (x, y) is a point on the trajectory of the alpha particle, then $(x, -y)$ is also a point on the trajectory, and if (X, Y) is a point on the trajectory of the target, then $(X, -Y)$ is also on the trajectory. If you are working in a computing laboratory situation, have different groups each take a row in the table, print the results, and then compare them. After seeing the results, and knowing that, in fact, alpha particles are scattered backward, what would you conclude about the size of the target, which we may now properly call a nucleus?

Explain why a target nucleus with a radius of 10^{-11} meters does not scatter the alpha particle, but a nucleus with a radius of 10^{-14} meters is an effective scatterer. At this point you should understand why we now consider the atom to have two components; a tiny nucleus with a radius of approximately 10^{-15} m and an electron structure with a radius of approximately 10^{-10} m.

RADIUS (meters)	DOMAIN (meters)	RANGE (meters)	h (seconds)
10^{-11}	$[-1.5, 1.5] \times 10^{-10}$	$[-1, 1] \times 10^{-10}$	5×10^{-20}
10^{-12}	$[-1.5, 1.5] \times 10^{-11}$	$[-1, 1] \times 10^{-11}$	5×10^{-21}
10^{-13}	$[-1.5, 1.5] \times 10^{-12}$	$[-1, 1] \times 10^{-12}$	5×10^{-22}
10^{-14}	$[-1.5, 1.5] \times 10^{-13}$	$[-1, 1] \times 10^{-13}$	5×10^{-23}
2×10^{-15}	$[-1.5, 1.5] \times 10^{-14}$	$[-1, 1] \times 10^{-14}$	5×10^{-24}

After Rutherford did his mathematical analysis of the scattering, Geiger did experiments with real alpha particles and gold atoms, and he found that most of the scattering angles were small, just as one would expect from a plum-pudding model of the atom of the kind proposed by J. J. Thomson. Then, to provide a little research work for their student, E. Marsden, Rutherford proposed "Why not let him see if any alpha particles can be scattered through a large angle?" Rutherford did not believe they would find any alpha particles scattered backward from a plum pudding atom with a radius of about 10^{-10} m. After you have done the computer experiment above, you should also understand why he didn't expect any scintillations at large angles.

But Geiger and Marsden found alpha particles scattered backward! Rutherford describes his own thoughts when he was informed of this:

> It was quite the most incredible event that has ever happened to me in my life. It was almost as incredible as if you fired a 15-inch shell at a piece of tissue paper and it came back and hit you.[2]

If you do not understand Rutherford's emotions, run the computational experiment performed in Problem 9.2 once again. The only way to explain the experimental results is to assume that the nucleus is very tiny, approximately 1×10^{-15} m in radius, in a cloud of electrons with a radius of about 1×10^{-10} m. What fraction of the volume of an atom is occupied by its nucleus?

Geiger and Marsden verified experimentally all of the predictions made by Rutherford when he adopted the fresh hypothesis of a tiny nucleus. The model of the atom became a miniature solar system. The hypothesis of a tiny nucleus has never been abandoned, but it posed great difficulties for the explanation of spectral lines and classical models of atoms as miniature solar systems. A completely new field of physics, quantum mechanics, had to be born. Finally, we should not underestimate the extraordinary experimental work of Geiger and Marsden who, in the course of their experimental work, used eyesight and a microscope to count more than 100,000 scintillations.

9.3 Additional Problems

PROBLEM 9.3 Collisions • The solution to Problem 9.1 may be modified to do a variety of collision experiments. For example, try an alpha-alpha collision in which one moving alpha particle collides with one at rest. In this case we would pick the target nucleus to have $Z = 2$ and $A = 4$. For the remaining parameters use $R = 1 \times 10^{-20}$ m, $h = 1 \times 10^{-23}$ s, $dom = [-15, 15] \times 10^{-15}$ m, and $ran = [-10, 10] \times 10^{-15}$ m. Shoot the alpha particle with a speed of about 2×10^{7} ms^{-1} straight at the one at rest. Try some glancing shots. It is informative to plot the velocity of each of the particles as well as their coordinates. Scaling the speed by a (multiplicative) factor of 5×10^{-22} will keep the speeds on the graph.

PROBLEM 9.4 Conservation of Momentum • Use the solution to Problem 9.1 to demonstrate conservation of momentum. Begin with the target and the missile having the same mass, and shoot the missile at the target. Remember, the missile and the target must be charged if the collision is to work. In addition to graphing the trajectory of both missile and target, graph the linear momentum of each. Now vary their mass ratio and repeat the experiment. What happens when a massive missile strikes a target of small mass, or vice versa? For example, what happens when an alpha particle collides with another alpha particle? A proton? An electron? Is kinetic energy conserved in these experiments? Vary the force law so that the force is proportional to $1/r^4$ and repeat the experiments. Describe how the results of this force law differ from a $1/r^2$ law.

9.5 References

[1] T. H. Osgood and H. S. Hirst, "Rutherford and his Alpha Particles," *Physics History from AAPT Journals*, M. N. Phillips (American Association of Physics Teachers, 1985), 91.

[2] Quoted in A. P. French, *Newtonian Mechanics* (New York: W. W. Norton, 1971), 616. This book is an outstanding reference work for classical mechanics.

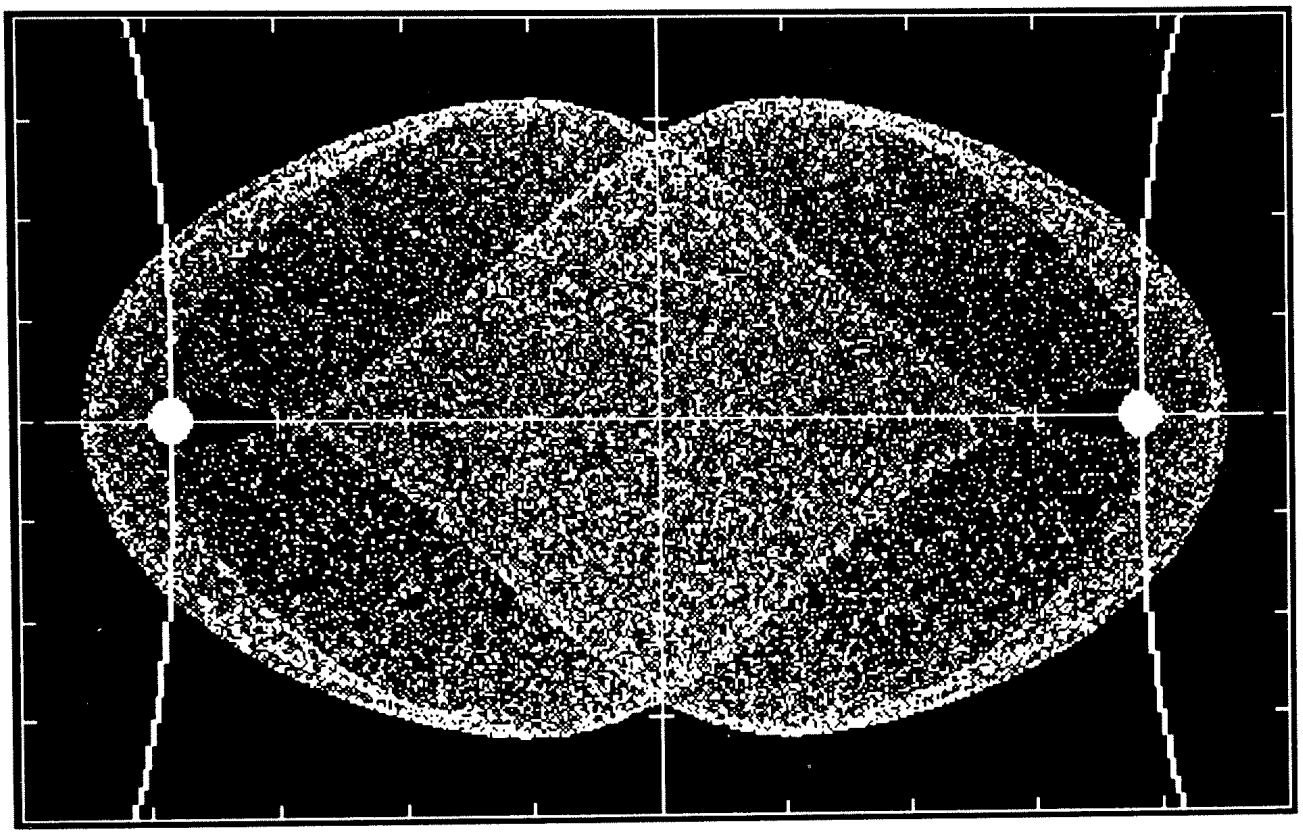
9.6 Program

PROGRAM 9.6.1 / Scattering of Alpha Particles by Gold Nuclei

```
! Algorithm 9.1.
LIBRARY "Comphys"
CALL GET_INFO (N,Mr)                                    ! Get parameters of the problem.
! ............
DECLARE DEF r                                           ! Calculate r, the distance
                                                          between q and Q.
DEF r = sqr((x-Xt)^2+(y-Yt)^2)
! ............
DEF ax                                                  ! Define the x acceleration of
                                                          q.
    IF r > Rt then
       LET ax = N*(x-Xt)/r^3
       ELSE                                             ! If alpha is inside the atom,
                                                          use a
       LET ax = N*(x - Xt)/Rt^3                         ! different formula for E (see
                                                          text).
    END IF
END DEF
!............
DEF ay                                                  ! Calculate the y acceleration
                                                          of q.
    IF r > Rt then
       LET ay = N*(y-Yt)/r^3                            ! y acceleration of alpha.
    ELSE
       LET ay = N*(y-Yt)/Rt^3
    END IF
END DEF
!............
DEF axt = -Mr*ax                                        ! x acceleration of target.
DEF ayt=-Mr*ay                                          ! y acceleration of target.
!............
INPUT PROMPT "Enter the radius of the atom." :Rt
INPUT PROMPT "Enter the step size, h." :h
CALL OPEN (xmin,xmax, ymin,ymax,wxmin,wxmax,wymin,wymax)  ! Open graphics window.
BOX CIRCLE -Rt, Rt, -Rt, Rt                             ! Draw a circle representing the
                                                          atom.
LET b = 1e-12                                           ! Prepare for loop to draw
                                                          trajectory of alpha
FOR I=1 to 10                                           ! particle for various impact
                                                          parameters, b.
    CALL Initial_Cond (x,y,Xt,Yt,vx,vy,Vxt,Vyt)         ! Enter the initial conditions.
    LET y = b/2^I                                       ! y will be the impact
                                                          parameter.
    Set cursor 2,5
    Print "Impact Parameter = ", y
    LET x = xmin
    LET vx = vx + ax*h/2                                ! Velocity calculation when
```

```
    LET vy = vy + ay*h/2                                    ! t is zero.
    LET Vxt = Vxt + h*axt/2                                 ! Velocity calculation when
    LET Vyt = Vyt + h*ayt/2                                 ! t is zero.
  DO                                                        ! This is the main loop.
    PLOT x,y                                                ! Plot the alpha particle
                                                              trajectory.
    PLOT x, -y                                              ! Plot the symmetric point.
    PLOT Xt,Yt                                              ! Plot the target nucleus
                                                              trajectory.
    PLOT Xt,-Yt
    LET x = x + h*vx                                        ! Calculate the new x position
                                                              of alpha.
    LET y = y + h*vy                                        ! Calculate the new y
                                                              position.of alpha.
    LET Xt = Xt + h*Vxt                                     ! Calculate the new x position
                                                              of atom.
    LET Yt = Yt + h*Vyt                                     ! Calculate the new y position
                                                              of atom.
    LET vx = vx + h*ax
    LET vy = vy + h*ay                                      ! Calculate velocities.
    LET Vxt = Vxt + h*axt
    LET Vyt = Vyt + h*ayt
    LET t = t + h                                           ! Increment the time.
  LOOP until (x < xmin or x > xmax) or (y < ymin or y > ymax)! Quit when alpha leaves
NEXT I                                                      ! the graphics window.
CALL Label (wxmin, wxmax, wymin, wymax)                     ! Print text on the graphic.
END                                                         ! End of the main program.
!....................
SUB GET_INFO (N,MR)                                         ! Get parameters of the problem.
 INPUT PROMPT "Enter the atomic number of the target atom. ":Z
 INPUT PROMPT "Input the atomic mass of the target atom. ": A
 LET k = 8.99e9                                             ! The constant k in Coulombs
                                                              Law.
 LET q1 = 2*1.60e-19                                        ! Charge of alpha particle.
 LET Z = z*1.60e-19                                         ! Charge of target nucleus.
 LET m = 4*1.66e-27                                         ! Mass of alpha particle.
 LET A= A*1.66e-27                                          ! Mass of target nucleus
 LET N = (k*q1*z)/m                                         ! Common multiplier in
                                                              acceleration calculations.
 LET MR = m/A                                               ! Mass ratio.
END SUB
!..........
SUB Initial_Cond(x,y,Xt,Yt,vx,vy,Vxt,Vyt)                   ! Enter the initial conditions.
 LET x = 0                                                  ! Start the alpha at (x,y).
 LET y = 0                                                  ! Approximate scale, e-15m
 LET Xt = 0                                                 ! Target starts at (0,0).
 LET Yt = 0
 LET v = 2e7                                                ! Speed of 7.7 Mev alpha
                                                              particle.
 LET vx = v                                                 ! Initial velocity in + x
                                                              direction.
```

```
 LET vy = 0
 LET Vxt=0                                          ! Target is at rest.
 LET Vyt = 0
END SUB
```



Electron Density

10 / The H_2^+ Ion[1]

10.1 Introduction

A review of the following topics may be helpful in this chapter:

- Coulomb's Law
- Newton's Second Law
- Kinetic and potential energy
- Atomic structure

In previous chapters we have examined the trajectories of particles in physical systems that involved either a single object or two objects. In some cases those systems have analytic solutions, but in other cases they do not. In this chapter we will study a system with three objects, a version of the so-called three-body problem. Unrestricted three-body systems have no analytic solutions; in fact it is possible to prove that no analytic solutions exist. In these

situations the computer becomes absolutely essential. In this chapter we will use the techniques of the preceding chapters to examine a three-body system, the ionized hydrogen molecule ion, H_2^+. At the end of the chapter we will suggest some projects you might like to undertake, all of which can be solved with our Feynman-Newton method of solving second-order differential equations. The spirit of this chapter is intended to be one of experimentation and exploration of more complex problems.

The three-body problem to be examined here is a hydrogen molecule, H_2, which has lost one electron. It is called "the once-ionized hydrogen molecule," and it is symbolized by "H_2^+." Hereinafter we will refer to it as the H_2 ion. It consists of two protons and one electron. Its parent molecule, H_2, exists throughout the atmosphere of the earth. Like its parent, the H_2 ion is a stable physical system; that is, left undisturbed by photons or collisions the three constituents of the H_2 ion remain together. The study of the H_2 ion will give some insight into why this molecule is stable. This in turn will lead to a deeper understanding of the covalent bond in which a shared particle, an electron in this case, ties or binds together the constituents of a molecule.

If the three particles in this system are simply placed in proximity to each other and given some initial velocities, most likely they will orbit each other briefly, one of the protons will be expelled from the system, and the electron will crash into the remaining proton, ending the life of the molecule. Because there are so many initial conditions that lead to this kind of instability, to begin we will revert to the simple expedient of assuming the protons are motionless, allowing the electron to orbit the two protons. For all practical purposes, nature ties the protons down because the mass of the proton is approximately 2000 times that of an electron, and therefore, the center of mass of the two protons, which is also the origin of our coordinate system, may be regarded as being at rest. With the protons tied in place by our assumptions, we will search for initial conditions that produce a trajectory for the electron that does not intersect the points where the protons are located. Stable trajectories must also make the system energetically stable. Once we have found a stable arrangement, then we will free the protons and see what happens. Finally, we will discuss briefly the relationship between our model and a quantum-mechanical model.

10.2 The H_2 Ion

Coulomb's Law and Newton's Second Law govern the motion of the electron in the H_2 ion. Coulomb's Law is

$$\mathbf{F} = k\frac{Q_1 Q_2}{r^2}\frac{\mathbf{r}}{r},$$

where $k = 8.99 \times 10^9\ \mathrm{NM^2C^{-2}}$. All of the charges in the H_2 ion have the same value (except for signs); thus $Q_1 = Q_2 = e = 1.60 \times 10^{-19}$ C. We will need the mass of the electron as well as its charge, and $m = 9.11 \times 10^{-31}$ kg.

The geometry of our system is illustrated in Figure 10.1. It is very similar to the geometry we used for satellite and projectile motion, but now there are two forces acting on the orbiting object, which means that the Feynman-Newton method will include more terms. We locate

the protons at points $(d, 0)$ and $(-d, 0)$, a distance $2d$ apart. The important thing to deduce from Figure 10.1 is how the components of the various accelerations are found. The acceleration, a_x, of the electron due to the proton at $(d, 0)$ is, for example,

$$a_x = -\frac{F_1 \cos\theta_1}{m} = -k\frac{e^2}{m}\frac{\cos\theta_1}{(x-d)^2 + y^2} = -k\frac{e^2}{m}\frac{x-d}{\left[(x-d)^2 + y^2\right]^{\frac{3}{2}}}.$$

To find the acceleration due to the force of the proton at $(-d, 0)$, in the equation above replace $x - d$ with $x + d$ to obtain

$$a_x = -\frac{F_2 \cos\theta_2}{m} = -k\frac{e^2}{m}\frac{\cos\theta_2}{(x+d)^2 + y^2} = -k\frac{e^2}{m}\frac{x+d}{\left[(x-d)^2 + y^2\right]^{\frac{3}{2}}}.$$

There is a similar expression for a_y, which may be obtained by replacing the $x \pm d$ terms in the numerators by y. In what follows it will be convenient to set

$$C = k\frac{e^2}{m}.$$

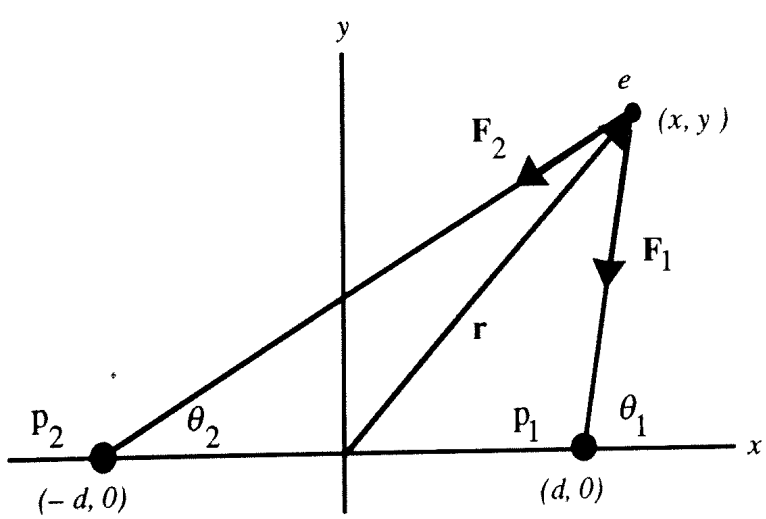
We are ready to give an algorithm to find the orbit of the electron in the H_2 ion.

ALGORITHM 10.1 / The Orbit of the Electron in the H_2 Ion

STEP 1. Assign values to the following parameters:

- The charges of each of the three particles, Q_1, Q_2, and e
- The mass of the electron

FIGURE 10.1 • The Geometry of the Displacement and Force Vectors in the H_2 Ion.



- The constant k in Coulomb's Law
- The distance, $2d$, between the protons in the molecule
- The step-size h

STEP 2. Assign initial values to x, y, v_x, and v_y for the electron.
STEP 3. Calculate the x and y components of the acceleration, a_x and a_y.

$$a_x = -C\left\{\frac{x-d}{\left[(x-d)^2+y^2\right]^{\frac{3}{2}}} + \frac{x+d}{\left[(x+d)^2+y^2\right]^{\frac{3}{2}}}\right\},$$

and

$$a_y = -C\left\{\frac{y}{\left[(x-d)^2+y^2\right]^{\frac{3}{2}}} + \frac{y}{\left[(x+d)^2+y^2\right]^{\frac{3}{2}}}\right\},$$

with

$$C = k\frac{e^2}{m}.$$

STEP 4. Find the half-step values for v_x and v_y:

$$v_x = v_{x0} + \frac{h}{2}a_x$$

and

$$v_y = v_{y0} + \frac{h}{2}a_y.$$

STEP 5. Loop until key is pressed.
STEP 6. Plot the point (x, y) on the trajectory of the electron.
STEP 7. Update the positions:

$$x \leftarrow x + hv_x$$

and

$$y \leftarrow y + hv_y.$$

STEP 8. Calculate the accelerations using the formulas in STEP 3.
STEP 9. Update the velocities:

$$v_x \leftarrow v_x + ha_x$$

and

$$v_y \leftarrow v_y + ha_y.$$

STEP 10. Increment the time:

$$t \leftarrow t + h.$$

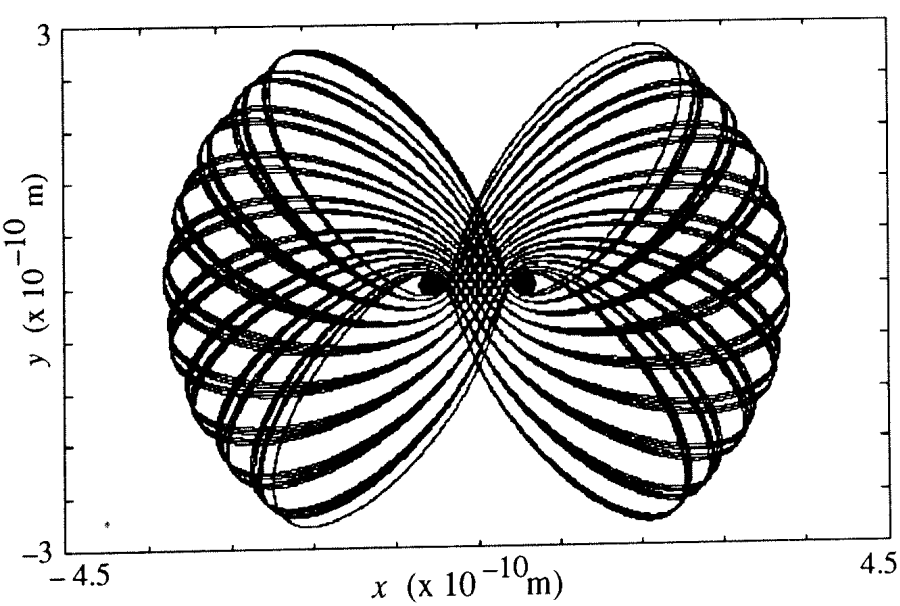
STEP 11. End of the loop begun in STEP 5 (loop until key is pressed).
STEP 12. Quit.

PROBLEM 10.1 Trajectory of the Electron in the H_2 Ion • Write a program to plot the trajectory of the electron in the H_2 ion. Assign the following values to the parameters: $d = 0.535 \times 10^{-11}$ m, $x_0 = -y_0 = -2.4 \times 10^{-10}$ m, $v_{x0} = v_{y0} = -2.8 \times 10^5$ ms^{-1}, $h = 5 \times 10^{-19}$ s, $dom = [-4.5, 4.5] \times 10^{-10}$ m, and $ran = [-3, 3] \times 10^{-10}$ m.

The trajectory of the electron for the set of parameters and initial conditions in Problem 10.1 is shown in Figure 10.2. The distance $d = 5.35 \times 10^{-11}$ m, which is half the distance between the protons, comes from quantum mechanics and experimental data.

PROBLEM 10.2 Another Trajectory of an Electron in an H_2 Ion • Try this set of parameters: $x_0 = 0$, $y_0 = 2 \times 10^{-10}$ m, $v_{x0} = 1.725 \times 10^5$ ms^{-1}, $v_{y0} = 0$, $dom = [-3, 3] \times 10^{-10}$ m, $ran = [-2.1, 2.1] \times 10^{-10}$ m, and $h = 1 \times 10^{-19}$ s. Be patient. How does this trajectory differ from the earlier ones?

FIGURE 10.2 • One Trajectory of an Electron in a Once-Ionized Hydrogen Molecule. Parameters: $d = 5.35 \times 10^{-11}$ m, $x_0 = -y_0 = -2.4 \times 10^{-10}$ m, $v_{x0} = vy_0 = -2.85 \times 10^5$ ms^{-1}, $dom = [-4.5, 4.5] \times 10^{-10}$ m, and $ran = [-3, 3] \times 10^{-10}$ m. The electron energy is −10.1 eV.



PROBLEM 10.3 One Last Trajectory • This time let $d = 0.535 \times 10^{-10}$ m, $h = 0.2 \times 10^{-19}$ s, $x_0 = y_0 = 0$, $v_{x0} = v_{y0} = -1.5 \times 10^6$ ms^{-1}, $dom = [-9, 9] \times 10^{-11}$ m, and $ran = [-6, 6] \times 10^{-11}$ m. Which of the three trajectories you have plotted do you think produces the most stable H_2 ion?

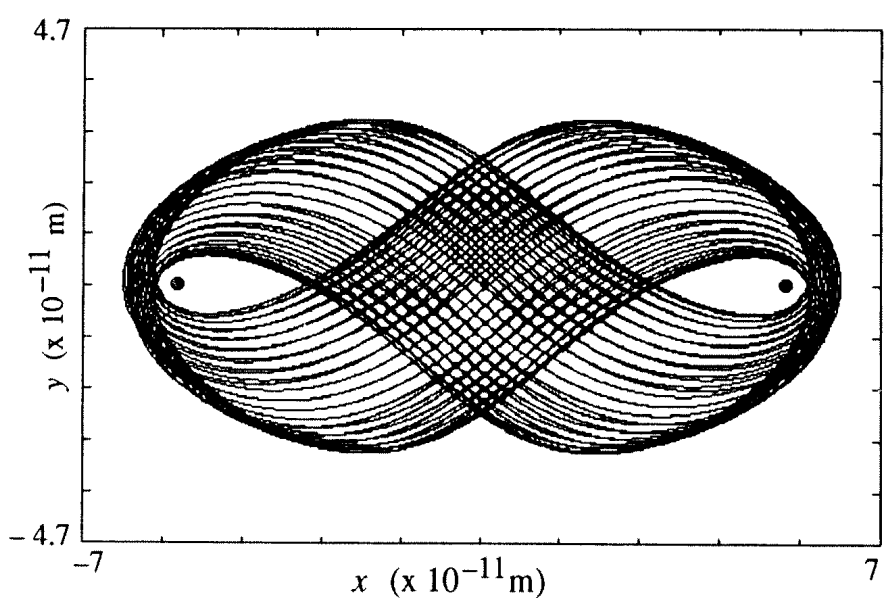
PROBLEM 10.4 Trajectory of the Electron in Velocity Space • Plot the set of points (v_x, v_y) for the parameters and initial conditions given in Problem 10.1. Your window coordinates must now be speeds. Can you make a one-to-one correspondence between points in the orbit and points on the trajectory of the electron in velocity space?

10.3 Stability of the H_2 Ion

10.3.1 Energy

Figure 10.3 shows a trajectory that is significantly different than the one shown in Figure 10.2 and the one obtained in Problem 10.1. You should recognize it as the trajectory obtained in Problem 10.3. There are two essential differences that you should note. First, the trajectory in Figure 10.3 is wound more tightly around the protons than the trajectory in Figure 10.2. Second, while certainly not obvious, the energy of the electron in Figure 10.2 is – 10.1

FIGURE 10.3 Another Electron Trajectory in an H_2 Ion. Parameters: $d = 5.35 \times 10^{-11}$ m, $h = 1 \times 10^{-20}$ s, $x_0 = y_0 = 0$, $v_{x0} = v_{y0} = 1.5 \times 10^6$ ms^{-1}, $dom = [-7, 7] \times 10^{-11}$ m, and $ran = [-4.67, 4.67] \times 10^{-11}$ m. This choice of parameters makes the electron energy, E, –41 eV.



electron volts, while the energy of the electron in Figure 10.3 is −41 eV. These energies were calculated from the following formulas:

$$U = ke^2\left[\frac{1}{\sqrt{(x-d)^2+y^2}} + \frac{1}{\sqrt{(x+d)^2+y^2}}\right],$$

$$K = \frac{1}{2}mv^2,$$

and

$$E = \frac{K+U}{1.6\times 10^{-19}}.$$

In these formulas U is the electrical potential energy, K is the kinetic energy, and E is the total energy *of the electron*. If K and U are in joules, then, in the last formula, E will be in electron volts. Remember that the e^2 term is the product of a minus charge (the electron) and a positive charge (the proton). Thus in this case, $U < 0$. In a moment we will proceed to a discussion of the electron energy and the covalent bonding that the shared electron provides. However, it may be worthwhile to point out that adding to your program the instructions necessary to calculate E and see that it is conserved while the electron orbits the protons provides a check on the validity of the model.

The formula for U, given above, comes from the fact that the electrostatic potential energy of a charge Q_1 in the vicinity of another charge Q_2 is defined by the formula

$$U = k\frac{Q_1Q_2}{r},$$

where k is the constant in Coulomb's Law. Refer to your textbook or to Chapter 5 of this book. If the charges have opposite signs, then U is negative, but it becomes less negative as r increases. Just like a baseball thrown into the air, if one charge moves away from the other, the potential energy increases while the kinetic energy decreases. If $E < 0$, the two charges will ultimately come together, perhaps in stable orbits. If $E \geq 0$, the two charges will separate indefinitely. The importance of the energy calculation should now be apparent. For the H_2 ion to be viable, its total energy must be negative.

PROBLEM 10.5 Electron Trajectories in a Real H_2 Ion • Try to find a stable orbit for which the energy of the electron is −29.8 eV. To do this, place the electron at some point in the vicinity of the two protons. Calculate its potential energy at this point. Then give it only enough kinetic energy to make its total energy −29.8 eV. Use the initial conditions to point it in what appears to be a suitable direction, not straight at a proton, and let the system evolve.

PROBLEM 10.6 Velocity-Space Diagram for the −29.8 eV Electron • Construct a velocity-space diagram for the system obtained in Problem 10.5. Correlate points in velocity space with those in ordinary space.

10.3.2 The Covalent Bond

Another condition for the survival of the H_2 ion is that the electron must bond the protons together. For example, if the electron is somewhere to the right of p_1 (see Figure 10.1) the protons will experience a force of repulsion as a result of their own charge, and p_1 will also experience a net force to the right because of the attraction of the electron, but the net force on p_2 is to the left. In other words, the system is being pulled apart, the electron may combine with the proton p_1 to make a hydrogen atom, and the potential energy of the system will turn into kinetic energy as the constituents fly apart.

On the other hand, if the electron is between the two protons, at the origin of Figure 10.1 for example, the attractive forces of opposite charges binds the system together. Therefore, if averaged over time the electron is between the protons more than away from them, then the system will also be stable from the point of view of the forces.

Refer once again to Figure 10.1. Consider the electron to be at a point in the first quadrant of Figure 10.1. The protons repel each other only in the $\pm x$ direction, but the force of the electron on the protons has both x and y components. In cases in which the orbit is symmetric with respect to the x-axis, however, the electron will spend as much time above the x-axis as below it, so the y components of the forces of the electron on the protons will average to zero, and we may disregard them. In what follows we are referring only to the *x-components* of the forces.

The procedure that follows describes how we will deal with the *x-components* of the forces.

- Put the electron at some point in the first quadrant.
- Calculate the *x-component* of the net force of p_1 and e on p_2. Call this F_2.
- Calculate the *x-component* of the net force of p_2 and e on p_1. Call this F_1.
- If F_1 is larger than F_2, then p_1 is being pulled away from p_2, and the electron is not binding the ion together.
- If F_1 is smaller than F_2, then p_1 is being pulled toward p_2, and the electron has the effect of pulling the protons together; it is binding the ion. Draw a picture to convince yourself that this is true.
- Starting from $x = 0$ with y fixed, test in succession each point (x, y) on the line to the right, find the point where F_1 changes from $F_1 \leq F_2$ to $F_1 > F_2$, and plot this point. All points including this point and points to the right are "non binding" points.
- If the point just plotted is called (x, y), the symmetry of the ion allows you to also plot the points $(x, -y)$, $(-x, y)$, and $(-x, -y)$.
- Repeat this row process for all possible y values in the first quadrant.

You will end up with two more or less vertical lines on the graph of the ion. If the electron is between these two lines, it is bonding; otherwise it is not binding. In the calculation ke^2/m is a factor of every expression, so it may be cancelled. The illustration for the chapter opening shows, among other things, these two lines for the once ionized hydrogen ion.

Now we know where the electron produces a binding effect rather than pulling the protons apart. An examination of the figures will reveal that the electron is in the bonding region only part of the time. We can assume that if, on the average, the electron spends more

time in the bonding region than in the non-binding region, then the ion is stable. The figure for the chapter heading was obtained by plotting points on the trajectory of a –41 eV electron at random time intervals. The density of points is proportional to the time spent there. You can see that in the case of a –41 eV ion the electron has a strong bonding influence. Indeed, we have succeeded in explaining the covalent (or shared electron) bond of the H_2^+ ion.

You might suspect that as soon as the electron is not in the binding region the ion will disintegrate. That is not the case. We have modified our programs to allow all three particles to move, and there are solutions that last for many orbits. One of them is a closed trajectory that looks like an "infinity" symbol or an "8" lying on its side. This solution is stable because the protons, being much more massive, accelerate very little during a complete orbit of the electron, and they do not fly apart instantaneously. Because our computer model of the molecular ion does not disintegrate immediately, a great deal of computer time is required to find stable solutions. This model of the ion runs for a long period of time before the model crashes. Furthermore, it is not clear whether the crash is a result of accumulated round-off errors or truncation errors, or inherent in the choice of initial conditions. For example, we have not succeeded in finding a solution that lasts for days. Can you?

PROBLEM 10.7 The Stability of the –10.1 eV Ion • Construct the boundary between the bonding and non-bonding regions for the ion described by Figure 10.2. Add the code to plot points on the trajectory at random intervals, and form some conclusions about the stability of the –10.1 eV ion. To get enough dots, you may need to allow your program to run for a day or two. Weekends are ideal.

10.3.3 Quantum Mechanics and the H_2 Ion

The most significant problem with our model of an H_2 ion comes from classical physics itself. According to classical physics, the accelerating electron emits electromagnetic energy, and it will radiate this energy until it spirals and crashes into one proton or the other. What twentieth century physicist Niels Bohr did to save the hydrogen atom from extinction (it too contains an accelerating electron) was to postulate that in some orbits the electron does not radiate. (Clearly, it is not the hydrogen atom or molecule that is endangered, it is our models for these systems that become extinct.) Bohr's hypothesis was another important step toward the quantum-mechanical theory that physicists use today to describe the behavior of molecules, atoms, and smaller particles.

To save our model of the once-ionized hydrogen ion from extinction, we shall do something similar to what Niels Bohr did. According to quantum mechanics, the electron in an atom is not a tiny point, it is spread out over an area resembling the illustration at the opening of this chapter. We will assume that the spread-out quantum-mechanical electron and our classical time-average electron are approximately the same, and therefore the –41 eV molecular hydrogen ion is stable, while the –10.1 eV ion is unstable. In fact, the electron probability density function from quantum mechanics is not altogether dissimilar to the pattern of dots shown in the chapter opening illustration.[2]

10.4 Additional Problems

In the spirit of this chapter, the following problems require a major programming effort, above and beyond the relatively simple programs in this book. These problems are meant to encourage you to explore the methods of last several chapters on your own and in some depth. They are more like term projects than overnight problems, and no solutions are provided. They all are challenging.

PROBLEM 10.8 Orbit of a Satellite about a Mascon • The methods of the last several chapters may be modified to study the orbit of a satellite about a planet with a simple non-uniformity in its distribution of mass. Suppose that the earth were perfectly spherical except for a single mountain having a certain fraction of the earth's mass. The mountain would be called a mascon, a concentration of mass, a label that seems to have originated in the first manned flights to the moon. Modify the solution to Problem 10.1 to plot the orbit of a satellite around the earth assuming that the mascon whose mass is one tenth of the mass of the earth is located at the surface of the earth. (Hint: one proton becomes the earth, the other its mascon). You must pick initial conditions that do not allow the satellite to crash into the earth. To begin, assume that the earth does not rotate. Later you may try studying the orbit if the earth rotates. Try to predict the nature of the orbit before running the program.

PROBLEM 10.9 Binary Stars • The double star system Sirius A and B has an orbital period of 50 years and the length of the major axis is 20 au with the Sirius A at one focus. (An au, astronomical unit, is the distance from the earth to the sun, 1.49×10^{11} m.) The mass of Sirius A is about two solar masses, and the mass of Sirius B is half that of A. Model this binary system with a computer program.[3]

PROBLEM 10.10 Putting a Satellite into Orbit • Model the problem of putting a satellite in an earth orbit. Can you project an object into an orbit that does not intercept the earth's surface if you have a single-stage rocket? You are allowed to use a two-stage rocket. Each stage will require its own program code, with the initial conditions of the second stage being the final conditions of the first stage. Assume you have suitable rockets to provide the thrust, and that these rockets can produce the initial conditions for each subsequent stage.

PROBLEM 10.11 Helium Atom • Construct a model of the helium atom that is a three-body system consisting of a nucleus and two electrons. The nucleus is very massive relative to the electron system and may assumed to be at rest. The nucleus has two protons and two neutrons.

PROBLEM 10.12 Model the Orbit of Halley's Comet • Construct a model of the orbit of Halley's Comet. Its period is approximately 75 years and the semi-major axis of the orbit is 17.8 au. What percentage of the time is it more than five au from the sun? Does the orbit of the earth have the potential of intersecting the orbit of Halley's Comet? The problem that you will run into in this and similar problems is the exceedingly long time intervals when Halley's Comet is several au away from the sun, and nothing much happens. Can you think of a solution to this problem? Might changing the step-size h to become greater when the

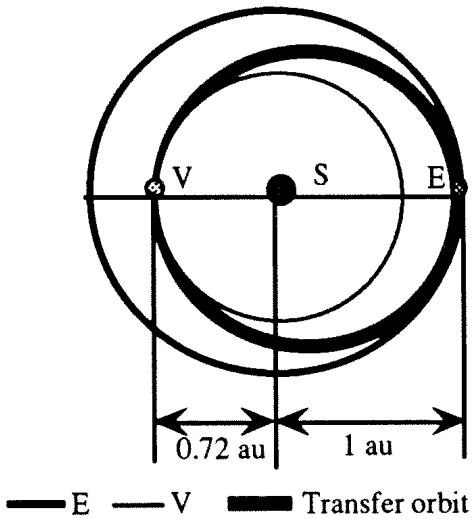


FIGURE 10.4 • Sending a Space Probe to Venus.

velocity is less, speed up the calculations? One way to modify the step-size h is as follows: Use h' where

$$h' = \frac{h}{\sqrt{v_x^2 + v_y^2}},$$

and h is chosen by experiment so that h' gives acceptable results when the velocity is large, close to the sun. Note that h' becomes larger when the comet moves slowly.

PROBLEM 10.13 Interplanetary Space Probes[4] • We wish to send a probe from earth to Venus. Assume circular orbits with Venus at 0.72 au from the sun and earth at one au. Refer to Figure 10.4. To carry out the transfer, the space probe must be given a sudden retardation at point E to put it in the transfer orbit with a semi-major axis of 1.72 au, and another sudden retardation at point V to put it in the orbit with Venus. This is a challenging problem. You may wish to try a variable step size similar to the one described in Problem 10.12.

PROBLEM 10.14 Fly By • Model a "fly by" in which a planetary space probe goes by a massive planet, picking up energy as it goes. The greater speed that the probe has after the fly-by shortens the length of its journey to another planet. These problems can get very tricky when the probe is close to the massive planet, and some way to monitor the energy of the system may be useful to ensure acceptable results.

10.5 References

[1] E. Huggins, *Graphical Mechanics* (Wentworth, NH: COMPress: 1979), 117. Readers of Huggins' book will recognize my deep indebtedness to Huggins and his student, Bob Piela, for their work on the H_2^+ ion.

[2] R. A. Alberty, *Physical Chemistry* (New York: John Wiley & Sons, 1987), 439. Almost any textbook on physical chemistry will discuss the quantum-mechanical model of the H_2^+ ion.
[3] A good reference for astronomy and astrophysics problems at the level of the first-year physics students is E. V. P. Smith and K.C. Jacobs, *Introductory Astronomy and Astrophysics* (Philadelphia: W. B. Saunders, 1973).
[4] A. P. French, *Newtonian Mechanics* (New York: W. W. Norton, 1971), 593.

10.6 Programs

PROGRAM 10.6.1 / Electron Trajectory in the H_2^+ Ion

```
! Algorithm 10.1, Problem 10.1, 10.2, etc.
! Case I: stable H ion-d = 0.535e-10, h = 1e-20, x=0,y=0,
! vx = vy = 1.5e6.
! ran = [-0.6,0.6]e-10 m.

! Case II: non-stable H ion-d=0.535e-10, x = -2.4e-10,
! y = 2.4e-10, vx = -2.8e5, vy = -2.8e5, h = 4e-19.
! ran = [-3,3] e-10 m
!
! Closed Orbit x = 0, y = 2e-10, vx = 1.725e5, vy = 0.
! ran = [-2.1,2.1]e-10 m.
Library "ComPhys"                                        ! Graphics subroutines.
LET d = 0.535e-10                                        ! Proton distance to origin.
LET h = .5e-18                                           ! Step size h.
LET k = 8.987e9                                          ! Constant in Coulomb's Law.
LET q = 1.602e-19
LET m = 9.110e-31                                        ! Mass of the electron.
LET C = (k*q^2)/m                                        ! Program constant.
CALL Initial_Cond (x,y,vx,vy)                            ! Enter the initial conditions.
CALL OPENB (xmin, xmax, ymin, ymax, wxmin, wxmax, wymin, wymax)
LET rsq1 = (x-d)^2 + y^2                                 ! Distance to proton one.
LET rth1 = rsq1^(3/2)                                    ! [x^2 + y^2]^3/2.
LET ax1 = -(x-d)/rth1                                    ! Acceleration due to p1.
LET ay1 = -y/rth1
LET rsq2 = (x+d)^2 + y^2                                 ! distance to p2 (squared).
LET rth2 = rsq2^(3/2)                                    ! to the three-halves.
LET ax2 =-(x+d)/rth2                                     ! Acceleration in the x
                                                           direction!
LET ay2 =-y/rth2                                         ! Acceleration is the y
                                                           direction.
LET ax = C*(ax1 + ax2)
LET ay = C*(ay1 + ay2)
LET vx = vx + h/2*ax                                     ! Half-step velocities.
LET vy = vy + h/2*ay
DO                                                       ! This is the main loop.
   LET n = n+1
   PLOT x,y;                                             ! Plot and connect the points on
                                                           the trajectory.
```

```
   CALL FEYNMAN_NEWTON (x, y, vx, vy,h,d,C)                  ! Calculate the next point.
   IF n > 100 then                                           ! Calculate and print the
                                                               electron energy.
      LET PE = -k*Q*Q*(1/sqr((x-d)^2 + y^2) + 1/sqr((x+d)^2 + y^2))
      LET KE = 0.5*m*(vx^2 + vy^2)
      LET E = (PE + KE)/1.602e-19
      SET CURSOR 1,5
      PRINT using "Total Energy = ###.##":E
      LET n = 0
   ELSE
   END IF
   IF key input then EXIT DO                                 ! This is a way out of the
LOOP                                                         ! loop.
CALL Label (wxmin, wxmax, wymin, wymax)
END                                                          ! End of the main program.
!...........
SUB Initial_Cond(x,y,vx,vy)                                  ! Enter the initial
    INPUT Prompt "x ":x                                      ! conditions of the electron.
    INPUT Prompt "y ":y
    INPUT Prompt "vx ":vx
    INPUT Prompt "vy ":vy
END SUB
!....................
SUB Feynman_Newton (x, y, vx, vy,h,d,C)
    LET x = x + h*vx                                         ! Calculate the new x position.
    LET y = y + h*vy                                         ! Calculate the new y position.
    LET rsq1 = (x-d)^2 + y^2                                 ! Distance to proton one.
    LET rth1 = rsq1^(3/2)
    LET ax1 = -(x-d)/rth1                                    ! Calculate the accelerations.
    LET ay1 = -y/rth1
    LET rsq2 = (x+d)^2 + y^2
    LET rth2 = rsq2^(3/2)
    LET ax2 =-(x+d)/rth2                                     ! Acceleration in the x
                                                               direction!
    LET ay2 =-y/rth2                                         ! Acceleration in the y
                                                               direction.
    LET ax = C*(ax1 + ax2)
    LET ay = C*(ay1 + ay2)
    LET vx = vx + h*ax                                       ! Velocity calculations.
    LET vy = vy + h*ay
END SUB
```

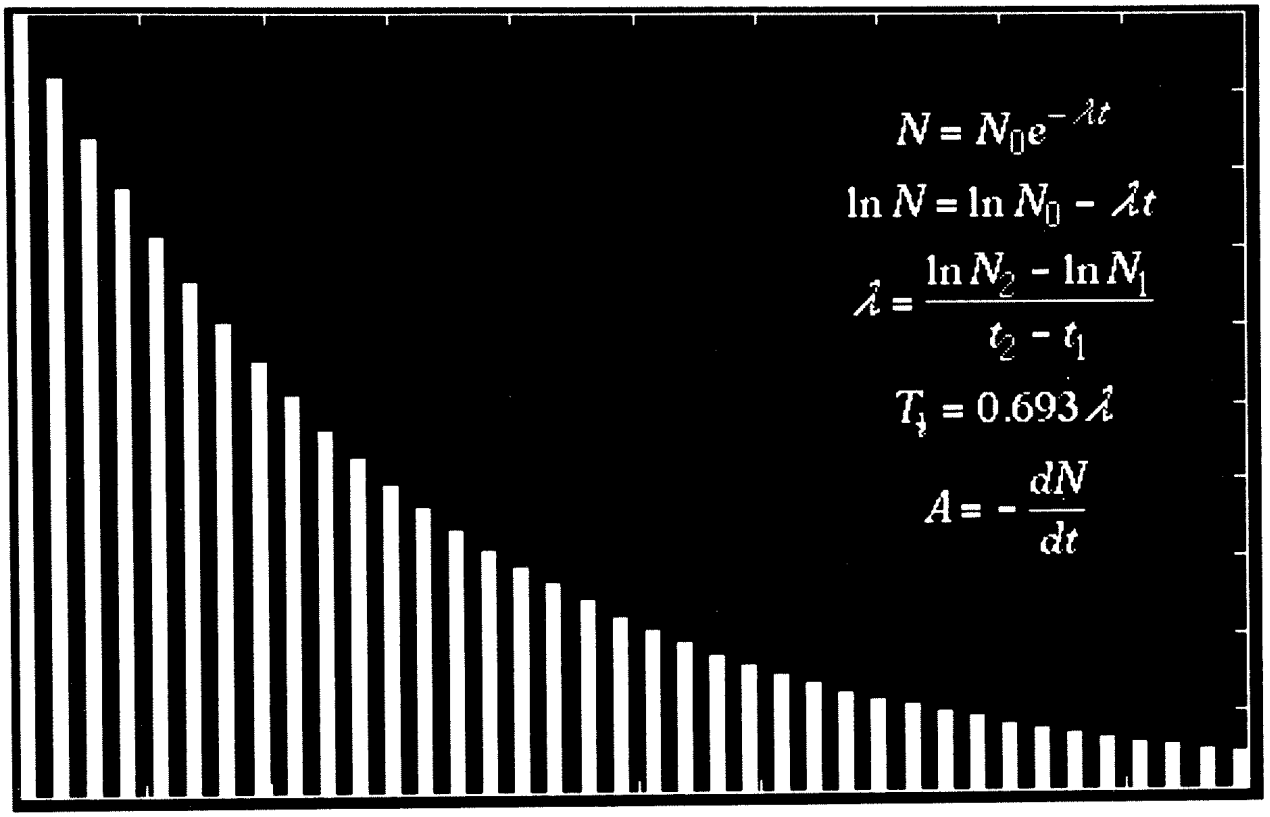
PROGRAM 10.6.2 / Electron Density Distribution

```
! Problem 10.7.
! Case I: stable H ion-d = 0.535e-10, h = 0.4e-19, x=0,y=0,
! vx = vy = 1.5e6.
! ran = [-0.6,0.6]e-10 m.
!
```

```
! Case II: non-stable H ion-d=0.535e-10, x = -2.4e-10,
! y = 2.4e-10, vx = -2.8e5, vy = -2.8e5, h = 4e-19.
! ran = [-3,3] e-10 m.
!
! Closed Orbit x = 0, y = 2e-10, vx = 1.725e5, vy = 0.
! ran = [-2.1,2.1] e-10 m.

LIBRARY "Comphys"                                        ! Also see comments in
Let d = 0.535e-10                                        ! Program 10.7.1.
Let h = 1e-20
LET k = 8.987e9
LET q = 1.602e-19
LET m = 9.110e-31
LET C = (k*q^2)/m
CALL Initial_Cond (x,y,vx,vy)                            ! Enter the initial conditions.
CALL OPENB (xmin, xmax, ymin, ymax, wxmin, wxmax, wymin, wymax)
!................! This routine takes a lot of time, be patient.
LET dy1 = (ymax-ymin)/200                                ! Draw the boundary between
LET dx1 = (xmax-xmin)/400                                ! "binding" and "non-binding"
LET y1 = dy1                                             ! regions.
DO                                                       ! Loop over y.
   LET x1 = dx1
   DO                                                    ! Loop over x
      LET rsq1 = (x1-d)^2 + y1^2                         ! Distance to proton one.
      LET rth1 = rsq1^(3/2)                              ! [r^2]^3/2.
      LET ax1 = (x1-d)/rth1                              ! p1 acceleration due to e.
      LET rsq2 = (x1+d)^2 + y1^2
      LET rth2 = rsq2^(3/2)
      LET ax2 =(x1+d)/rth2                               ! p2 acceleration due to e.
      LET axp1 = 1/(2*d)                                 ! p1 acceleration due to p2.
      LET axp2= -axp1                                    ! Newton's Third Law.
      LET ax1= ax1 + axp1                                ! Acceleration of p1.
      LET ax2 = ax2 +axp2                                ! Acceleration of p2.
      IF ax1>ax2 then                                    ! Is force attractive or
         PLOT x1,y1                                      ! repulsive.
         PLOT -x1,y1                                     ! Plot symmetric points.
         PLOT x1,-y1
         PLOT -x1,-y1
         EXIT DO                                         ! Get out of the loop.
      ELSE
         LET x1 = x1 + dx1                               ! Try the next x.
      END IF
   LOOP until x1=>xmax                                   ! When to quit.
   LET y1 = y1 + dy1                                     ! Try the next row.
LOOP until y1=>ymax                                      ! Quit when all rows have been
                                                           tests.
!.................................
LET rsq1 = (x-d)^2 + y^2                                 ! See Program 10.6.1.1 for
LET rth1 = rsq1^(3/2)                                    ! comments. Most of the rest
LET ax1 = -(x-d)/rth1                                    ! of this program is a repeat
LET ay1 = -y/rth1                                        ! of that program.
LET rsq2 = (x+d)^2 + y^2
```

```
LET rth2 = rsq2^(3/2)
LET ax2 =-(x+d)/rth2                                    ! Acceleration in the x
                                                          direction!
LET ay2 =-y/rth2                                        ! Acceleration is -g in the y
                                                          direction.
LET ax = C*(ax1 + ax2)
LET ay = C*(ay1 + ay2)
LET vx = vx + h/2*ax                                    ! Velocity calculation when
LET vy = vy + h/2*ay                                    ! t is zero.
RANDOMIZE                                               ! Randomly ample trajectory.
DO                                                      ! This is the main loop.
   LET n = INT(300*rnd)                                 ! Pick a random number between
   If n = 150 then plot x,y                             ! 0 and 299.
                                                        ! Plot a random point on the
                                                          trajectory.
   CALL FEYNMAN_NEWTON (x, y, vx, vy,h,d,C)             ! Calculate the next point.
   IF key input then EXIT DO
LOOP                                                    ! The main trajectory loop.
CALL Label (wxmin, wxmax, wymin, wymax)
END                                                     ! End of the main program.
! Add SUB Initial_Cond(x,y,vx,vy) from Program 10.6.1.
! Add SUB Feynman_Newton (x, y, vx, vy,h,d,C) from Program 10.6.1.
```



Radioactive Decay

11 / Using Random Numbers to Model Radioactivity

11.1 Introduction

A review of the following topics may be helpful in this chapter:

- Structure of the nucleus, atomic mass, atomic number
- Radioactive decay, half-life, radioactive series
- Alpha particles, beta particles, and gamma rays

We will provide our own very brief review. The substances with which we are familiar are constructed from 92 different kinds of atoms called elements. Atoms, in turn, consist of a tiny nucleus and a system of electrons. The nuclei of some atoms are unstable; some spontaneously emit *alpha* particles, others emit *beta* particles, and still others emit *gamma* rays, which are high energy electromagnetic waves. All of these atoms are said to be *radioactive*. Before

the spontaneous emission of an alpha or a beta particle, the atom is one element, uranium for example; after the emission, it is another element, thorium. In this case the uranium nucleus is called the *parent* and the thorium is called the *daughter*. Thorium is itself radioactive and decays into another element. The sequence of such a set of radioactive elements is called a *decay scheme* or a radioactive series. (It is unfortunate for physicists that all such decay schemes end in lead rather than gold or platinum.) The elements as we know them were synthesized either in the Big Bang beginning of the universe, in exploding stars called supernova, or in the particle accelerators in the laboratories of nuclear physicists.

For our purposes, the single most important fact about a radioactive nucleus is that the *probability* of the nucleus emitting a particle or a wave is *constant*, completely independent of when or where the nucleus was formed. In other words, radioactive decay is random. The probability of the radioactive decay of a uranium nucleus produced five billion years ago in some star is the same as the uranium nucleus formed in Supernova 1987, a star whose explosion we observed in the Large Magellanic Cloud in 1987. It is a much different situation for human beings. An examination of mortality tables will show that our mortality depends rather dramatically upon our age. Perhaps it is this fact that makes radioactive decay so non-intuitive. We have the dismal feeling that our luck runs out. But the luck of a uranium nucleus never runs out. Just as a coin does not remember its flips, a nucleus has no history, and the probability of decay is constant in time. The random nature of radioactivity makes it possible to model the process on a computer and thereby make the kinetics of radioactive decay more intuitive.

How can the computer simulate a random process such as radioactive decay? It is quite simple. Consider the elementary toss of a coin experiment with its two outcomes, H and T. We can simulate this experiment by asking the computer for a random number between zero and one. If the number is less than half, we call the outcome H; otherwise we call it T. You would agree that, like the coin toss, this computer experiment has two equally likely outcomes, each with a probability of 50 percent. The computer has successfully simulated a random process by generating random numbers and testing them to find where in the interval [0, 1) they fall.

We will formulate this result more carefully. Consider an experiment, similar to the coin toss, with two mutually exclusive outcomes, **A** and **B**, with outcome **A** having probability p. In the case of the coin toss $p = 1/2$. The computer will simulate this experiment if for each random number x supplied by the computer we regard outcome **A** as happening when

$$0 \leq x < p,$$

and not happening when

$$p \leq x < 1.$$

Clearly outcome **B** has probability $1 - p$. Figure 11.1 illustrates these concepts. The line segment from zero to one represents the entire range of the random numbers supplied by the computer. Probabilities are always between zero and one; therefore p must be in the interval between zero and one. If the random number supplied by the computer is between zero and p, we will say outcome **A** occurred, but if the random number is between p and one we will say that the outcome was **B**.

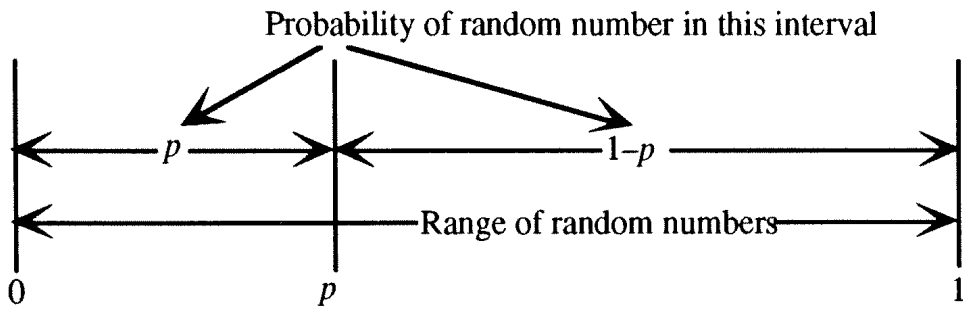


FIGURE 11.1 • Probability and Random Numbers. Computer-generated random numbers have a probability p of falling in the interval between zero and p. Computer-generated random numbers have a probability $1 - p$ of falling in the interval $1 - p$ to 1.

Suppose that we wish to simulate an experiment in which two coins are flipped. The probability of each of the four equally likely outcomes *TT*, *TH*, *HT*, *HH* is one fourth. The computer program simulates this experiment by supplying a random number x between zero and one. To each of the four equal intervals

$$0 \le x < \tfrac{1}{4},$$
$$\tfrac{1}{4} \le x < \tfrac{1}{2},$$
$$\tfrac{1}{2} \le x < \tfrac{3}{4},$$

and

$$\tfrac{3}{4} \le x < 1,$$

we assign one of the four outcomes *TT*, *TH*, *HT*, or *HH*. Clearly the probability of each of the outcomes is one fourth, so we have simulated the flip of two coins. The law of large numbers also works. If we supply a large number of random numbers, about one fourth of them will fall in each of the intervals mentioned. In what follows we will confine ourselves to an experiment with only two mutally exclusive outcomes; in a specific time interval a nucleus either decays (probability p) or it does not decay (probability $1 - p$).

A procedure to conduct this computer experiment is given in Algorithm 11.1. Most languages or spreadsheets will have an instruction that produces a random number. In BASIC the instruction LET x = RND produces a random number for x. A computer program that generates random numbers will produce the *same* sequence each time it is run. We say that the random numbers in this sequence have the same *seed*. If a new sequence is required, we need a new seed. An instruction similar to the BASIC instruction RANDOMIZE provides the computer with a new seed, which it gets by choosing a number from its internal date and time information, which is continually changing. The probability of getting the same seed twice is remote.

ALGORITHM 11.1 / Modelling the Toss of Two Coins

STEP 1. Generate a new sequence of random numbers with a new seed.
STEP 2. Loop until $N = 10000$ (while $N < 10000$).
  STEP 3. Get a random number x.
  STEP 4. If $0 \le x < 0.25$, then increment HH by one.
  STEP 5. If $0.25 \le x < 0.5$, then increment HT by one.
  STEP 6. If $0.5 \le x < 0.75$, then increment TH by one.
  STEP 7. If $0.75 \le x < 1$, then increment TT by one.
  STEP 8. Increment the loop counter, N.
  STEP 9. Print the value of the variables HH, HT, TH, and TT every 1000 random numbers.
STEP 10. End of loop.

PROBLEM 11.1 Tossing Two Coins • Write a program that samples the random numbers generated by the computer and sorts them into four intervals as listed above. Count and print the number in each category. It will be more efficient to print the results at intervals of 500 or 1000 trials rather than after each trial.

11.2 Modelling Radioactive Decay

We model radioactivity by starting with N_0 radioactive nuclei in a sample. A radioactive nucleus is the nucleus of an atom that is capable of emitting an alpha particle, a beta particle, or a gamma ray. Assume that the probability of a radioactive decay in a certain time is p. To each of the N_0 nuclei we assign a random number x in the interval [0, 1). Then, if

$$0 \le x < p,$$

we agree that the nucleus has made the transmutation from mother to daughter, we decrement the sample by one, and we increment the number of daughter nuclei by one. Otherwise the nucleus lives. This completes one time interval of arbitrary length.

This process is repeated for successive time intervals, each beginning with the number of nuclei that have not already decayed. The process ends when $N = 0$. A procedure to model a radioactive parent and a stable daughter is given in Algorithm 11.2.

ALGORITHM 11.2 / Model of a Radioactive Parent and a Stable Daughter

STEP 1. Let there be N_0 radioactive nuclei. Choose $N_0 = 1000$ to start.
STEP 2. Let there be D_0 daughter nuclei. Assign $D_0 = 0$ to start.
STEP 3. Let p the probability of decay in a unit time interval. To begin, choose $p = 0.05$.
STEP 4. Start a new random number sequence.

STEP 5. Let $t = 0$.
STEP 6. Loop until $N \leq 0$ (while $N > 0$).
  STEP 7. Plot or print points on the graph of $N(t)$ and/or $D(t)$.
  STEP 8. Set $M = N$, the number of nuclei still present. (M will also serve as a loop counter.)
  STEP 9. Loop M times to sample each remaining nucleus.
    STEP 10. Fetch a random number x and assign it to a nucleus.
    STEP 11. If $0 \leq x < p$, then decrement N and increment D:

$$N \leftarrow N - 1,$$

and

$$D \leftarrow D + 1.$$

  STEP 12. End of STEP 9 loop.
  STEP 13. Increment the time:

$$t \leftarrow t + 1.$$

STEP 14. End of STEP 6 loop.
STEP 15. Quit.

PROBLEM 11.2 Radioactive Decay with a Stable Daughter • Write a program to implement Algorithm 11.2. For parameters, choose $N_0 = 1000$, $p = 0.05$, with a domain and range of [0, 100] and [0, 1000], respectively. Experiment with different values of N_0, 10, 100, 1000, 10,000, ... You will have to change the domain and range. Describe the shape of the curve. Also experiment with different values of p, 0.005, 0.1, 0.5,... . This will give you a feeling for the effect of various parameters on the solution. A family of such curves produced the illustration at the opening of this chapter. The results for a single parent and a stable daughter are shown in Figure 11.2.

The effect of various choices of the probability on the rate at which the number of nuclei decreases should be clear from the curves you make. The slope of each curve is the derivative of N with respect to t. This derivative is called the *activity* of the sample. It is what you would measure if you had a G-M tube and a counter near the sample. Specifically, the activity A is defined by

$$A = -\frac{dN}{dt},$$

where the negative sign is needed to make A positive because the slope of the $N(t)$ curve, and hence the derivative, is negative. The activity is measured in curies, abbreviated Ci, a unit of activity defined to be 3.70×10^{10} disintegrations per second. Use the results of Problem 11.2 to answer these questions.

- How does the activity vary with time?
- When is it a maximum?
- When is it a minimum? Explain why.
- Is A directly proportional to N?

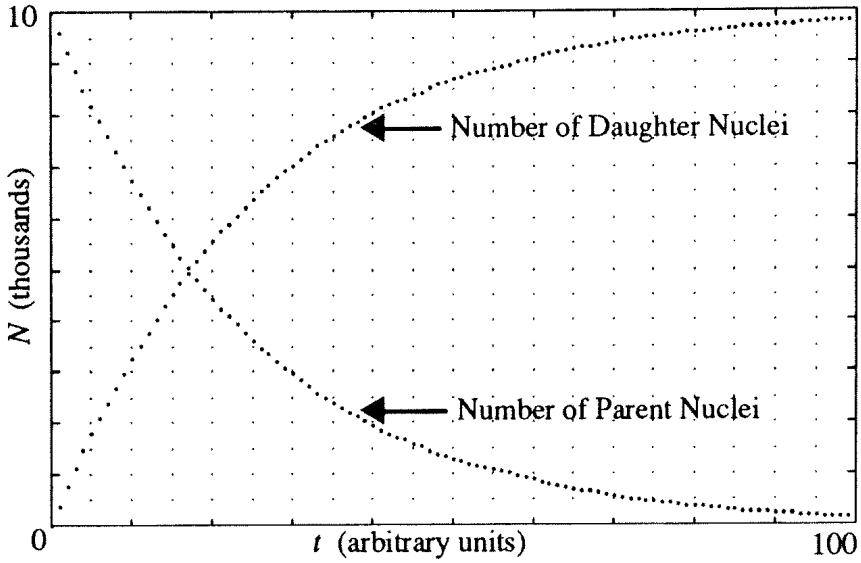


FIGURE 11.2 • Radioactive Decay of a Parent with a Stable Daughter. Parameters: N_0 = 10,000, D_0 = 0, p = 0.04, *dom* = [0, 100], and *ran* = [0, 10,000].

PROBLEM 11.3 Population of Daughter Nuclei • Graph the number of daughter nuclei, assuming for the time being that the daughter is a stable nucleus. The steps to do this are already outlined in Algorithm 11.2. It is not difficult to predict the shape of this curve since $N + D$ at any time is the number of nuclei with which the experiment was started.

11.3 A Mathematical Model of Radioactive Decay

The graphs you obtain from Problem 11.2 give the appearance of the exponential function

$$y = Ce^{-kt},$$

where C and k are constants. The exact expression we will attempt to fit is

$$N = N_0 e^{-\lambda t},$$

where λ is a constant related to the rate at which the nuclei decay. Because the logarithm function and the exponential function are inverses, it is a good idea to examine the logarithmic form of the equation when the latter function appears in your calculations. Use the natural log function that we shall designate by "ln." Thus rather than trying to fit the preceding equation to our simulated decay curves, we will first take the ln of both sides of the preceding equation. Doing this gives

$$\ln N = \ln\left[N_0 e^{-\lambda t}\right] = \ln N_0 + \ln e^{-\lambda t} = \ln N_0 - \lambda t.$$

Thus

$$\ln N = \ln N_0 - \lambda t.$$

But this equation has the form

$$y = mx + b,$$

with $y = \ln(N)$, $m = -\lambda$, $x = t$, and $b = \ln(N_0)$. Therefore a graph of $\ln N$ versus t should be a straight line with slope $-\lambda$ and intercept $\ln(N_0)$.

PROBLEM 11.4 Graphs of ln(N) versus t • Modify Algorithm 11.2 to graph ln(N) versus t rather than N versus t; then make a corresponding change in the solution to Problem 11.2 and rerun the program. In Algorithm 11.2, STEP 7 must be modified to read:

STEP 7. Plot or print a point on the graph of $\ln N(t)$.

But somehow you must avoid the case of finding ln(0) since it does not exist. The easiest modification is to exit the loop when $N \leq 1$. Run the program. Do you obtain a straight line? Try other values for the probability of decay. How does the probability affect the graph? Explain the erratic nature of the graph for small N. Is

$$N = N_0 e^{-\lambda t}$$

a suitable mathematical model for radioactive decay?

The time it takes for half of a sample of atoms to decay is called the *half-life*. Putting $N = 0.5N_0$ into the preceding expression gives

$$T_{\frac{1}{2}} = 0.693\lambda.$$

Finding the decay constant λ is equivalent to finding the half-life. The decay constant may be found from the slope of the ln(N) versus t curves as follows:

$$\lambda = \frac{\ln N_2 - \ln N_1}{t_2 - t_1}.$$

It is wise to choose points on the graph that are separated as far as possible.

11.4 Radioactive Series

In this section we will assume the daughter element is also radioactive. Algorithm 11.3 outlines the procedure to model this situation. We add a new variable D, the number of daughter nuclei with a probability p_d of decaying. During each time interval for which $D > 0$, we assign a random number x to each daughter nucleus. If $0 \leq x < p_d$, the daughter nucleus decays, otherwise it lives. Of course, we treat the parent nuclei the same say as before.

ALGORITHM 11.3 / Model of a Radioactive Parent and a Radioactive Daughter

STEP 1. Let there be N_0 radioactive nuclei. Try $N_0 = 5000$ to start.
STEP 2. Let there be D_0 daughter nuclei. Set $D_0 = 0$ to start.
STEP 3. Let p_p be the probability of decay of the parent nucleus in a unit time interval. Try $p_p = 0.05$.
STEP 4. Let p_d be the probability of decay of the daughter. Try $p_d = 0.01$
STEP 5. Choose a seed for the random number sequence.
STEP 6. Loop until $N = 0$ and $D = 0$ (while $N > 0$ and $D > 0$).
  STEP 7. Plot a point on the graphs of $N(t)$ and/or $D(t)$. Or, plot ln ($N(t)$ and/or ln ($D(t)$.
  STEP 8. Set $M = N$ and $C = D$.
  STEP 9. Loop C times if $C > 0$. Sample all of the daughter nuclei to see which decay.
    STEP 10. Fetch a random number x.
    STEP 11. If $x < p_d$ then decrement D:

$$D \leftarrow D - 1.$$

  STEP 12. End of STEP 9 loop.
  STEP 13. Loop M times if $M > 0$. Sample all of the parent nuclei to see which decay.
    STEP 14. Fetch a random number x.
    STEP 15. If $x < p_p$, then decrement N and increment D:

$$N \leftarrow N - 1,$$

and

$$D \leftarrow D + 1.$$

  STEP 16. End of STEP 13 loop.
  STEP 17. Increment the time:

$$t \leftarrow t + 1.$$

STEP 18. End of STEP 6 loop.
STEP 19. Quit.

In conclusion we raise a minor philosophical point. Although radioactivity involves a random process and it is impossible to tell when a particular nucleus will decay, there is order in the midst of this randomness. Given a sample of radioactive nuclei, it is easy to predict how many will be present after a certain time interval. The rate of disintegration as a function of time is also easy to predict. It is even possible to date the bones of an animal to find how long the animal has been dead by measuring the activity of carbon 14. The presence of randomness does not preclude the possibility of making predictions.

PROBLEM 11.5 Radioactive Parent and Daughter • Write a program to implement Algorithm 11.3. Use $N_0 = 5000$, $D_0 = 0$, $p_p = 0.05$, and $p_d = 0.025$. Let your domain and range

be [0, 200] and [0, 5000], respectively. Write a paragraph explaining the results. Also find the answers to these questions.

- How do the curves change when p_d is changed from $0.5p_p$ to $0.25p_p$?
- What if $p_d = p_p$? Try $p_p = p_d = 0.01$.
- What if $p_d = 2p_p$?
- What if $p_d = 4p_p$?

Write a summary of your conclusions.

PROBLEM 11.6 A Stable Granddaughter • We can also graph the growth of the granddaughter when the daughter is radioactive. Simply increment a variable G_d when you decrement D. Try these cases:

- Let $p = 0.02$ and $p_d = 0.01$.
- Let $p = 0.02$ and $p_d = 0.005$.
- Let $p = p_d = 0.02$, and describe your conclusions.

PROBLEM 11.7 Logarithmic Graphs for Radioactive Decay • In the previous problems graph log (N) versus t, log (D) versus t, and log (G_D) versus t. How do these graphs differ from those generated in Problem 11.6?

PROBLEM 11.8 The Probability of Decay and the Half-Life of Radioactive Nuclei (Challenging) • Try to find a mathematical relationship between p, the probability of decay in a certain time, and the half-life, $T_{1/2}$.

11.5 Programs

PROGRAM 11.5.1 / Modelling the Toss of Two Coins

```
! Algorithm 11.1 and Problem 11.1
LET M = 10000
RANDOMIZE
DO until n = M

   LET x = rnd
   IF x >= 0 and x < 0.25 then LET HH = HH + 1
   IF x >= 0.25 and x < 0.5 then LET HT = HT + 1
   IF x >= 0.5 and x < 0.75 then LET TH = TH + 1
   IF x >= 0.75 and x < 1 then LET TT = TT + 1
   LET n = n + 1
   IF mod (n, 500) = 0 then PRINT HH, HT, TH, TT
LOOP
END
```

```
! Repeat experiment M times.
! Plant a new seed.
! Quit loop after M passes
  through it.
! Get a random number.
! Both heads.
! Head first, tail second.
! Tail first, then head.
! Both tails.

! Print results every 500
! flips.
```

PROGRAM 11.5.2 / Model of a Radioactive Parent and a Stable Daughter

```
! Use domain [0, 50], range [0, N]
LIBRARY "ComPhys"                                          ! Library of graphics.
RANDOMIZE                                                  ! Start a new sequence.
INPUT PROMPT "Input the probability p. ":p                 ! Try p = 0.05 to start.
INPUT PROMPT "Input the number of nuclei. ":N              ! Start with 1000 nuclei.
CALL Open (xmin,xmax,ymin,ymax,wxmin,wxmax,wymin,wymax)    ! Try xmax = 100, ymax = 10000.
DO until N =< 0                                            ! Quit when all the nuclei
   LET M = N                                               ! are gone. M is loop counter.
   PLOT T,M                                                ! Plot N versus T.
   PLOT T, D                                               ! Plot D versus T.
   FOR I = 1 to M                                          ! Assign a random number
       LET x = rnd                                         ! to each nucleus
       IF x < p then                                       ! Has it decayed?
          LET n = n -1                                     ! Yes, Decrement the number of
                                                             parent nuclei.
          LET D = D + 1                                    ! and increment the number of
                                                             daughter nuclei.
       END IF                                              ! No, it hasn't decayed.
   NEXT i
   LET T = T + 1                                           ! Increment the clock.
   IF T =>xmax then EXIT DO                                ! Quit if curve goes off
LOOP                                                       ! the graph.
CALL LABEL(wxmin, wxmax, wymin, wymax)
END
```

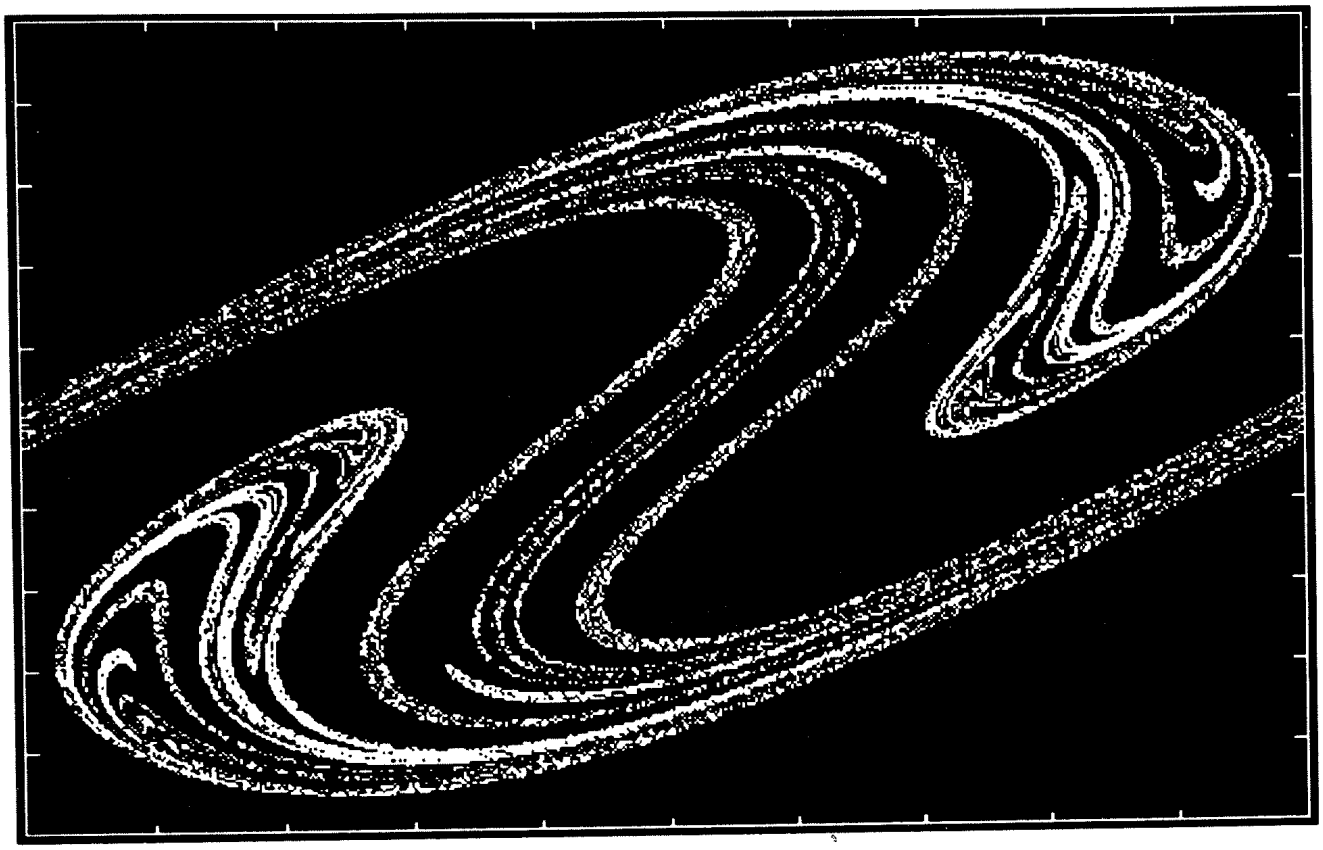
PROGRAM 11.5.3 / Graphing Log *N* versus Time

```
! Algorithm 11.2 modified. Problem 11.4
! Use domain [0, 100], range [0, log(N)] where N is the number of nuclei.
! If N = 5000, then log(N) is approx. 9.
LIBRARY "ComPhys"
RANDOMIZE                                                  ! Start a new random number
INPUT PROMPT "Input the probability p. ":p                 ! sequence. Try p = 0.05.
INPUT PROMPT "Input the number of nuclei. ":N              ! Try N = 5000 nuclei.
CALL Open (xmin, xmax, ymin, ymax, wxmin, wxmax, wymin, wymax)
RANDOMIZE                                                  ! New seed.
DO until N =< 1
   LET M = N
   PLOT T,log (M)                                          ! Plot ln(N) versus T.
   FOR I = 1 to M                                          ! Assign a random number
       LET x = rnd                                         !to each nucleus
       IF x < p then                                       ! Has it decayed?
          LET n = n -1                                     ! Yes, Decrement the number of
                                                             parent nuclei.
          LET D = D + 1                                    ! and increment the number of
                                                             daughter nuclei.
```

```
      END IF                                          ! No, it hasn't decayed.
   NEXT i
   LET T = T + 1                                      ! Increment the clock.
   IF T =>xmax then EXIT DO
LOOP
CALL LABEL(wxmin, wxmax, wymin, wymax)
END
```

PROGRAM 11.5.4 / Radioactive Parent and Daughter

```
! Algorithm 11.3, Problems 11.5 - 11.7.
! Domain [0, 100], range [0, N] where N is the number of nuclei.
LIBRARY "ComPhys"                                     ! Graphics subroutines.
CALL open (xmin, xmax, ymin, ymax, wxmin, wxmax, wymin, wymax)
LET N = 5000                                          ! Try N = 5000.
LET p = 0.05                                          ! Probability of parent
                                                        decaying.
LET pd = 0.03                                         ! Probability of daughter
                                                        decaying.
RANDOMIZE                                             ! Start new random number
                                                        sequence.
DO until N = 0 and D = 0                              ! Quit experiment both parent
                                                        and daughter nuclei are gone.
   PLOT t, N                                          ! Graph N versus T.
   PLOT t, D                                          ! Graph D versus T.
   LET M = N
   LET C = D
   FOR j = 1 to C                                     ! Let the daughter nuclei decay.
       IF rnd < pd then LET D = D -1                  ! The FOR..NEXT.. loop does not
                                                        execute if C ≤ 0.
   NEXT J
   FOR I = 1 to M                                     ! Let the parent nuclei decay.
       IF rnd < p then                                ! This loop does not execute
          LET N = N-1                                 ! if M ≤ 0.
          LET D = D + 1
       END IF
   NEXT i
   LET t = t + 1                                      ! Update the time.
   IF T >= xmax then EXIT DO                          ! Exit loop if the graphs
LOOP                                                  ! reach the screen limits.
END
```



Strange Attractor

12 / Chaotic Dynamics

Arguably the most broad based revolution in the world view of science in the twentieth century will be associated with chaotic dynamics.[1]

12.1 Introduction

If any introductory physics textbook has a section on chaos, I am not aware of it. Therefore it is not possible for you to review a textbook before getting started in this chapter. You may wish to review Chapter 2 of this book before beginning this chapter, in which we will carry forward our study of oscillating systems. Of special importance are those sections of Chapter 2 dealing with phase space. Almost every description of chaotic dynamical systems is rooted in phase space. You should be able to visualize the behavior of a simple system, like the object-spring system or a simple pendulum, in phase space. Once again, our work will involve

differential equations, the mathematical metaphor that is at the heart of so much physics. It would be helpful at this point if you were able to recognize Newton's Second Law as a differential equation. Like the reader of this book, the author approaches the subject of deterministic chaos from the point of view of a student rather than an expert.

In the past two decades the study of chaotic phenomena has brought together scientists from meteorology, medicine, mathematics, physics, chemistry, biology, economics, ecology, fluid mechanics and other disciplines. It is an understatement to say that the study of chaos is interdisciplinary. A marvelous view of the development of this field from many perspectives may be found in James Gleick's easy-to-read and highly recommended book, *Chaos*.[2] Furthermore, it is safe to say that the study of chaotic dynamics cannot flourish without the computer. Although mathematicians have been aware of chaotic phenomena for many decades, the modern digital computer has been responsible for the explosive growth of this field. Whether you are an engineer, physicist, biologist, chemist, or a student in some other discipline, in this chapter you will start down a road that has the potential of taking you many places.

What are the dynamical systems that are the substance of the study of chaos? A dynamical system may be a driven pendulum, the stock market, sand dunes along Lake Michigan, an insect population, a mathematical equation, a wave breaking into surf, the ozone layer, your brain, or your heart. It is best to start simply, with the oscillations of a pendulum for example, and that is where we shall begin. In order to keep your work fresh and exciting, the approach in this chapter will be to describe a specific dynamical system, namely a driven and damped pendulum, while, in parallel with this, you will carry through a study of a different system, described below. Your system is called a Duffing two-well oscillator, a system that has its origin in electronic circuits. A mechanical version of the Duffing oscillator has been described by Keith Briggs.[3]

There are one or two misconceptions with which we should deal. "Chaos" is a misleading word that implies many things to different people. We wish to point out that chaotic behavior and random behavior are not the same. Radioactive nuclei behave randomly, meaning they can be modelled mathematically with random numbers or described with probabilities, which we did in Chapter 11. There is no element of randomness in the orbit or trajectory of a chaotic dynamical system; unpredictability yes, randomness no. The throw of a die is random, the behavior of a chaotic system is not. The trajectory of a dynamical system can be calculated again and again and the results are always the same. Identical initial conditions always lead to identical results. Perhaps that is why some students of chaotic phenomena prefer to call it "deterministic chaos," underlining the fact that random behavior is excluded. The unpredictability is a result of the fact that tiny differences in initial conditions are amplified by the differential equations to give enormously different results, and so, *in practice*, it becomes impossible to make predictions.

12.2 Some Definitions

Reading about chaos can be confusing because you are unfamiliar with the terminology. To help with this problem, we offer the following definitions. You may study them now or refer to them as you work your way through this chapter.

Dynamical system: A system of one or more particles whose behavior (position, velocity, and acceleration) may be modelled by a system of differential equations together with a set of initial conditions. An example is the object-spring system we studied in Chapter 2. Frequently it is described with the expression

$$ma = -kx - cv,$$

cleverly hiding the fact that it is the following differential equation:

$$\frac{d^2x}{dt^2} = -\frac{k}{m}x - \frac{c}{m}\frac{dx}{dt}.$$

Here is one possible set of initial conditions: at $t = 0$, $x = 1$, and $v = 0$. Once formulated, the differential equations are themselves called dynamical systems and all references to the physical system may cease.

Non-linear dynamical system: If the differential equations involve non-linear terms like x^2, xv, $\sin(x)$, e^x, $\log(x)$, etc., then the system is said to be non-linear.

Deterministic system: A dynamical system (the physical system or the system of differential equations that describe it) that provides a *unique* (one and only one) solution for a given set of initial conditions. The systems studied in this chapter are all deterministic.

Phase space: A coordinate system with coordinates x, p_x, y, p_y, . . . , where p is the momentum. For a unit mass $p_x = v_x$, $p_y = v_y$, etc. For a one-dimensional system such as the object-spring system, phase space has only two dimensions, x and v_x. Refer once again to Chapter 2, in which phase space is introduced. Also see Figure 12.3.

Fixed point: Substitute a set of initial conditions into the system of differential equations. If the result is identical to the initial conditions, then the initial conditions constitute a fixed point. For example, if for the object-spring we use the initial conditions at $t = 0$, $x = 0$ and $v = 0$, then the solution is $x = 0$, $v = 0$. The point (0, 0) is a fixed point of this system. A mechanical system is in *equilibrium* at a fixed point.

Orbit: The set of points in phase space that satisfy the differential equations that describe the dynamical system. For a fixed point, the orbit is a single point. In the undamped case of an object-spring system the orbit is an ellipse (Chapter 2). Figure 12.3 shows several orbits of a simple pendulum.

Periodic orbit: If the set of points in the solution of the differential equations repeats itself, then the orbit is said to be periodic. The orbit of an undamped object-spring system, if not the fixed point (0, 0), is an ellipse in phase space, hence it is a periodic orbit. In space-time, the solution is the sine function which is a periodic function (refer to Chapter 2). The orbits in Figure 12.3 are periodic.

Limit cycle: A closed curve in phase space toward which an orbit may evolve as $t \to \infty$. The damped object-spring system driven by a low-amplitude sinusoidal force has a limit cycle. A limit cycle for a damped and driven pendulum is shown in Figure 12.6.

Attractor: The set of points in phase space toward which a dynamical system evolves as t gets large. This might be a fixed point, as it is for the damped object-spring problem (refer to Chapter 2), a limit cycle, as it is for the damped and driven simple pendulum, or an extremely complex set of points, as in the case of some non-linear dynamical systems. In Figure 12.4 there are three attractors, each a point. In Figure 12.6 the attractor is a limit cycle.

Chaotic Attractor: Loosely speaking, if the orbit is very *sensitive to the choice of initial conditions*, in the sense that nearby orbits in phase space diverge exponentially, then the attractor is said to be chaotic. Only non-linear dynamical systems have chaotic attractors. Such attractors are sometimes called *strange* and are examples of *fractals*. Dynamical systems of this type are also *unpredictable*. Small differences in the initial conditions lead to widely divergent orbits *making predictability a practical impossibility.*

Be aware that the study of chaotic dynamical systems is in an early stage, and some of the definitions are not as simple or crisp as those in a fully developed discipline. For example, what does "sensitivity to the choice of initial conditions" mean? It means that trajectory of the system in phase space given one set of initial conditions diverges exponentially from a trajectory with *almost the same* set of initial conditions. So what does "almost the same" mean? You must feel like you are being given the run-around, but there is a mathematical way of phrasing the definition and computational techniques to test dynamical systems; unfortunately both are beyond the scope of this book.

12.3 The Simple Pendulum

The system we will study is shown in Figure 12.1. A mass m is attached to a rigid massless rod so the center of gravity of the mass is a distance L from the frictionless pivot. The system is constrained to move in a plane. If there is no viscous damping, then the motion of this system is governed by the equation

$$\tau = -Ia$$

where τ is the torque around the pivot point, I is the moment of inertia about the pivot point, and a is the angular acceleration,

$$a = \frac{d^2x}{dt^2}.$$

(For convenience in studying a number of dynamical systems and economy and clarity of programming, we will use x as a "displacement", v as its time rate of change, and a as the second derivative of x. In the case of the simple pendulum the displacement is an angle measured in radians, v is the angular velocity in (rad)s^{-1}, and the acceleration a is the angular acceleration in (rad)s^{-2}. We will also standardize the use of ω for the frequency of the sinusoidal driving forces. Thus, for example, ω is not the angular velocity of the simple pendulum, but v is. Finally, we will drop most references to units. All of these simplifications are relatively common in the literature.)

For the simple pendulum $\tau = mg\sin(x)$ and $I = mL^2$. Making these substitutions gives

$$-mgL\sin x = mL^2\frac{d^2x}{dt^2}.$$

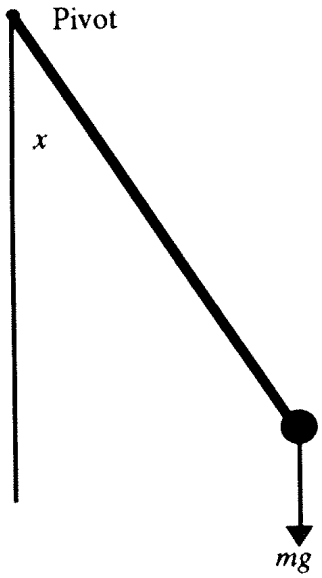


FIGURE 12.1 • The Simple Pendulum.

This equation is easily simplified by dividing by mL. In addition we choose our pendulum to be a long one with $L = 9.8$ m, then the equation becomes much simpler, namely,

$$\frac{d^2x}{dt^2} = -\sin x.$$

Recall that it was Newton's Second Law that led to this differential equation. We will refer to the equation that Newton's Second Law gives as the *equation of motion*. Because this differential equation has "$\sin(x)$" rather than "x" on the right-hand side, and since the sine function is non-linear (think about its graph), this is a (second-order) *non-linear* differential equation, and it is difficult to solve. In the special case where x is small and we can equate x with $\sin(x)$, the non-linearity is removed and there are analytic solutions.

We, on the other hand, wish to keep the non-linearity because only non-linear systems exhibit chaotic behavior. Furthermore, our numerical solutions do not require that the equation be linear. With the simple pendulum we introduce a *non-linear dynamical system*. It will be the focus of the remainder of this chapter, albeit with complications. The non-linear dynamical system that you will study is the Duffing oscillator. Its equation of motion is

$$\frac{d^2x}{dt^2} = x - x^2,$$

so it too is a non-linear system. In both the simple pendulum and the Duffing oscillator equation we have, for the moment, omitted the viscous damping term, $-cv$, and the driving term, $F\cos\omega t$.

12.4 Runge-Kutta Solutions to the Equation of Motion

Algorithm 12.1 outlines a method of finding a numerical solution. This method of solving differential equations is known as the Runge-Kutta method, and a detailed description of this technique is beyond the scope of this book. The method is described in most numerical analysis textbooks. We use the method in this chapter because it is more accurate and faster than the Feynman-Newton algorithm, and our work will demand both the speed and the accuracy. Be mindful that the Runge-Kutta method is accomplishing the same end as the Feynman-Newton method. So, with considerable faith on your part, the formulas to integrate the equation of motion follow.

Begin by writing Newton's Second Law (the equation of motion) in the following (unconventional) form:

$$\mathbf{a} = \frac{\mathbf{F}(t,x,v)}{m},$$

where we have indicated the possibility that the net force **F** varies with time, position, and velocity. We confine ourselves to one-dimensional problems so we may drop the vector notation, we assume that m is a unit mass (1 kg), and we recall that

$$a = \frac{d^2x}{dt^2}.$$

This allows us to write the equation of motion very neatly, as follows:

$$\frac{d^2x}{dt^2} = F(t,x,v),$$

where $F(t, x, v)$ comes from the application of Newton's Second Law to the system. Specifically, F is the net force *per unit mass*, and F is the function that appears in the Runge-Kutta formulas that follow.

The Runge-Kutta formulas needed to solve the equation of motion, a second-order differential equation, follow. The quantity x will be updated with the expression

$$x_{n+1} = x_n + h\left[v_n + \frac{1}{6}(k_1 + k_2 + k_3)\right],$$

and v will be updated with the expression

$$v_{n+1} = v_n + \frac{1}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right),$$

where

$$k_1 = hF(t_n, x_n, v_n),$$

$$k_2 = hF\left(t_n + \frac{h}{2}, x_n + \frac{h}{2}v_n, v_n + \frac{k_1}{2}\right),$$

$$k_3 = hF\left(t_n + \frac{h}{2}, x_n + \frac{h}{2}v_n + \frac{h}{4}k_1, v_n + \frac{k_2}{2}\right),$$

and

$$k_4 = hF\left(t_n + h, x_n + hv_n + \frac{h}{2}k_2, v_n + k_3\right).$$

ALGORITHM 12.1 / Numerical Solution Using the Runge-Kutta Method

STEP 1. Use Newton's Second Law to find the equation of motion, $a = F(t, x, v)$.
STEP 2. Choose the initial conditions, x_0 and v_0.
STEP 3. Decide on the step size h.
STEP 4. Begin a loop. Loop until key is pressed.
STEP 5. Plot (t, x), or make a phase portrait by plotting (x, v).
STEP 6. Calculate k_1, k_2, k_3, and k_4 from the Runge-Kutta formulas, and use them to update x:

$$x_{n+1} = x_n + h\left[v_n + \frac{1}{6}(k_1 + k_2 + k_3)\right],$$

STEP 7.Update v with the Runge-Kutta formulas:

$$v \leftarrow v + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4).$$

STEP 8. Increment the time:

$$t \leftarrow t + h.$$

STEP 9. End of STEP 4 loop.
STEP 10. Quit.

In the case of the simple pendulum described in the previous section, the angular acceleration a will be determined by the equation

$$a = -\sin x.$$

But there are some complicating factors that we wish to introduce as we proceed through the chapter. The first complicating factor is viscous damping. Pendulums do not swing forever, they lose energy as a result of friction at the pivot or in the medium in which they move. We will assume that the latter is the dominating term, in which case the losses are proportional to v, and the angular acceleration becomes

$$a = -\sin x - cv,$$

where c is called the damping constant.

At another point in this chapter we will also want to drive the pendulum to add energy. For a child on a swing, the driving force is pulsed by pushing the child at periodic intervals. In our case we will bounce the pivot up and down sinusoidally at a periodic rate or we will make gravity vary sinusoidally. In practice either one is difficult to do, but the computer gives us the freedom to model these experiments. In either case the acceleration with the complicating factors of viscous damping and a driving force becomes

$$a = -cv - (1 + F\cos\omega t)\sin x,$$

where

$$\omega = 2\pi f,$$

f is the frequency of the driving force, and F is its amplitude. Our simple pendulum has become a *damped and driven pendulum.* We will experiment with the parameters c and F.

PROBLEM 12.1 The Simple Pendulum • Write a program to implement Algorithm 12.1. Choose $x_0 = 1$, $v_0 = 0$, $c = F = 0$, $h = 0.05$, $dom = [0, 12]$, and $ran = [-1.2, 1.2]$. Graph $x(t)$. Your output should resemble a cosine function.

PROBLEM 12.2 The Duffing Oscillator • Use Algorithm 12.1 to study the motion of a unit mass for which the equation of motion is

$$a = -cv + bx - dx^3 + F\cos\omega t.$$

Start simply with $c = F = 0$, $b = d = 1$, and $h = 0.05$, $x_0 = 1$, $v_0 = -0.4$, $dom = [0, 25]$, and $ran = [-2, 2]$. Make a graph of $x(t)$. Also try these initial conditions: $x_0 = 1$ and $v_0 = -0.72$.

12.5 Potential Energy of a Dynamical System

Besides the equation of motion, there is another perspective from which we can examine the motion of a dynamical system, namely its energy. The potential energy, V, of a dynamical system may be defined as the negative of the integral of the forces (other than the damping

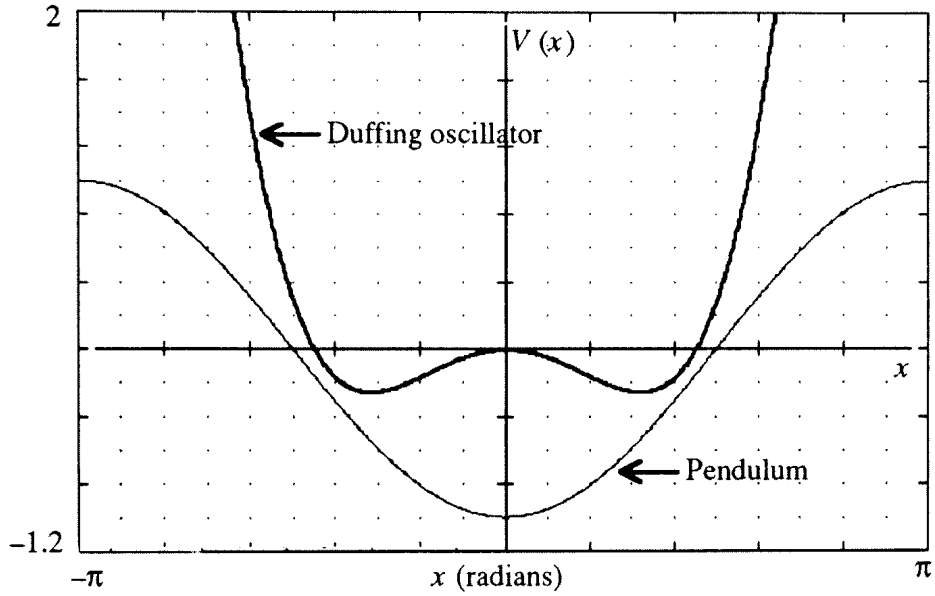


FIGURE 12.2 • Potential Curves for the Simple Pendulum and the Duffing Oscillator. $V(x)$ is Plotted vs. x. The thin line is the pendulum potential, the thick line is the Duffing Oscillator potential. *Dom* = $[-\pi, \pi]$ and *ran* = $[-1.2, 2]$.

and driving forces). For the simple pendulum, this definition gives

$$V = -\int F(x)\,dx = -\int -\sin x\,dx = -\cos x + C,$$

where we have taken $L/g = 1$. C is a constant of integration that depends only on where the zero for potential energy is chosen. We choose to let $V = 0$ when $x = \pi/2$, making C zero. Thus

$$V(x) = -\cos x$$

is the potential energy of our simple pendulum.

Integrating the force equation for the Duffing oscillator gives

$$V(x) = -\frac{1}{2}x^2 + \frac{1}{4}x^4,$$

where we have chosen to let $V = 0$ at $x = 0$. Graphs of both of these functions are shown in Figure 12.2. The *potential energy* of a particle is simply the potential times the mass of the particle. If we think of a particle of unit mass moving in either one of these potentials, then the graphs also describe the potential energy of the particle.

To understand what these curves mean, think of the pendulum and the Duffing oscillator as one-dimensional marbles of unit mass rolling up and down one-dimensional hills having a shape identical to their potential or potential-energy curves. Thus the pendulum is equivalent to a marble rolling on a "washboard" terrain, while the Duffing oscillator is equivalent to a marble between two mountains with a small hill in the middle. The mountains are *quartic potentials*, while the hill is a *quadratic potential*.

If the kinetic energy of the marble is less than the potential energy at the top of one of the hills, the marble will be trapped between two hills, and it will roll back and forth just as a pendulum swings back and forth. If its kinetic energy exceeds the potential energy of the hills, the marble keeps on rolling forever, up and down the hills of its sinusoidal potential energy function. The Duffing oscillator, on the other hand, is like a marble forever trapped between the two mountains, or, if its energy is sufficiently small, it will be trapped between a mountain and the central hill. (Both of the potential functions may be called *potential wells*, although the Duffing potential is more like a well than the pendulum potential. In fact, the Duffing potential is a *two-well* potential, while the simple pendulum is a *periodic* potential.)

PROBLEM 12.3 Energy of a Duffing Oscillator • Refer to Problem 12.2. Explain the x versus t curves for the two different sets of initial conditions of the Duffing oscillator in terms of the potential diagram in Figure 12.2.

12.6 Portraits in Phase Space

12.6.1 Undamped Motion

Most of our work will not involve graphs of x versus t. Instead we will be looking at v versus x graphs, i.e. trajectories called *phase curves* of the system in phase space. Phase space has position graphed along the horizontal axis and velocity graphed along the vertical axis (refer to Chapter 2). A set of phase curves is called a *phase portrait*. We begin by constructing the phase curves of the simple pendulum with neither damping nor driving forces. It will be useful to make a phase portrait using a variety of initial conditions. To do this we will start the pendulum at $x = 0$, but we will give it various velocities corresponding to various kinetic energies.

PROBLEM 12.4 The Simple Pendulum in Phase Space • Modify the program in Problem 12.1 to graph v versus x. Use *dom* = $[-3\pi, 3\pi]$ and *ran* = $[-3, 3]$. Use the initial conditions $x_0 = 1$ and $v_0 = 0$. When the program is working, modify it to graph a family of orbits in phase space corresponding to different energies. Our results are shown in Figure 12.3. To give some clues about this diagram we have shown direction of flow at several points. Refer to Figure 12.3 and answer these questions.

- If the initial conditions are $x_0 = 2\pi$ and $v_0 = 2$, where on a phase curve does the pendulum begin?
- Where on a phase curve does the pendulum go through the bottom of its swing?
- At which points and on what phase curves does the pendulum just go "over the top?"
- Why do the phase curves on the top half of the diagram flow to the right, while those on the bottom half flow to the left? Are there any other options for a phase portrait? The motion along a phase curve is called the *phase flow*.
- What is the difference between the closed phase curves and the open ones?
- What is happening to the pendulum at $x = \pi$ and $v = 0$? Do you think you could get the pendulum to stay there?

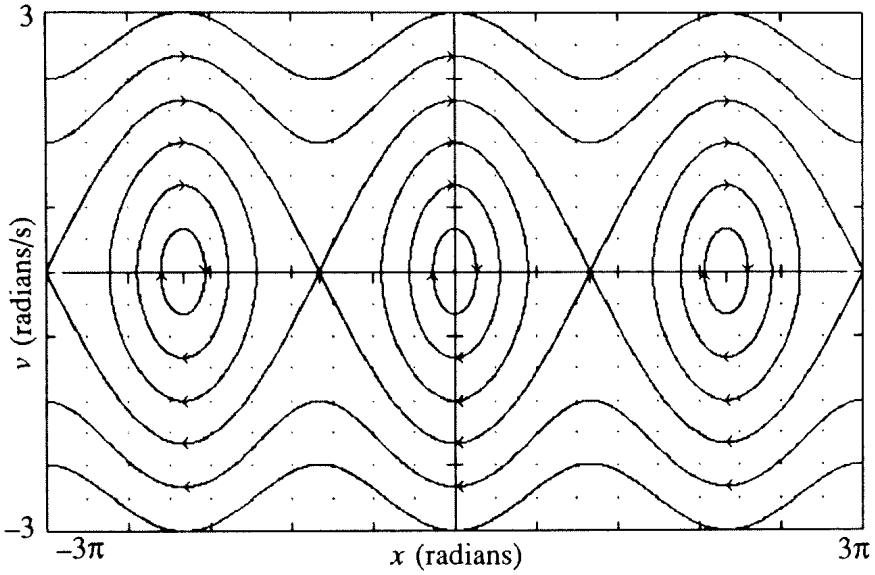


FIGURE 12.3 • Phase Portrait of a Simple Pendulum with No Damping or Driving Forces. The velocity ranges from −3 to 3 in steps of 0.5. *Dom* = [−3π, 3π], and *ran* = [−3,3] s^{-1}

- How could you modify the initial conditions to get a set of concentric ellipses centered on (2π, 0)? Try it.
- Are the orbits periodic in phase space?

In the vocabulary of a physicist, the point in phase space (0, 0) and the set of points (± 2nπ, 0), where n is an integer (. . . −2, −1, 0, + 1, +2, . . .), are points of stable equilibrium for the simple pendulum; the pendulum is hanging down from the pivot. The set of points [±(2n − 1)π, 0] are points of unstable equilibrium; the pendulum is "standing" on its pivot. Mathematicians and students of chaotic dynamics would call both sets of equilibrium points *fixed points*. A simple pendulum at a fixed point stays there forever; hence the name.

PROBLEM 12.5 The Duffing Oscillator Phase Portrait • Construct a phase portrait for the Duffing oscillator system, the study of which began in Problem 12.2. At $t = 0$ let $x = 1$ and step the initial velocity from 0.1 to 1.6 in steps of 0.3. For the moment, keep $c = 0$ and $F = 0$, let *dom* = [−2, 2], and *ran* = [−2, 2]. Explain your phase portrait in terms of the potential function graphed in Figure 12.2. Why does it not extend over large values of x like the phase portrait of the simple pendulum? You should be able to correlate points on the phase portrait with points on the potential diagram; try it.

12.6.2 Damped Motion

In this section we will study the motion of the pendulum and the Duffing oscillator when damped. You will begin to learn to use graphs of phase space to understand and visualize the behavior of a system. This capacity does not come instantly, but this skill is absolutely necessary if we are to understand chaotic dynamics.

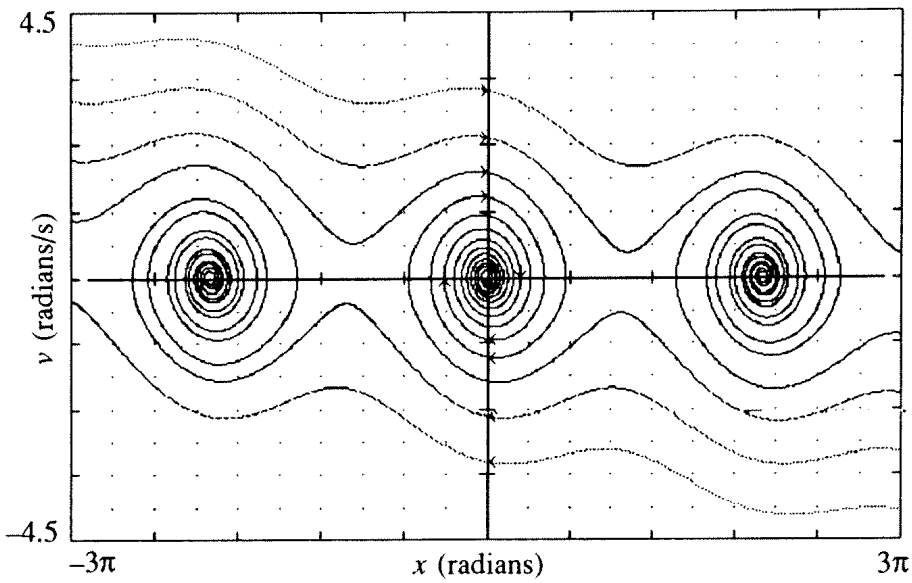


FIGURE 12.4 • Phase Portrait of a Damped Pendulum. Damping constant c is 0.1. The velocity varies from −4 to 4 in steps of one. *Dom* = [−3π, 3π] and *ran* = [−4.5, 4.5] s^{-1}

Viscous damping causes the pendulum to slow down. The damped pendulum is an example of a *dissipative* system because energy is being lost to the surroundings. A "$-cv$" term is added to the right-hand side of the equation of motion to take into account the air resistance of the bob. The equation for the angular acceleration becomes

$$a = -\sin x - cv,$$

where once again we assume that $m = 1$ and $g/L = 1$.

Our new phase portrait is shown in Figure 12.4. Can you describe what is happening to the pendulum for each of the phase curves in Figure 12.4? Take, for example, the second-from-the-bottom curve on the right. At $t = 0$ the pendulum is given a clockwise swing from an initial angle of 3π (almost straight up provided the pendulum rod is rigid). It goes around once, then again, then a little more than half a revolution, whereupon it swings back and forth until it comes to rest at $x = -2\pi$.

Once again, there is a fixed point at (0, 0) and, in fact, at $(n\pi, 0)$ where n is an integer. The points $(2n\pi, 0)$ are also called *attractors*. A set of initial conditions will always evolve toward one of these points in phase space. In the case of an object-spring undergoing damped oscillatory motion, the attractor is a single point in phase space (0, 0); springs cannot go round and round like pendulums. The attractor for an undamped oscillator with non-zero energy is an ellipse; refer to Chapter 2.

PROBLEM 12.6 The Duffing Oscillator with Damping • Investigate the Duffing oscillator with damping, that is, add a damping term to the angular acceleration and then study the behavior of the system in phase space. With damping

$$a = -cv + x - x^3$$

is the equation of motion of the Duffing oscillator. Experiment with various values of c from 0.1 to 0.4. Begin with initial condition $x_0 = 1$ and make a family of curves $v_0 = 0.1$ to $v_0 = 1.9$. Then try $x_0 = -1$ with the same set of initial velocities. Choose $dom = [-2, 2]$ and $ran = [-2, 2]$.

12.6.3 Driven and Damped Oscillators: The Road to Chaos

In this section we will add driving forces to our dynamical systems. For the simple pendulum, the equation of motion becomes

$$a = -cv - (1 + F\cos\omega t)\sin x,$$

while in the case of the Duffing oscillator the equation of motion becomes

$$a = -cv + x - x^3 + F\cos\omega t,$$

where ω is the angular frequency ($\omega = 2\pi f, f$ in Hz) of the driving force with amplitude F. Adding a driving force increases the complexity of the phase portraits, as you will soon see, and it becomes desirable to be able to change the driving force while the equation of motion is being solved. Algorithm 12.2 describes how Algorithm 12.1 may be modified to vary the driving force. To avoid large abrupt changes in the driving force, we only allow the force to be changed every nth step, where n is 25 or more.

ALGORITHM 12.2 / Making Phase Portraits of Damped and Driven Dynamical Systems

STEP 1. Express the equation of motion in the form $a = F(t, x, v)$.
STEP 2. Assign initial conditions x_0 and v_0. Choose a step size h.
STEP 3. Begin a loop.
  STEP 4. Loop n times where n is 25 or more.
    STEP 5. Plot the point (x, v).
    STEP 6. Calculate k_1, k_2, k_3, and k_4 from the Runge-Kutta formulas (Algorithm 12.1).
    STEP 7. Update x:

$$x \leftarrow x + h\left[v + \frac{1}{6}(k_1 + k_2 + k_3)\right].$$

    STEP 8. Update v:

$$v \leftarrow v + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4).$$

    STEP 9. Increment t:

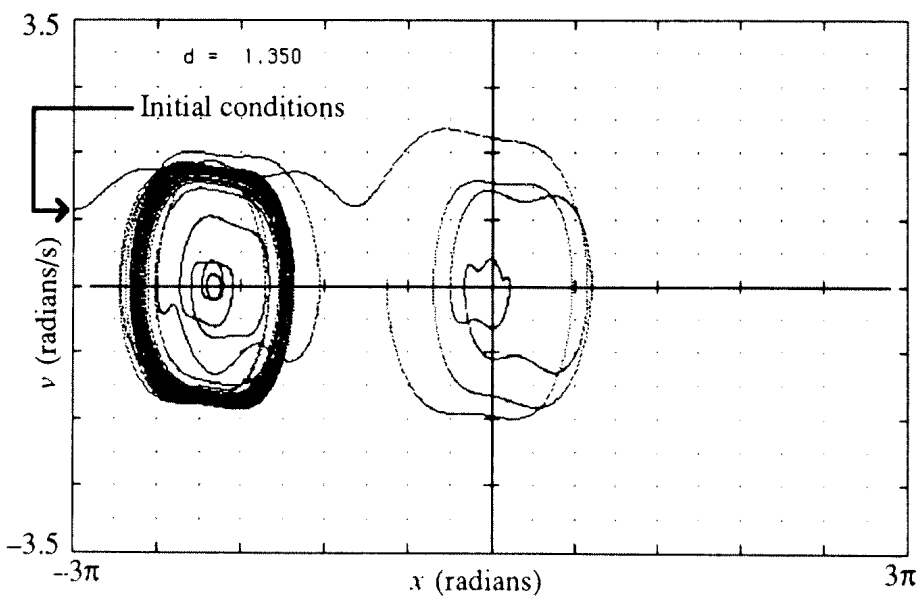
$$t \leftarrow t + h.$$

STEP 10. End of STEP 4 loop.
STEP 11. Get keyboard input.
STEP 12. If key is "*u*" key, increment the driving force by 0.01, print F.
STEP 13. If key is "*d*" key, decrement the driving force by 0.01, print F.
STEP 14. If key is "*c*" key, put a fresh coordinate system on the screen.
STEP 15. If key is "*q*" key, exit the loop that started with STEP 4.
STEP 16. End of STEP 3 loop.
STEP 17. Quit.

Our phase portrait for the damped and driven simple pendulum is shown in Figure 12.5. We began with $x_0 = -3\pi$, $v_0 = 1$, $c = 0.1$, $\omega = 2$, and $F = 0.9$. After some transient behavior, the system stabilizes, the trajectory encircles the point at $(-2\pi, 0)$, and the orbit becomes periodic. If F is held constant, the trajectory evolves toward an orbit in phase space called a *limit cycle*. In Figure 12.5 you can see the high density of trajectories about the point $(-2\pi, 0)$. This is a family of periodic orbits obtained by increasing the driving force from 0.9 to about 1.35. Figure 12.6 shows the limit cycle for $F = 1$, an orbit produced by clearing the screen after the system had evolved for some time from the initial conditions. The period of these orbits is twice that of the driving force.

The driving force is further increased. At approximately $F = 1.35$, the trajectory begins to wander away from the point $(-2\pi, 0)$ in a haphazard manner that appears to be chaotic, but is not. The pendulum eventually begins to turn either clockwise or counterclockwise and the

FIGURE 12.5 • Phase Portrait, Part A, of the Road to Chaos for a Driven and Damped Pendulum. As the driving force is increased, the system evolves from the initial conditions through a family of periodic orbits. When F reaches 1.35, the trajectory meanders but the system is not yet chaotic. Parameters: $c = 0.1$, $F = 0.9$ to 1.35, $\omega = 2$, $dom = [-3\pi, 3\pi]$, and $ran = [-3.5, 3.5]\ s^{-1}$.



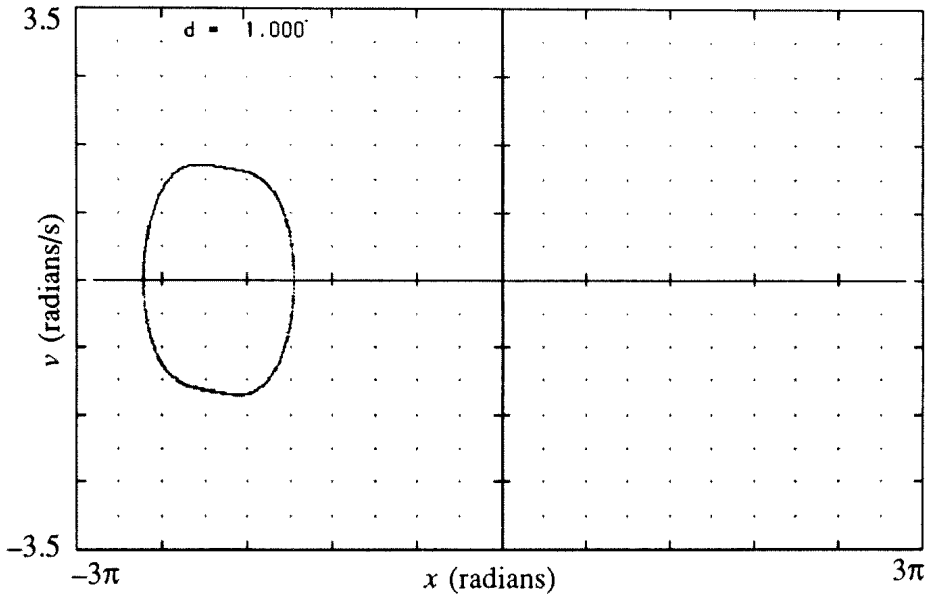
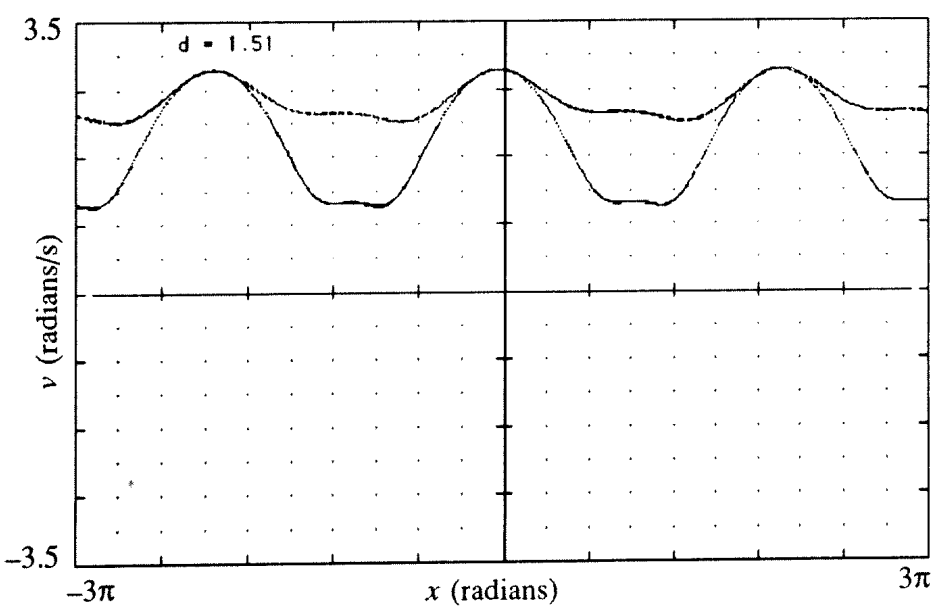


FIGURE 12.6 • The Road to Chaos, Part B: A Limit Cycle. If F is fixed and is less than 1.35, the trajectory of the pendulum system in phase space is periodic and it evolves toward a limit cycle. We show the limit cycle for $F = 1$. The period of the motion is twice that of the driving force.

trajectory in phase space becomes stable and periodic once again. If there is no increase in the driving force, the pendulum's behavior settles down and it continues to revolve in the same direction. Its period in phase space is still twice that of the driving force. Refer to Figure 12.7, which was obtained by clearing the screen to get rid of the history of the system, without

FIGURE 12.7 • The Road to Chaos, Part C: After extending Figure 12.5 to $F = 1.51$, we allow the transient behavior to die, the screen is cleared, and we observe a trajectory with *period two.*



modifying the behavior of the pendulum. The orbit is periodic but there are two distinct curves, meaning the period of the trajectory in phase space is twice that of the driving force and identical to the limit cycle in Figure 12.6.

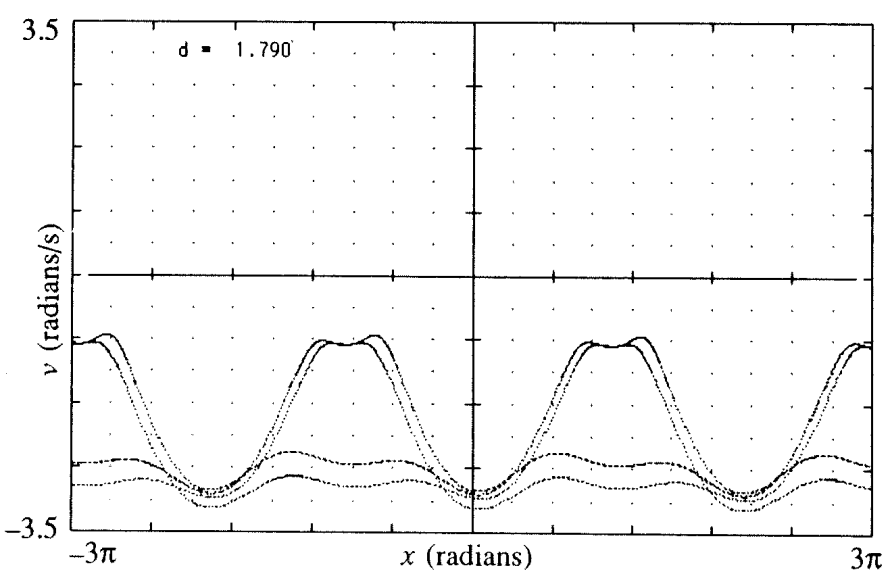
Between Figure 12.7 and Figure 12.8, the period of the pendulum has doubled. In contrast to Figure 12.7, in Figure 12.8 the pendulum moves with a *period four* times that of the driving force, a phenomenon called period doubling. This behavior is a result of increasing the driving force from 1.51 to 1.79.

The driving force is increased again until at approximately $F = 1.8$ the motion of the pendulum becomes chaotic, its trajectory in phase space wanders about more or less aimlessly, a feature illustrated in Figure 12.9. Sometimes the pendulum rotates clockwise about its pivot point; other times it rotates counterclockwise. It appears to be pulled toward one of the stable points, then wanders away again. The motion is truly chaotic.

PROBLEM 12.7 Chaotic Behavior of the Duffing Oscillator • Implement Algorithm 12.2 for the Duffing oscillator. Use initial conditions $x_0 = 1$, $v_0 = -0.1$, and choose the parameters as follows: $c = 0.1$, $\omega = 1.4$, $F = 0.1$, and $h = 2\pi/(\omega n)$, where $n = 150$ (you may wish to experiment with n, smaller values are permitted). Let $dom = [-1.6, 1.6]$ and $ran = [-1.2, 1.2]$. Start the program.

- Let the program run until the trajectory stabilizes and the motion becomes periodic. Clear the screen. Do you observe a closed trajectory around the point (1, 0)? Remember this is a trajectory in phase space, not in space-time. What is the particle doing in space-time? In phase space you are seeing the trajectory approach a limit cycle.
- Press the "*u*" key to increase the driving force. Allow the transients to die and the trajectory to stabilize while incrementing the driving force. Clear the screen now and then and be patient.

FIGURE 12.8 • The Road to Chaos, Part D: Period Doubling. With another run, to show that the pendulum can rotate in both directions, we observed this behavior with $F = 1.79$.



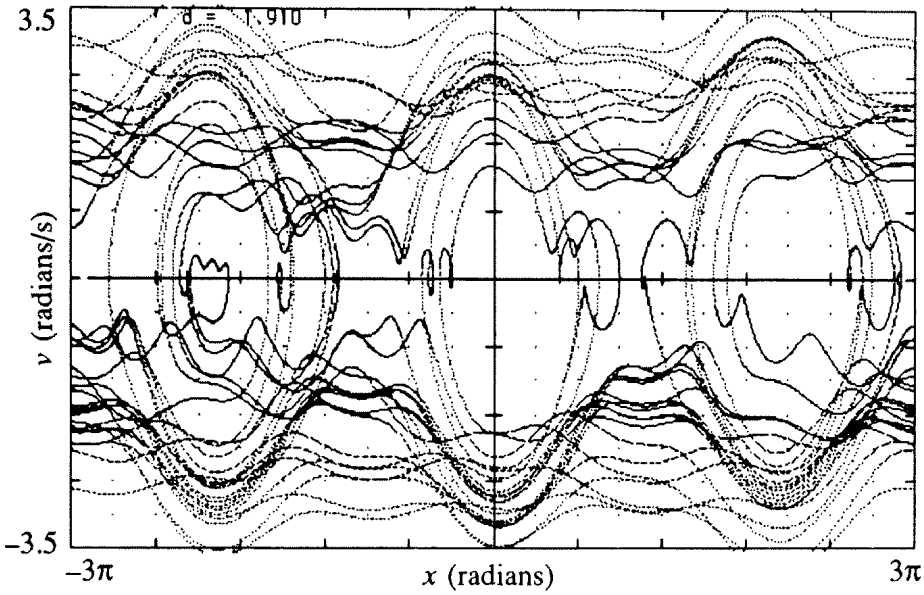


FIGURE 12.9 • The Road to Chaos, Part E: Chaos. This is an extension of Figure 12.5 to $F = 1.9$ and chaos.

- Slowly increment the driving force until $F = 0.30$. Allow the transients to settle and obtain a clean periodic trajectory.
- Increment the driving force to $F = 0.31$, clear the screen, and allow the story to unfold. At $F = 0.31$ or 0.32, you will observe a phenomena called *period doubling*. The trajectory in phase space is no longer a single closed loop around the point (1, 0); it is two loops. The time it takes for the trajectory in phase space to repeat itself is twice what it was at $F = 0.30$, for example. Once again we observe period doubling. The marble in the potential well rolls back and forth, but it no longer repeats itself every time it goes back and forth; it repeats itself every other time. One time it goes farther up the middle hill than the other, while it appears to go the same height up the mountain on the right.
- Increment the driving force until F is 0.34 or 0.35. Describe the behavior of the system. Is the motion periodic? Chaotic?
- What happens when you increase the driving force to 0.36? Always be patient. At what value of F does the marble cross over the central hill to the other potential well? The system becomes increasingly chaotic at about $F = 0.36$, although there may be a few larger values of F for which the orbit is periodic but with period three or four.
- Increment F to about 0.4 and observe the chaotic behavior of this system.

You have now observed chaotic behavior. Experiment, if you wish, with other values of c, ω, and F; try $\omega = 1$, $c = 0.5$ and values of F from 0.30 to 0.36, for example.

The system certainly appears to be chaotic, but how do you know the system is chaotic? That is a difficult question from a computational perspective. One way to prove that the system is chaotic is to show that slightly different initial conditions produce trajectories in phase space that diverge exponentially. A second way to prove that the system is chaotic is

to show that its attractor is a fractal. We will take a non-rigorous but intuitive approach to the second method. We begin by introducing a new concept, the Poincaré section.

12.7 Poincaré Sections

There is a way to study chaotic motion that is better than watching a trajectory wander around in phase space. Consider the phase portraits we have been studying but imagine a third axis, in our case a time axis, pointing out of the screen, and imagine the time evolution of the system to take place along this axis. To get a firmer grip on this concept, suppose that we are studying a simple undamped and undriven pendulum swinging back and forth in a small arc near the bottom of its swing. In phase space this motion appears to generate an ellipse around the stable point (0, 0); refer again to the phase portrait in Figure 12.3. The phase portrait is a set of points $(x_0, v_0, t_0), (x_1, v_1, t_1), (x_2, v_2, t_2), \ldots$ generated by solving the differential equation that describes the system. Note that all we graphed were the x-v coordinates, the screen of the video monitor being the x-v plane. If we add a time axis perpendicular to the screen, we will generate a curve in x-v-t space. Of course, it is impossible for the video monitor to display this space curve. For the moment we must use our imaginations to see this curve.

Imagine, in fact, that the ellipse in the x-v plane is a circle, and that time is positive out of the x-v plane. Then, as time progresses, the dot representing the point in phase space that describes the state of the pendulum traces out a helix in x-v-t space. A point now has three coordinates—x, v, and t— and a point is specified by the ordered triple (x, v, t) rather than the ordered pair (x, v). If the orbit in (x, v) space is elliptical, then the trajectory in (x, v, t) space will be a squashed helix.

We must complicate the picture even more. We imagine the time axis poking straight out of the video monitor at us. We take that time axis and bend it in a circle so that it circles around behind the screen and pierces the screen in exactly the same place. Now time, like a clock, runs in circles. Refer to Figure 12.10.

Assume that we choose the circumference of the time circle to be exactly one period of our dynamical system (assuming it to be periodic). In the case of the simple pendulum the t-axis completes its circle when

$$t = 2\pi\sqrt{\frac{L}{g}},$$

and we choose $L/g = 1$ giving it period $T = 2\pi$. Then the trajectory of the system in (x, v, t) space will pierce the screen of the video monitor at exactly the same (x, v) point each time the pendulum completes its cycle. We agree to plot an (x, v) point at $t = 0$, $t = 2\pi$, $t = 4\pi$, and so on. We will call the set of $(x, v, 2n\pi)$ points so obtained a *Poincaré section*. In the case of our simple pendulum, the Poincaré section is a single point.

Add viscous damping to the pendulum system. In this case the Poincaré section will be a series of dots that lead to the attractor at (0, 0). Now we add a sinusoidal driving force. The natural period of the pendulum is not now so important. We choose to use the period of the

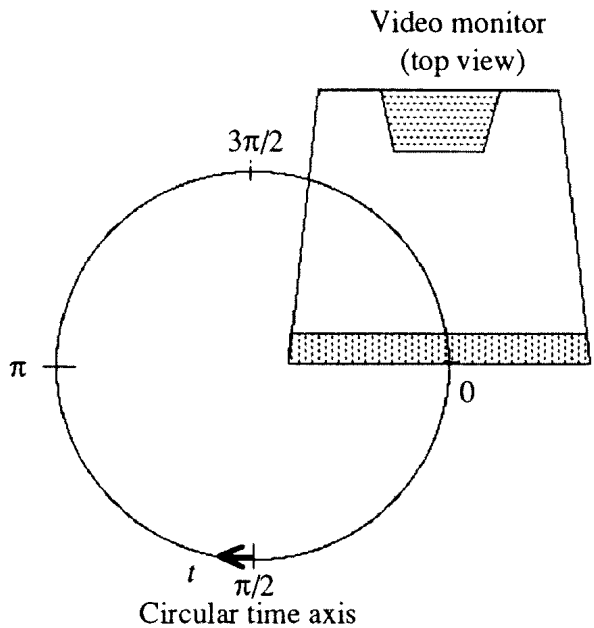


FIGURE 12.10 • A Circular Time Axis out of the Screen of the Video Monitor. The x and v axes are in the plane of the screen; the circular time axis punctures the screen at $x = 0$, $v = 0$.

driving force as the circumference of the time circle. Thus, if our driving force is periodic in sine or cosine, then we will construct the Poincaré section from points plotted every

$$t = \frac{2\pi}{\omega}$$

seconds. If the orbit in phase space is periodic with this period, then we will get one dot on the screen of the video monitor, now functioning as a Poincaré section. If the orbit has a period equal to twice the period of the driving force, then the Poincaré section will have two dots. How does the Poincaré section appear if the dynamical system is almost periodic? If the system is chaotic, the Poincaré section will consist of a pattern of dots called the attractor (actually this is one cross section of the attractor). The attractor has structure that is frequently beautiful, and it is surprising that "deterministic systems can exhibit unpredictability and apparent chaos, . . . , and yet at the same time preserve a coherent global structure."[4]

The illustration at the opening of this chapter is a Poincaré section of a damped and driven pendulum with a damping constant $c = 0.1$ and a driving force $F = 2\cos(2t)$. The domain is $[-\pi, \pi]$, the range is $[-3.4, 3.4]$, and the initial conditions are at $t = 5.5\pi/4$, $x = 1$ and $v = 3$. Different initial conditions give different pictures, and we chose this set of initial conditions because it gave a beautiful result. Because the pendulum can make a great many rotations in one or the other direction, its coordinate x is kept on the map with the instructions:

```
IF X > π THEN LET X = X - 2π
```

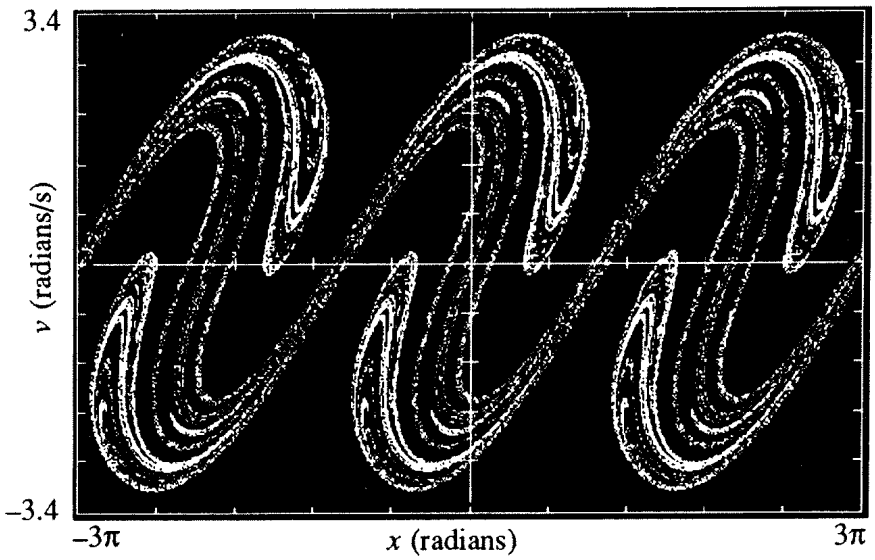


FIGURE 12.11 • Poincaré section for the Damped and Driven Pendulum. Parameters: $c = 0.16$, $F = 2$, $\omega = 2$. Initial conditions at $t = 5.4\pi/4$, $x = 1$ and $v = 3$, *dom* = $[-3\pi, 3\pi]$, and *ran* = $[-3.4, 3.4].s^{-1}$.

and

```
IF X < - π THEN LET X = X + 2π.
```

Since the driving force is periodic in 2π, these substitutions have no effect on the dynamics of the system. Figure 12.11 illustrates this periodicity of the Poincaré section in phase space; for this figure x was in $[-3\pi, 3\pi]$, v in $[-4, 4]$, $c = 0.16$, $F = 2$, and at $t = 5.5\pi/4$, $x = 1$, and $v = 3$. It may take a day or two to make a Poincaré section such as the one in Figure 12.11, depending on how fast your computer is and how many dots you wish to have on the Poincaré section.

How do we construct a Poincaré section in practice? It is convenient to choose the period of the driving force as the circumference of the time axis. Suppose that the driving force is sinusoidal with angular frequency ω, for example,

$$F = F_0 \cos \omega t.$$

Sine and cosine are periodic in 2π, so a period, T, of the driving force is obtained by solving the equation

$$\omega T = 2\pi$$

for T; $T = 2\pi/\omega$. If we want n points per cycle of the driving force, we set the step size h with the formula

$$h = \frac{2\pi}{n\omega}.$$

Thus after n integrations, time will have advanced one period of the driving force. To implement these changes, modify Algorithm 12.2 so that $h = 2\pi/\omega$, and take STEP 5 out of the STEP 4 loop. In other words, simply interchange STEPs 4 and 5. To reduce computing time, it is also beneficial to use straight-line programming with no subroutine calls. We provide such a program in the program section.

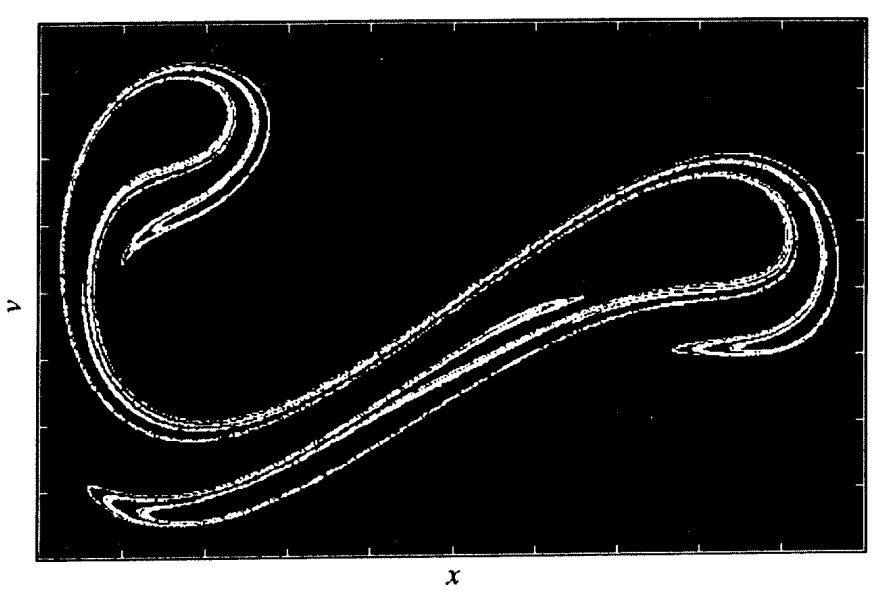
PROBLEM 12.8 Making a Poincaré Section of the Duffing Oscillator • Modify Algorithm 12.2 as suggested in the previous paragraph. For parameters, let $F = 0.68$, $\omega = 1.7$, $c = 0.24$, use a domain [–1.6, 1.6] and range [–1.5, 1.5]. For initial conditions use at $t = 0$, $x = 1$, and $v = -1$. This produces a beautiful Poincaré section.[5] Also try $F = 0.42$, $\omega = 1$, $c = 0.5$.

PROBLEM 12.9 The Road to Chaos • Repeat Problem 12.7 while observing the Poincaré section rather than phase space. Proceed slowly enough to see the Duffing oscillator approach a limit cycle. Watch it go through period doubling and then become chaotic.

12.8 The Fractal Nature of Chaotic Attractors

What we will do in this section is simply examine one Poincaré section in detail to point out a very interesting phenomena. We will use a particularly graceful Poincaré section from the work of Roger W. Rollins.[6] Figure 12.12 shows this Poincaré section of the Duffing two-well oscillator. It appears that there is fine structure. To investigate further, we enlarge a small

FIGURE 12.12 • Poincaré section of the Duffing Two-Well Oscillator. Parameters: $c = 0.24$, $F = 0.68$, $\omega = 1.7$. Initial conditions: at $t = 0$, $x = 1$, $v = -1$, $dom = [-1.5, 1.4]$, and $ran = [-0.9, 1.4]$.



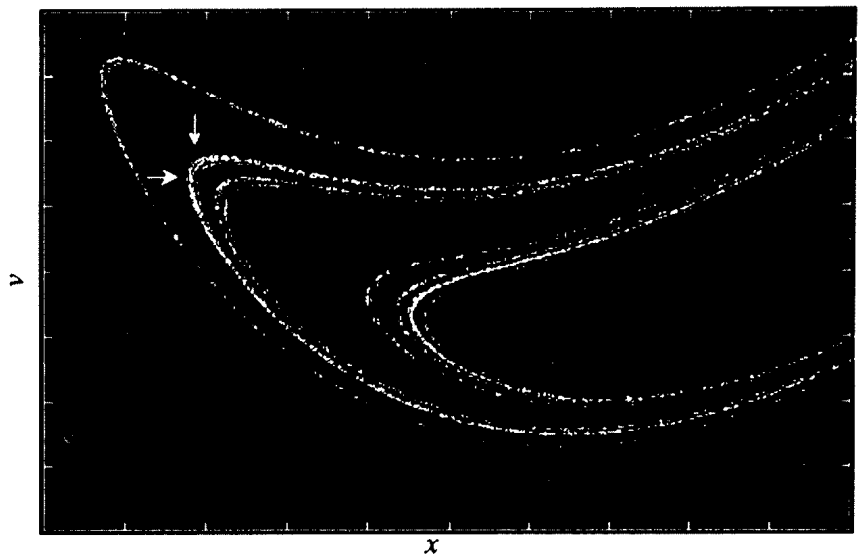
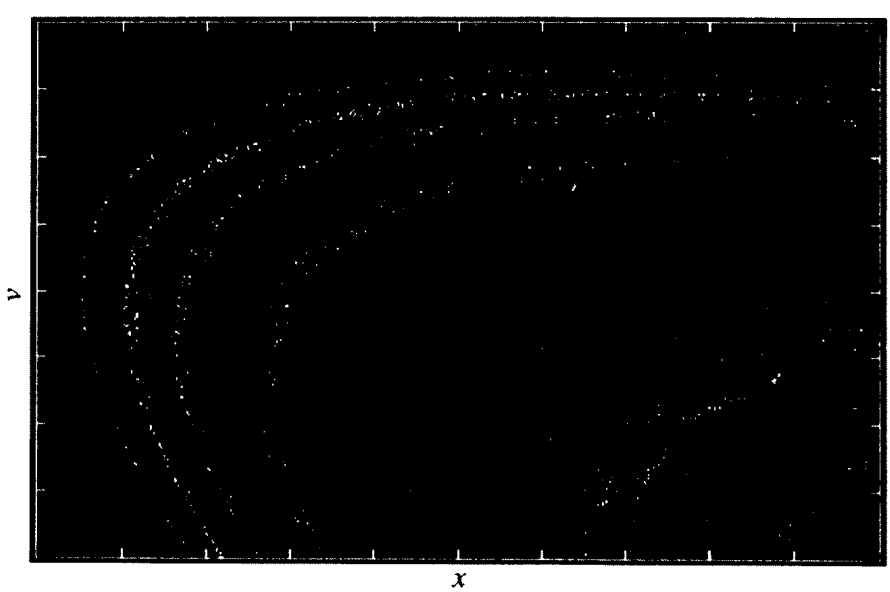


FIGURE 12.13 • Expanded View of the Lower Left Portion of the Attractor in Figure 12.12. Notice the structure that emerges. *Dom* = [−1.32, −0.8] and *ran* = [−0.8, −0.56].

section of the lower left corner, and this enlargement is shown in Figure 12.13. Not only do we verify the fine structure, we can see that there might be a similar structure on even a finer scale. Does the fine structure extend to even smaller scales? Is there self-similarity no matter how fine the scale? If so, then *the attractor is a fractal*, a set of points that shows self-similarity as the scale being used decreases. Figure 12.14 shows an expanded section of the portion of Figure 12.13 identified by arrows, and once again we observe the fine structure that suggests the attractor has a fractal structure.

FIGURE 12.14 • Expanded View of the Area in Figure 12.13 Identified by Arrows. Structure on an even smaller scale emerges. *Dom* = [−1.23, −1.20] and *ran* = [−0.649, −0.624].



Why not continue and look at finer and finer scales? Each time you pick a smaller scale, the entire attractor must be created again, and as the region studied gets smaller, fewer points fall in the region being studied. In other words, it takes too much time. There is another problem with our tentative conclusion that the attractor exhibits fractal structure. Might the structure be a computational artifact of chaos-amplified roundoff error? After all, tens of thousands of calculations are being made. After comparing analytic with numerical trajectories made with very high-precision calculations, Grebogi, Ott, and Yorke conclude: "Thus we believe that the apparently fractal structure seen in pictures such as [our Figures 12.9, 10, and 11] is real."[7]. This is quite exciting. The beautiful world of fractals tentatively touches the new and fascinating field of chaotic dynamics, and another unification of scientific endeavors occurs.

12.9 Additional Problems

PROBLEM 12.10 The Logistic Equation • There are two classic problems in chaotic dynamics. The first is the so-called logistic function, and the second is the Lorenz system. Because it is simpler, we first look at the logistic function. It is defined in terms of a *difference* equation rather than a differential equation. The difference equation is

$$x_{n+1} = \lambda x_n(1 - x_n),$$

where λ is a constant and x is constrained so that

$$0 \le x \le 1.$$

This is how the system works. Choose a value for x, call it x_0. Substitute it into the right-hand side of the logistic function and you get a new value for x, x_1. Substitute this value into the right-hand side of the equation and get a third value for x, and so on. The set of x's $\{x_0, x_1, x_2, \ldots, x_n\}$ so obtained is called an orbit.

If you are interested in applications, this equation is an exceptionally simple model of population growth for x_n butterflies (for example) who lay their eggs in year n and their x_{n+1} offspring are born in year $n + 1$. In this case, x is some fraction of the total number of butterflies. Clearly, the equation describes the situation correctly when there are no butterflies, $x_n = 0$. No butterflies, no eggs. At the other extreme, if there are too many butterflies, $x_n = 1$, then the competition for food is so great that they do not lay eggs, so x_{n+1} is again zero. Our concern is for the effect of λ on the population of butterflies. To discover what this relationship is, implement the following algorithm.

• • • •

100 times, with x from the left-hand side going into the right-hand side.
STEP 4. Iterate the logistic equation 250 more times, starting with x_{100}, this time *graphing* the points (λ, x), where

$$x \leftarrow \lambda x(1-x).$$

STEP 5. End of STEP 1 loop.

• • • •

STEP 7. Increment t:

$$t \leftarrow t + h.$$

STEP 8. End of STEP 3 loop.
STEP 9. Quit.

You should obtain a plot similar to the figure at the head of Supplement A. It is fascinating to see this graph develop; every student of chaos should see it at least once.

PROBLEM 12.12 The Spinning Magnet (Challenging) • Investigate the behavior of a compass needle free to rotate in a periodically reversing magnetic field perpendicular to the axis of the needle, a mechanical version of which is described by Briggs.[9] The differential equation describing this system is

$$\frac{d^2x}{dt^2} = -\frac{M}{I} B_0 \cos \omega t \sin x,$$

where x is the angle of the needle, M its magnetic moment, and B_0 and ω are the amplitude and angular frequency of the field.Choose $\omega = 1.5$ and $MB_0/I = 4.5$. Also try smaller values for MB_0/I. Consult the reference if you wish to build an actual mechanical model.

PROBLEM 12.13 The Simple Pendulum Modified • Investigate a slightly modified version of the pendulum. Use the equation of motion

$$\frac{d^2x}{dt^2} = -cv - \sin x + F_0 \cos \omega t.$$

Use $c = 0.2$, $\omega = 1$, and $F_0 = 2.7$.

PROBLEM 12.14 The Duffing Hard–Spring Oscillator (Challenging) • Investigate the motion of a unit mass with the following equation of motion:

$$\frac{d^2x}{dt^2} = -x - x^3 + F_0 \cos \omega t.$$

12.10 References

[1] S. N. Rasband, *Chaotic Dynamics of Nonlinear Systems* (New York: John Wiley & Sons, 1990), 1.
[2] J. Gleick, *CHAOS Making a New Science*(New York: Viking Penguin, 1987).
[3] K. Briggs, "Simple Experiments in Chaotic Dynamics," *Am. J.Phys.* 55 (12) (December 1987): 1083.
[4] E. C. Zeeman, F.R.S. in *Dynamical Chaos*, ed: M. V. Berry, I. C. Percival, and N. O. Weiss (Princeton: Princeton University Press, 1987), 3.

[5] R. W. Rollins, *Chaotic Dynamics Workbench* (New York: American Institute of Physics, 1990). See also S. De Souza-Machado, R. W. Rollins, D. T. Jacobs, and J. L. Hartman, "Studying Chaotic Systems using Microcomputer Simulations and Lyapunov Exponents," *Am J. Phys.* 58 (4) (April, 1990) 321.

[6] R. W. Rollins, *Chaotic Dynamics Workbench* (New York: American Institute of Physics, 1990).

[7] C. Grebogi, E. Ott, and J. A. Yorke, "Chaos, Strange Attractors, and Fractal Basin Boundaries in Nonlinear Dynamics," *Science*, 238, (October 30, 1987): 238.

[8] K. Briggs, "Simple Experiments in Chaotic Dynamics," *Am. J.Phys.* 55 (12) (December 1987): 1083.

12.11 Programs

PROGRAM 12.11.1 / Computer Model of a Dynamical System

```
! Algorithm 12.1, Problem 12.1 and 12.2
! Dom = [0, 3pi], ran = [-1.2,1.2]
LIBRARY "Comphys"
CALL GET_INFO (c,d,w)                                          ! Subroutine to define
                                                                 parameters.
CALL Open (xmin,xmax,ymin,ymax,wxmin,wxmax,wymin,wymax)        ! In ComPhys.
!....................
DEF f(t,x,v) = - c*v - (1 + d*cos(w*t))*sin(x)                 ! This is the equation of
                                                                 motion.
! For Duffing two-well oscillator use
!DEF f(t,x,v) = -c*v + x - x*x*x + d*cos(w*t)
!....................
LET h = 0.05                                                   ! Fix the step size, h.
LET t = 0                                                      ! Initial conditions could
LET x = 1                                                      ! also be input from keyboard.
LET v = -0.4
DO                                                             ! This is the main loop.
    PLOT t,x
       LET k1 = h*f(t, x, v)                                   ! Runge-Kutta method.
       LET k2 = h*f(t + .5*h, x + h*.5*v,v+ .5*k1)
       LET k3 = h*f(t + .5*h, x + .5*h*(v + .5*k1),v + .5*k2)
       LET k4 = h*f(t + h, x + h*v + h*k2*.5,v + k3)
       LET x = x + h*v + h*(k1 + k2 + k3)/6
       LET v = v + (k1 + k2 + k2 + k3 + k3 + k4)/6
       LET t = t + h
       IF key input then GET KEY z                             ! Get out of the loop when
       IF z = 32 then exit do                                  ! space bar is pressed.
    LOOP
CALL Label(wxmin, wxmax, wymin, wymax)                         ! In ComPhys.
END
!...........
SUB GET_INFO (c,d,w)
    LET c = 0                                                  ! These parameters may also
    LET d = 0                                                  ! be entered from the
    LET w = 2                                                  ! keyboard.
END SUB
```

PROGRAM 12.11.2 / Phase Portraits of Dynamical Systems

```
! Problems 12.4 and 12.5
! Try dom = [-3, 3], ran = [-2, 2].
LIBRARY "Comphys"
CALL GET_INFO (c,d,w)                                          ! Subroutine to define
                                                                 parameters.
CALL Open (xmin, xmax, ymin, ymax, wxmin, wxmax, wymin, wymax)
!....................
! For damped and driven pendulum use
!DEF f(t,x,v) = - c*v - (1 + d*cos(w*t))*sin(x)
! For Duffing two-well oscillator use
DEF f(t,x,v) = -c*v + x - x*x*x + d*cos(w*t)
!....................
LET n = 314
LET h = 2*pi/n                                                 ! Fix the step size, h.
FOR vo = 0.1 to 1.6 step 0.3                                   ! Make a family of phase
    LET t = 0                                                  ! portraits with vo (energy)
    LET x = 1                                                  ! as a parameter.
    LET v = vo                                                 ! Initial conditions.
    DO                                                         ! Main loop.
       PLOT x,v                                                ! Graph in phase space.
       LET k1 = h*f(t, x, v)                                   ! Runge-Kutta method.
       LET k2 = h*f(t + .5*h, x + h*.5*v,v+ .5*k1)
       LET k3 = h*f(t + .5*h, x + .5*h*(v + .5*k1),v + .5*k2)
       LET k4 = h*f(t + h, x + h*v + h*k2*.5,v + k3)
       LET x = x + h*v + h*(k1 + k2 + k3)/6
       LET v = v + (k1 + k2 + k2 + k3 + k3 + k4)/6
       LET t = t + h
       IF key input then                                       ! To go to a new member
          GET KEY z                                            ! of the family, press the
          IF z = 32 then EXIT DO                               ! space bar.
       END IF
    LOOP
NEXT vo
CALL Label(wxmin, wxmax, wymin, wymax)
END
!..........
SUB GET_INFO (c,d,w)                                           ! Determine the parameters.
    LET c = 0                                                  ! May also be entered from
    LET d = 0                                                  ! the keyboard.
    LET w = 2
END SUB
```

PROGRAM 12.11.3 / A Chaotic Dynamical System

```
! Algorithm 12.2, Problem 12.7
! For Duffing choose c = 0.1, w = 1.4, F = 0.1, h = 2π/(wn) where n = 100.
!domain [-1.6,1.6] range [-1.2,1.2]
! For pendulum try c=0.1, w = 2, F = 0.9, h same as above, x = -3π, v = 1,
```

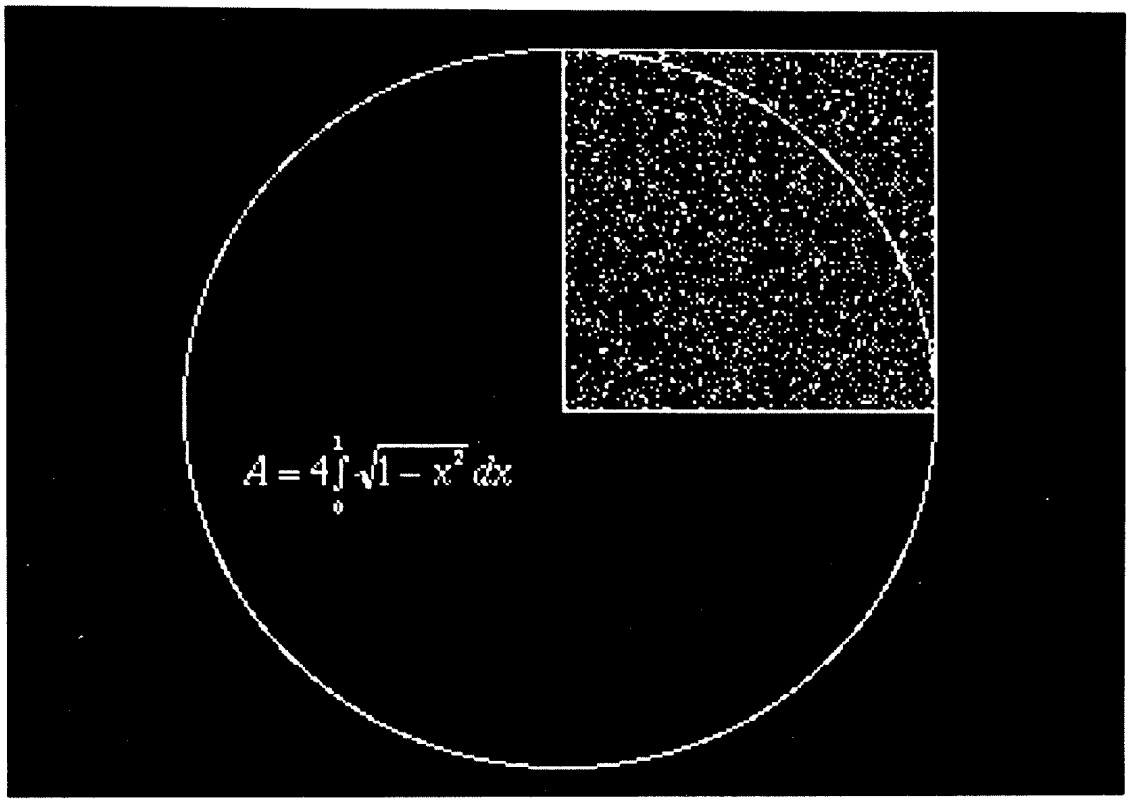
```
! dom = [-3π, 3π], ran = [-3.5, 3.5]
LIBRARY "comphys"                                          ! Graphics routines.
LIBRARY"mactools*"                                         ! Mac feature.
!...................
! For damped and driving pendulum use                      ! Pendulum force function.
DEF f(t,x,v) = - c*v - (1 + d*cos(w*t))*sin(x)
! For Duffing two-well oscillator use                      ! Duffing oscillator
!DEF f(t,x,v) = x - c*v - x*x*x + d*cos(w*t)               ! force function
!.........................
DO                                                         ! Repeat program loop.
   CALL GET_INFO (c,d,w)                                   ! Subroutine to define
                                                             parameters.
   INPUT prompt"Please enter the initial conditions, x and v. ":x,v
   CALL Open (xmin,xmax,ymin,ymax,wxmin,wxmax,wymin,wymax) ! Graphics routine.
   CALL save (wxmin, wxmax, wymin, wymax, keep$)           ! Save coordinate system.
   IF w=0 then LET w=1                                     ! Protect against w=0.
   LET n = 100
   LET h = 2*pi/(w*n)                                      ! Fix the step size, h.
   LET t = 0                                               ! Initial value.
   DO                                                      ! The main loop to plot
      FOR I = 1 to n                                       ! the trajectory in
          PLOT x,v                                         ! phase space.
          LET k1 = h*f(t, x, v)                            ! Calculate Runge-Kutta
          LET k2 = h*f(t + .5*h, x + h*.5*v,v+ .5*k1)      ! parameters.
          LET k3 = h*f(t + .5*h, x + .5*h*(v + .5*k1),v + .5*k2)
          LET k4 = h*f(t + h, x + h*v + h*k2*.5,v + k3)
          LET x = x + h*v + h*(k1 + k2 + k3)/6             ! Calculate position x.
          IF x > 3*pi then LET x = x - 6*pi                ! Keep the trajectory
          IF x < -3*pi then LET x = x + 6*pi               ! on the screen.
          LET v = v + (k1 + k2 + k2 + k3 + k3 + k4)/6      ! Calculate v.
          LET t = t + h
      NEXT i                                               ! Now check for some
      CALL macpensize(3,3)                                 ! other things.
      PLOT x,v                                             ! Plot special point each
                                                             period.
      CALL macpensize(1,1)                                 ! Back to normal dots.
      IF key input then
         GET KEY z                                         ! Read the keyboard.
         IF z = ord("u") then                              ! Increment driving force.
            LET d = d + 0.01
            SET CURSOR 2,10                                ! Print driving force.
            PRINT using " d = ##.###":d
         END IF
         IF z = ord("d") then                              ! Decrement driving force.
            LET d = d -0.01
            SET CURSOR 2,10
            PRINT using " d = ##.###":d
         END IF
         IF z = ord("c") then                              ! "c" clears screen.
            CLEAR
            CALL show (wxmin, wymin, keep$)                ! Bring back coordinate system.
            SET CURSOR 1,10
            PRINT using " d = ##.###":d
```

```
        END IF
        IF z = ord("q") then EXIT DO                    ! q is for quit main loop.
      END IF
   LOOP
   PRINT "Press q to quit."                             ! Another q quits the program.
   GET KEY z
LOOP until z = ord("q")                                 ! Or else go back and
END                                                     ! do another trajectory.
!..........
SUB GET_INFO (c,d,w)
    PRINT "We are going to study the phase-space trajectory of a "
    PRINT "damped and driven not-so-simple pendulum."
    INPUT Prompt "Please input the damping constant, c. ":c
    INPUT Prompt "Please input the amplitude of the driving force. ":d
    INPUT prompt "Please input the angular frequency of the driving function. ":w
END SUB
```

PROGRAM 12.11.4 / Poincaré Sections

```
! Poincaré sections. You may wish to modify the graphics subroutines to eliminate hash marks
! Problems 12.8 * 12.9.
LIBRARY "Comphys"
CALL GET_INFO (c,d,w)                                   ! Subroutine to define
                                                          parameters.
CALL Open (xmin, xmax, ymin, ymax, wxmin, wxmax, wymin, wymax)
!....................
! For damped and driven pendulum use
DEF f(t,x,v) = - c*v - (1 + d*cos(w*t))*sin(x)
! For Duffing two-well oscillator use
! DEF f(t,x,v) = x - c*v -x*x*x + d*cos(w*t)
!....................
LET n = 25
LET h = 2*pi/(w*n)                                      ! Set step size, h to give
LET t = 5.2*pi/4                                        ! 25 steps for each driving
!let t = 0                                              ! cycle.
LET x = 1                                               ! Initial conditions for
LET v = 3                                               ! pendulum.
LET j = 1
DO
   PLOT x,v                                             ! Plot a point
   FOR i = 1 to n                                       ! Calculate n points before
       LET k1 = h*f(t, x, v)                            ! plotting another.
       LET k2 = h*f(t + .5*h, x + h*.5*v,v+ .5*k1)      ! Runge-Kutta.
       LET k3 = h*f(t + .5*h, x + .5*h*(v + .5*k1),v + .5*k2)
       LET k4 = h*f(t + h, x + h*v + h*k2*.5,v + k3)
       LET x = x + h*v + h*(k1 + k2 + k3)/6
       LET v = v + (k1 + k2 + k2 + k3 + k3 + k4)/6
       LET t = t + h
       IF x > pi then LET x = x - 2*pi                  ! This is unnecessary for
```

```
      IF x < - pi then LET x = x + 2*pi                    ! Duffing two-well oscillator.
   NEXT i
LOOP
CALL Label(wxmin, wxmax, wymin, wymax)
END
!..........
SUB GET_INFO (c,d,w)
    LET c = .1
    LET d= 2
    LET w = 2
END SUB
```



Monte Carlo Integral

13 / Doing Physics with Random Numbers

13.1 Introduction

A review of the following topics may be helpful in this chapter:

- Areas and volumes found by integration
- Moments of inertia
- Mean and standard deviation

Most programming languages and spreadsheets allow the user to generate random numbers. For example, in many versions of BASIC the instruction

```
y = RND
```

will make y a random number in the interval [0, 1). In other words,

$$0 \le y < 1.$$

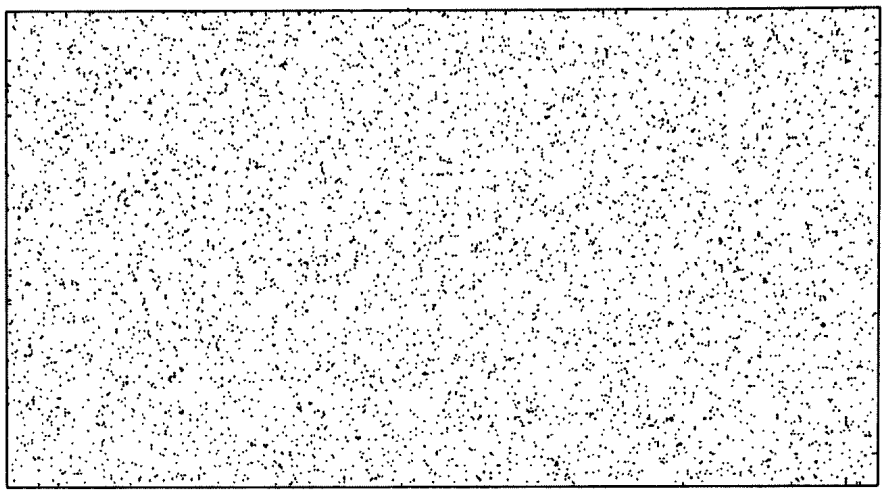


FIGURE 13.1 • Random Numbers (y-axis) Uniformly Distributed in the Interval [0, 1).

Writing and running a simple program to generate and plot random numbers with the random number as ordinate (y) and the sequence of the numbers as abscissa (x) will give you some intuition for random numbers. We have done this to generate Figure 13.1. A sequence of random numbers (y values) are equally spaced along the x-axis. The distribution of random numbers in Figure 13.1 is known as a *uniform* distribution. Another kind of distribution of random numbers which is called *Gaussian* may be generated with Algorithm 13.1.[1] We will discuss this distribution in more detail later in the chapter.

We again point out that random numbers are generated by some algorithm, and this algorithm would provide the same sequence of random numbers over and over again if each time it were not initiated by a new number called a *seed*. The seed itself should be random, and getting the seed from the calendar and/or clock in the circuitry of the computer makes the probability of getting identical seeds very small.

Random numbers can be used to evaluate integrals and to model processes such as radioactive decay or any other physical process in which randomness is inherent. The process that generated the figure at the opening of this chapter will be our starting point. (Before getting started, we note that most of the formulas in this chapter will be given without a derivation. Consult any textbook on statistics for a rigorous and detailed development of the formulas.[2])

ALGORITHM 13.1 / Generating a Gaussian Distribution of N Random Numbers

STEP 1. Begin a new random number sequence with a new seed.
STEP 2. Specify the number of random numbers to be generated, N.
STEP 3. Loop N times.
STEP 4. Initialize a sum variable, x. Let $x = 0$.

STEP 5. Loop 12 times.
STEP 6. Generate a random number, y.
STEP 7. Add it to the sum variable, x:

$$x \leftarrow x + y.$$

STEP 8. End of STEP 5 loop.
STEP 9. Subtract six from the sum:

$$x \leftarrow x - 6.$$

STEP 10. The new random number is x, the result of the preceding subtraction.
STEP 11. Plot or tabulate x, or use it in another routine.
STEP 12. End of STEP 3 loop.
STEP 13. Quit.

PROBLEM 13.1 Generating a Gaussian Distribution of Random Numbers • Write a program to implement Algorithm 13.1. Graph the random numbers along the y-axis as we did in Figure 13.1. Describe the difference between uniformly distributed and Gaussian distributed random numbers.

13.2 Using Random Numbers to Evaluate Integrals

We begin by using uniformly distributed random numbers to find the area of a circle, and we will include graphics to help make the technique more intuitive; refer to the figure at the opening of this chapter. A circle of radius one is inscribed in a square with a side length of two. We consider only the upper-left hand quarter of this square, which is a one-by-one square. Two random numbers are generated and they become the coordinates, (x, y), of a point in the one-by-one square; m such points are generated. The variable n counts the number of points in the quarter of a circle of radius one centered at the lower-left-hand point on the square. (Refer again to the figure at the opening of this chapter.) These points lie in the square *and* their coordinates satisfy the relationship $x^2 + y^2 \leq 1$. If n points lie in the quarter circle and m points lie in the one-by-one square, then the ratio of the area of the circle, A_c, to the area of the square, A_s, is expressed with the relationship

$$\frac{A_c}{A_s} \cong 4\frac{n}{m}.$$

The area of the one-by-one square is one, therefore

$$A_c \cong 4\frac{n}{m}.$$

In calculus you find areas by integration. In particular, the area of a quarter circle of radius one is

$$A = \int_0^1 \sqrt{1-x^2}\, dx.$$

Thus we have found an approximation for this integral, namely,

$$\int_0^1 \sqrt{1-x^2}\, dx \cong \frac{n}{m},$$

where n is the number of points that lie in the quarter circle of unit radius and m is the number of points that lie in the one-by-one square. Algorithm 13.2 describes in more detail the process needed to calculate this integral.

ALGORITHM 13.2 / Finding the Area of a Unit Circle

STEP 1. Start a new random number sequence.
STEP 2. Loop m times where m is the number of random numbers to use.
  STEP 3. Choose random numbers for x and y such that $0 \le x < 1$ and $0 \le y < 1$.
  STEP 4 If $x^2 + y^2 \le 1$, then increment area counter, n:

$$n \leftarrow n + 1.$$

STEP 5. End of STEP 2 loop.
STEP 6. Output $4(n/m)$, our estimate for the area of a circle.
STEP 7. Quit.

PROBLEM 13.2 Finding the Area of a Unit Circle • Write a program to implement Algorithm 13.2. Make $m = 1000$. By what percent does the calculated area differ from the actual area?

The solution to Problem 13.2 implements the algorithm to find the area of the circle. Each run with a series of m random numbers, $m = 1000$ in Problem 13.2, constitutes a *measurement* of the area. How can we increase the precision of our results? There are two ways: Either increase the number of points in a measurement or increase the number of measurements. The second way allows us to make a calculation of a quantity called the *standard deviation*. The standard deviation, to be described in more detail in a subsequent section, is an estimate of how precise the area is calculated with this technique. The significance of the standard deviation is as follows: There is a 68 percent probability that the true area is somewhere between the value of the area calculated by the program and plus or minus one standard deviation. This is expressed by writing the answer as

$$A \pm \sigma,$$

where σ is the standard deviation. At least 25 measurements should be made for the standard deviation to be meaningful.

To calculate σ, we use the formula

$$\sigma = \sqrt{\frac{\sum_{i=1}^{i=N} (x_i - < x >)^2}{N}},$$

where x_i signifies the result of a single measurement, and $< x >$ symbolizes the arithmetic mean of N measurements. With some algebra, it can be shown that

$$\sigma = \frac{\sqrt{N \sum x_i^2 - \left(\sum x_i\right)^2}}{N},$$

where it is understood that the summation is over the N measurements, as expressed in the previous formula. The latter formula is frequently more convenient, and it is the one we use in most of the calculations that follow.

It is clear that σ is proportional to $\frac{1}{\sqrt{N}}$, but it is less obvious that σ is proportional to $\frac{1}{\sqrt{M}}$, where M is the number of random points used to make a measurement. Once we have made a series of measurements, these facts allows us to predict the total number of measurements we would need to achieve a given precision.

PROBLEM 13.3 Finding the Area of an Ellipse • Use the techniques just described to find the area of an ellipse with a semimajor axis of four and a semiminor axis of three. The equation of an ellipse is

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1,$$

where a is the semimajor axis and b is the semiminor axis. Hint: Find the area in the first quadrant and then multiply by four to get the entire area. Compare this with the true area, πab.

Here is another example of random-number integration. We will evaluate an integral for which the integrand has no antiderivative, namely

$$\int_0^1 e^{x^2}\, dx.$$

In the interval of integration the integrand lies between 0 and e. Construct a rectangle having dimensions one-by-e and through which the $y = e^{x^2}$ curve passes, as illustrated in Figure 13.2. (There are other options.) Choose a sequence of random numbers x with $0 \le x < 1$ and y with $0 \le y < e$. How do you get random numbers in the interval $[0, e)$? Your programming language or spreadsheet will have a function called RND or something similar that produces random numbers between 0 and 1. If we let

$$y = e * \text{RND},$$

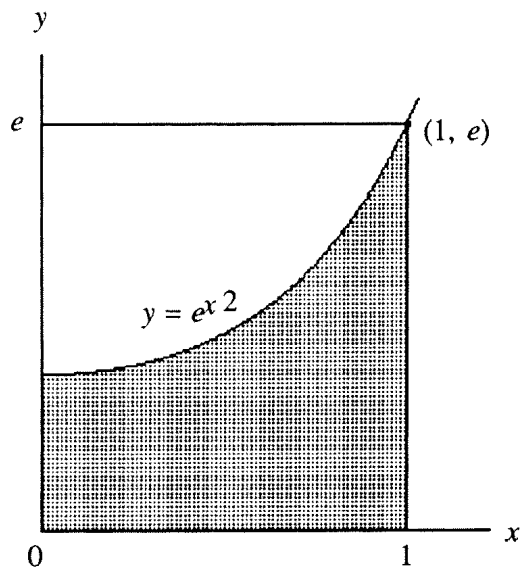


FIGURE 13.2 • Finding the Area under the curve of $y = e^{x^2}$.

we will obtain random numbers in the interval $[0, e)$. Counting the points under the curve, namely those with

$$y \leq e^{x^2},$$

and calculating the ratio of the number of points under the curve to the total number of points in the one-by-e rectangle will determine the area. Multiply this ratio by the area of the rectangle, namely e, to find the area under the curve.

The techniques we have described in this section are usually called *Monte Carlo techniques*. This section ends with another problem.

PROBLEM 13.4 Evaluating an Integral Using Monte Carlo Techniques • Using Monte Carlo techniques, find

$$\int_1^3 \sqrt{x^3 - 1}\, dx$$

to three significant digits. Can you check your answer by finding an antiderivative?

13.3 Finding Masses and Moments of Inertia

13.3.1 Masses

We can use Monte Carlo techniques to find the volume, center of mass, or moment of inertia of a solid. Although the technique is generally not used with simple shapes, we choose to illustrate the technique with a sphere. Imagine a unit sphere, $r = 1$, that centered at the origin and inscribed in a cube with a side-length of two centered on the origin. Assume that both cube

and sphere have a density of one. The volume of the cube is $2^3 = 8$. We divide the cube into N parts, so each volume element is $1/N$ and, since the density is one, each element of volume has mass dm where

$$dm = \frac{1}{N}.$$

To find the mass of the sphere, we pick J *random* points by choosing random numbers between minus one and one for each of the three coordinates, x, y, and z. In order to lie in the sphere, the coordinates must satisfy the condition

$$x^2 + y^2 + z^2 \leq 1.$$

otherwise they lie in that part of the cube outside of the sphere. If a point lies within the sphere, we add dm to the mass of the sphere, and not otherwise. When all J points have been tested, the sum of the dm terms is the mass of the sphere. In short, we have found

$$\int \rho \, dV,$$

where the integration is taken over the volume occupied by the sphere and $\rho = 1$. A procedure to compute this integral is given in Algorithm 13.3.

ALGORITHM 13.3 / Using Monte Carlo Techniques to Find the Volume (Mass) of a Sphere

STEP 1. Decide how many volume measurements, K, to make. ($K \geq 25$.)
STEP 2. Decide on the number of points, N, to use in each measurement. ($N = 1000$.)
STEP 3. Let $\rho = 1$; then

$$dm = \rho dV = \frac{1}{N}.$$

STEP 4. Loop K times.
  STEP 5. Initialize the variables: Set $V = 0$, $\langle V \rangle = 0$, and $V_{SQ} = 0$. ($\langle V \rangle$ and V_{SQ} are used to calculate σ).
  STEP 6. Begin a random number sequence with a new seed.
  STEP 7. Loop N times.
    STEP 8. Choose three random numbers, (x, y, z), between zero and one.
    STEP 9. If

$$x^2 + y^2 + z^2 \leq 1,$$

then increment V, the volume variable, by dm:

$$V \leftarrow V + dm.$$

STEP 10. End of STEP 7 loop. This completes one measurement of V.
STEP 11. Accumulate volume measurements into an average:

$$<V> \leftarrow <V> + V.$$

STEP 12. Also accumulate V_{SQ} to calculate σ:

$$V_{sq} \leftarrow V_{sq} + V^2.$$

STEP 13. End of STEP 4 loop. This completes K measurements of V.
STEP 14. Calculate and print the average volume:

$$<V> \leftarrow \frac{<V>}{K}.$$

STEP 15. Calculate and print the standard deviation σ, where

$$\sigma = \frac{\sqrt{V_{sq} - <V>^2}}{K}.$$

STEP 16. Quit.

PROBLEM 13.5 Finding the Volume of an Ellipsoid • The equation for the surface of an ellipsoid centered on the origin is

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1.$$

Find its volume. Hint: The ellipsoid may be inscribed in a solid with six rectangular faces.

PROBLEM 13.6 Volume of a Sphere with a Cylindrical Hole • Suppose that a cylindrical hole of radius r is drilled along the z-axis through the center of a sphere of radius R. What is the volume of the sphere minus the hole? We simply omit all those volume elements dm whose coordinates satisfy the inequality

$$x^2 + y^2 \le r^2,$$

where, of course, $r < R$, the radius of the sphere. Modify Algorithm 13.3 to include this calculation.

13.3.2 Moments of Inertia

It is relatively easy to modify Monte Carlo techniques to deal with a variety of problems. Suppose that we wish to find the moment of inertia of a sphere about the z-axis. Refer to the

chapter on rotation in your textbook to see how moments of inertia are defined. To find I_z, the moment of inertia about the z-axis, we must evaluate the integral

$$I_z = \int \rho(x^2 + y^2)\, dV.$$

Once again, we assume $\rho = 1$. STEP 9 of Algorithm 13.3 must be modified as follows:

STEP 9. If

$$x^2 + y^2 + z^2 \leq 1,$$

then, increment V, the volume variable, by dm:

$$V \leftarrow V + dm,$$

and increment the moment of inertia variable by $(x^2 + y^2)dm$:

$$dI_z \leftarrow dI_z + (x^2 + y^2)dm.$$

It would be wise to include calculations of the average of I_z and its standard deviation. With $N = 1000$ and $K = 25$, we obtained $I_z = 1.70 \pm 0.07$ kgm^2. The theoretical answer is $2/5mR^2$, and given that $m = \rho(4/3)\pi R^3$ with $\rho = 1$ kgm^{-3}, $I_z = 1.6755$ kgm^2, well within our estimated uncertainty.

PROBLEM 13.7 Moment of Inertia of an Ellipsoid • Refer to Problem 13.3 and find the moment of inertia of this ellipsoid about the x-axis, the y-axis, and the z-axis. Take $N = 1000$. Also calculate the average of I and the standard deviation for 25 measurements.

13.4 Noise

13.4.1 Calculating the Mean and Standard Deviation of a Noise Distribution

Associated with every measurement is error or noise. Physics professors average your examination grades because there may be noise present (anyone can have a bad day) and professors hope that the average is a better estimate of your ability than any individual grade. Students frequently ask to "throw out their lowest score," I suppose with the assumption that it was produced by noise. It would appear to be fair only if they also agree to throw out their best score. In any case, this section is about measurements and noise and how to recover measurements from noise.

Random numbers may be used to represent noise. To model noise, we may start with a measurement x_i of a variable v, where

$$x_i = v + n_i,$$

and n_i is a random noise (number) value. An average of the measurements of x_i from $i = 1$ to N will approach v as the number of measurements, N, increases.

In Figure 13.1 we find an example of *uniform* noise pulses, x_i (along the vertical axis) with i along the horizontal axis. In this figure the noise is uniformly distributed between zero and one. To generate *Gaussian noise*, use Algorithm 13.1. It is almost always Gaussian noise with which physicists are dealing. If you solve Problem 13.1, you will get an intuitive feeling for the fact that Gaussian random numbers cluster around a central value, called the mean. But they are also dispersed around this mean; in fact, a few numbers are found quite far from the mean. The measure of dispersion or variability used to describe a Gaussian (or normal) distribution is the *standard deviation*, σ. This is the same standard deviation we used in the previous section. Specifically, with a Gaussian distribution of noise pulses, the probability of a pulse of amplitude x_i is given by the equation

$$p(x_i) = Ae^{\left[\frac{(x_i - \langle x \rangle)^2}{2\sigma^2}\right]},$$

where $< x >$ is the mean of the pulses and σ is the standard deviation. You may regard $p(x_i)$ as proportional to the probability of finding a specific *deviation*, d_i, from the mean, where

$$d_i = x_i - < x >.$$

The constant A is chosen so the integral of $p(x)$ over the interval $(-\infty, \infty)$ is one (the number x is certain to fall somewhere in the set of real numbers).

To calculate σ from experimental data, we note the standard deviation is defined as the *root mean square* of the deviations from the mean. In the form of a formula this is

$$\sigma = \sqrt{\frac{\sum (x_i - < x >)^2}{N}},$$

where the summation is from $i = 1$ to N, which is the number of measurements. Using algebra, we can show that

$$\sigma = \frac{1}{N}\sqrt{N\sum x_i^2 - \left(\sum x_i\right)^2}.$$

Thus σ can be found from experimental data with either of the last two formulas. In preceding programs we have used the second form and in the programs that follow we will continue to use it.

The Gaussian distribution function allows us to calculate the probability that a given value of a random number x_i will fall in a specific interval. For example, it can be shown that 34 percent of the measurements will fall between the mean and one standard deviation above

(or below) the mean, and, in fact, if σ is known it is possible to calculate the percentage of the measurements between any two x values. This type of information is tabulated in books on statistics and in mathematical handbooks.

Our purpose in introducing the concept of noise is to illustrate how, through averaging, a signal can be extracted from a noisy measurement. The computer allows us to model this process because the random number generator simulates noise, which can be turned into Gaussian noise with a simple procedure. Adding the noise to a simulated signal produces a noisy signal, the kind that experimental physicists, engineers, and astrophysicists *always find* in the real world. We will illustrate how the signal can be extracted from the noisy signal by averaging. Our study will help build intuition about signals, noise, and the data processing that is required in experimental work.

We began this chapter by showing how to generate Gaussian noise; next we show how to test the distribution of noise to see if it is, indeed, Gaussian. The procedure in Algorithm 13.1 will produce a Gaussian distribution of random numbers with a zero mean and a standard deviation of one. Algorithm 13.4 is a procedure to calculate the mean and the standard deviation of a set of random numbers generated by the procedure in Algorithm 13.1.

ALGORITHM 13.4 / A Procedure to Calculate the Mean and the Standard Deviation of a Set of N Random Numbers

STEP 1. Loop N times.

STEP 2. Accept a random number x (from some prior procedure, e.g. Algorithm 13.1).

STEP 3. Sum the random numbers with a variable S:

$$S \leftarrow S + x.$$

STEP 4. Sum the squares of the random numbers with a variable r:

$$r \leftarrow r + x^2.$$

STEP 5. End of STEP 1 loop.

STEP 6. Calculate the mean $< x >$,

$$< x > = \frac{S}{N}.$$

STEP 7. Calculate σ.

$$\sigma = \frac{1}{N}\sqrt{Nr - S^2}.$$

STEP 8. Output the results.

STEP 9. Quit.

PROBLEM 13.8 Finding the Average and Standard Deviation of a Noise Distribution • Extend the solution of Problem 13.1 to calculate the mean and standard deviation of a Gaussian distribution of random numbers as generated by Algorithm 13.1. Is $< x > = 0$? Is $\sigma = 1$?

13.4.2 The Form of a Gaussian Noise Distribution

The solution to Problem 13.8 will show that the mean of the distribution of noise generated with Algorithm 13.1 is zero and $\sigma = 1$. That does not prove that the distribution is Gaussian. Other distributions can have the same characteristics. What we would like to do is to compare the *shape* of the distribution with the shape of a theoretical Gaussian distribution function. To do this, we need a more elaborate sampling process. Algorithm 13.5, a rather complex procedure, outlines a method to sample the numbers generated by a procedure that generates random numbers (noise pulses) to see how many numbers (noise pulses) fall in specific intervals. The procedure outlined could be replaced by 21 IF... THEN... statements, but we thought we could do something more elegant. If the spreadsheet is your tool, then the IF... THEN... approach might be more useful. In any case, the procedure calls for a graph of these results so that you can see if the distribution approximates the bell-shaped Gaussian distribution. In particular, the theoretical distribution will also be graphed so that you can compare the experimental distribution of noise pulses with the theoretical Gaussian distribution.

A new concept introduced in Algorithm 13.5 is the use of an *array* to store and process information. The function of this algorithm is to record the number of random numbers (noise) that falls in 21 identical-width intervals as follows:

$$-3.05 \le x < -2.85,$$

$$-2.85 \le x < -2.55,$$

$$\vdots$$

$$-0.45 \le x < 0.15,$$

$$\mathbf{-0.15 \le x < 0.15,}$$

$$0.15 \le x < 0.45,$$

$$\vdots$$

$$2.55 \le x < 2.85,$$

$$2.85 \le x < 3.05.$$

where we have highlighted the central interval to emphasize the symmetry. We will use 10 intervals on each side of the central interval, each with an arbitrary width 0.35. (You may wish to experiment with other widths.)

It is useful to assign each element of an array,

$$A(1),\quad A(2),\quad A(3),\quad \cdots,\quad A(21),$$

to each of the identical-width intervals, and let each element store the number of numbers that fall in its interval. The central interval is associated with $A(11)$. We can also symbolize the array in this way:

$$\cdots\quad A(9),\quad A(10),\quad A(\mathbf{11}),\quad A(12),\quad A(13),\quad \cdots$$

to show its similarity to the set of x-intervals listed above.

ALGORITHM 13.5 • Generating, Sampling, and Displaying a Noise Distribution

STEP 1. Loop N times, where N is the total number of random numbers generated.

STEP 2. Generate a random number x (see Algorithm 13.1).

STEP 3. Sum the random numbers with a variable S:

$$S \leftarrow S + x.$$

STEP 4. Sum the squares of x with a variable R:

$$R \leftarrow R + x^2.$$

STEP 5. Question 1. Is $x \geq 0$?

Yes. (Handle non-negative x here, negative x at STEP 17.)

STEP 6. Subtract 0.15 (half-width of central interval) from x:

$$x \leftarrow x - 0.15.$$

STEP 7. Question 2. Is $x < 0$?

Yes. (If so, it is in the first interval to the right of the mean.)

STEP 8 Increment array variable $A(11)$:

$$A(11) \leftarrow A(11) + 1.$$

No. (If not, x is in one of the other intervals.)

STEP 9. Begin loop over I from $I = 12$ to $I = 22$.

STEP 10. Replace x with $x - 0.3$. (Intervals are 0.3 in width.)

$$x \leftarrow x - 0.3.$$

STEP 11. Question 3. Is $x < 0$?

Yes. (Then x is in the interval identified by the current value of I.)

STEP 12. Increment array variable $A(I)$:

$$A(I) \leftarrow A(I) + 1.$$

STEP 13. Exit the loop. (Continue at STEP 15.)
No. (Then x is in another interval.)
STEP 14. Increment I:

$$I \leftarrow I + 1.$$

STEP 15. End of STEP 9 loop.
STEP 16. End of Question 2.
No. (Handle negative x here.)
STEP 17. Add 0.15 (half-width of central interval) to x:

$$x \leftarrow x + 0.15.$$

STEP 18. Question 4. Is $x \geq 0$?
Yes. (If so, x is in the first interval to the left of the mean.)
STEP 19. Increment array variable $A(11)$:

$$A(11) \leftarrow A(11) + 1.$$

No. (If not, x is in one of the other intervals.)
STEP 20. Begin loop over I from $I = 10$ to $I = 1$.
STEP 21. Replace x with $x + 0.3$. (Intervals are 0.3 in width.)

$$x \leftarrow x + 0.3.$$

STEP 22. Question 5. Is $x \geq 0$?
Yes. (Then x is in the interval identified by I.)
STEP 23. Increment array variable $A(I)$:

$$A(I) \leftarrow A(I) + 1.$$

STEP 24. Exit the loop. (Continue at STEP 27.)
No.
STEP 25. Decrement I:

$$I \leftarrow I - 1.$$

STEP 26. End of Question 5.
STEP 27. End of STEP 20 loop.
STEP 28. End of Question 4.
STEP 29. End of Question 1.
STEP 30. End of STEP 1 loop.
STEP 31. Calculate and print the mean and the standard deviation.
STEP 32. Plot the non-zero array elements $A(I)$ versus I for $I = 1$ to 22.
STEP 33. Plot the theoretical Gaussian curve. The theoretical values are given by the expression

$$A(I) = A(11)e^{\left[\frac{(I-11)^2}{2}\right]}.$$

STEP 34. Quit.

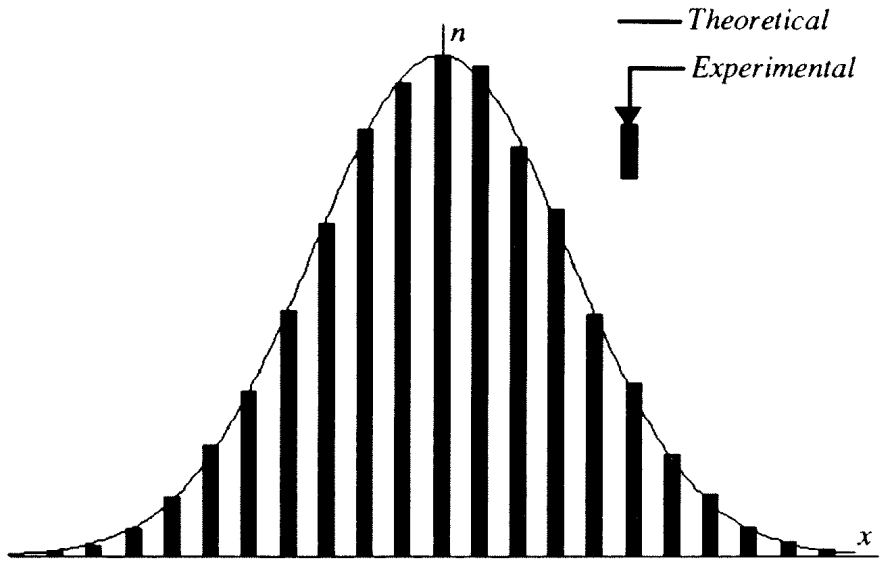


FIGURE 13.3 • A Gaussian Distribution of Random Numbers.

PROBLEM 13.9 Testing a Distribution of Random Numbers (Challenging) • Write a program to implement Algorithm 13.5. Test both a uniform distribution and Gaussian distribution of random numbers or noise pulses. Our results to test whether or not the noise distribution is Gaussian are shown in Figure 13.3. The thin line is a theoretical Gaussian distribution function. We conclude that, indeed, the noise pulses do have a Gaussian distribution.

PROBLEM 13.10 Graphing the Gaussian Distribution Function • It is instructive to graph the Gaussian distribution

$$p(x_i) = Ae^{\left[\frac{(x_i - \langle x \rangle)^2}{2\sigma^2}\right]},$$

for several values of σ. This exercise will illustrate that the most likely measurement is the mean, and that the dispersion or variability is related to σ, the standard deviation. The larger the value of σ, the larger the dispersion and the more "spread out" is the graph.

PROBLEM 13.11 Gaussian Noise Parameters • Use the solution to Problem 13.7 to study the distribution of random numbers if in the routine to generate the numbers you sum 24 random numbers instead of six. What happens?

13.4.3 Extracting a Signal from Noise

In this section we will illustrate how a signal may be extracted from a signal "buried" in noise. The process to do this is, of course, averaging. Assume that the simulated signal we are dealing with is the one shown in Figure 13.4. This signal is merely a graph of

$$y = \frac{\sin x}{x}.$$

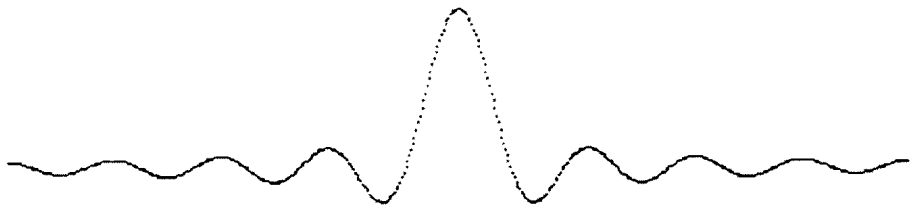
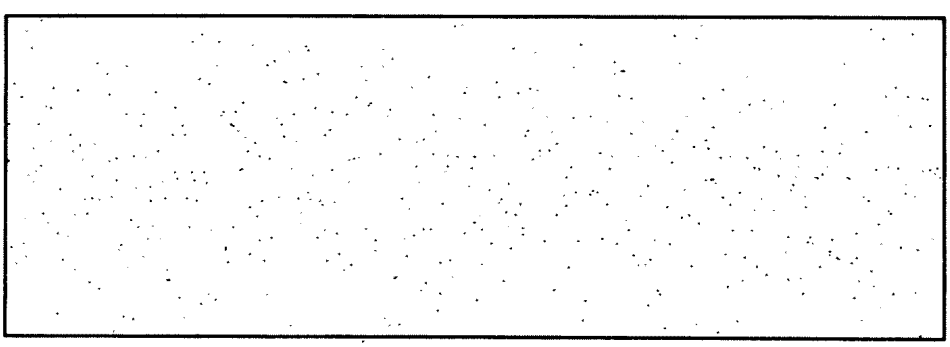


FIGURE 13.4 • A Simulated Signal, y = (sin x)/x.

If you wish, think of the x-axis as the frequency axis and the y-axis as intensity, and then imagine a radio astronomer searching the electromagnetic spectrum for a spectral line of an organic molecule in the interstellar medium. The "line" consists of electromagnetic radiation at a definite frequency, but we will assume that the radio instrumentation smears it into the shape shown in Figure 13.4. We assume that microwave noise from other sources (microwave ovens, cosmic background noise, the sun, etc.) masks the signal, as shown in Figure 13.5. Figure 13.5 is obtained when Gaussian distributed noise with zero mean and a standard deviation of approximately four is added to the signal in Figure 13.4, which is also substantially reduced. Notice that almost no trace of the signal is apparent. We will call one of the observations such as the one found in Figure 13.5 a record.

How do astronomers search for the spectral line? Their instruments scan this same frequency interval over and over again, and each time a new record is obtained an average with one more set of points is calculated. The procedure of Algorithm 13.6 models this process by creating successive records with the signal and fresh noise added, after which it averages the new data with the old. You are to observe the effect of averaging while the process proceeds. Do this and notice the decrease in the noise as more data are averaged. Let the program continue at least until you can recognize the signal, and if possible, let the computer run overnight.

FIGURE 13.5 • The Signal in Figure 13.4 Substantially Reduced with Noise Added.



ALGORITHM 13.6 / Extracting a Signal From Noise

STEP 1. Initialize two array variables, $A(I)$ and $B(I)$, $I = 1$ to 500.
STEP 2. Generate the signal and store it in a 500 element array, $A(I)$.
A. Loop from $I = 1$ to 500.
B. Let I determine x where

$$x = \frac{I - 250.5}{3\pi}.$$

C. Set

$$A(I) = \frac{\sin x}{x}.$$

D. End of loop over I.
STEP 3. Initialize a loop counter k to count the number of records we will average; $k = 1$.
STEP 4. Begin the main loop with k as counter.
STEP 5. Loop from $I = 1$ to 500.
STEP 6. Initialize x: $x = 0$.
STEP 7. Begin loop to generate Gaussian noise. Make 12 loops.
STEP 8. Get a random number r where $0 \le r < 1$.
STEP 9. Add it to x:

$$x \leftarrow x + r.$$

STEP 10. End of STEP 7 loop.
STEP 11. Subtract 6 from x to make the mean zero:

$$x \leftarrow x - 6.$$

STEP 12. Multiply x by three to increase the dispersion of the noise distribution.
STEP 13. Add noise to the signal and to the previous average:

$$B(I) \leftarrow B(I) + A(I) + x.$$

STEP 14. Calculate a new average:

$$B(I) \leftarrow \frac{B(I)}{k}.$$

STEP 15. Graph point $[I, B(I)]$.
STEP 16. End of loop over I beginning at STEP 5.
STEP 17. Increment the main loop counter, k:

$$k \leftarrow k + 1.$$

STEP 18. End of main loop beginning at STEP 4.

PROBLEM 13.12 Averaging Records to Reduce the Noise • Write a program to implement Algorithm 13.6. Start the program and watch the output as the algorithm is executed. If possible allow the program to run overnight.

To find some measure of the variability that the noise adds to the data, you can select one of the points in the array that stores the averaged signal and calculate not only the mean but also the standard deviation. As an alternative, you can calculate the deviations from the true signal for all 500 points in a record, square these, sum the squares, divide by 500, and then extract the square root, a process that is also an estimate of the standard deviation of the noise. This gives a better estimate of σ, especially in the early stages when only a few records are averaged. After STEP 19, insert the following steps into Algorithm 13.6.

STEP A. Initialize a sum variable, S. Set $S = 0$.
STEP B. Loop from $I = 1$ to 500.
STEP C. Sum the squares of the deviations:

$$S \leftarrow S + [A(I) - B(I)]^2 .$$

STEP D. End of STEP C loop.
STEP E. Extract the square root:

$$S \leftarrow \sqrt{S} .$$

STEP F. Print S, the result.
STEP G. Pause long enough to read the answer on the screen.

This completes our analysis of a signal buried in noise.

PROBLEM 13.13 The Relationship of σ to the Number of Records in the Average • Add STEP A - STEP F (above) to Algorithm 13.6, and make the appropriate modifications to the solution to Problem 13.12. With your program running, jot down the number of records and the standard deviation. Do this for at least 50 records. Then make a graph σ versus N, the number of records. Also make a graph of log σ versus log$N$, called a log-log graph. How does σ change with N, the number of records? Is σ proportional to $\sqrt{N}$? What is the value of σ? With some cleverness you can incorporate both the calculation of σ and the log-log plot into your program, omitting the display of the updated average. It is informative to let this program run for 12 to 24 hours.

13.5 Additional Problems

PROBLEM 13.14 Moment of Inertia of a Washer • Use the Monte Carlo techniques to find the moment of inertia of a disk of radius R with a hole of radius r drilled through its center. The moment of inertia is to be taken around an axis through the center of both hole and disk, perpendicular to the surfaces. Compare your results with the theoretical moment of inertia $I = 1/2M(R^2 + r^2)$, where M is the mass of the washer. What is the moment of inertia of a disk about an axis through its center with a square of side length d cut from its center?

PROBLEM 13.15 Variation of σ with N • Use the solution to Problem 13.2 to make 25 measurements of the area of a unit circle. Start with 25 measurements with 50 points, 100 points, 500 points, 1000 points, and 25 measurements with 5000 points (this one will take some time). Calculate σ for each set of 25 measurements, and make a graph of σ versus $\log N$, where N is the number of points in the measurement. What is the relationship of σ to N?

PROBLEM 13.16 The Frequency Distribution of Uniform Noise • Modify Algorithm 13.7 to generate "ordinary" noise and make a graph similar to Figure 13.4, for a uniform distribution of noise rather than a Gaussian distribution. In your own words describe the difference between uniform noise and Gaussian noise.

13.6 References

[1] H. Gould and J. Tobochnik, *Computer Simulation Methods* (Reading, MA: Addison-Wesley, 1988), 375.

[2] D. S. Edmonds, Jr., *Cioffari's Experiments in College Physics* (Lexington, MA: D. C. Heath, 1983) is a good starting point as is D. W. Preston and E. R. Dietz, *The Art of Experimental Physics* (New York: John Wiley, 1990). J. W. McGhee, *Introductory Statistics* (St. Paul, MN: West Publishing, 1985) is just one of many textbooks. Textbooks for engineering statistics courses are also excellent resources.

13.7 Programs

PROGRAM 13.7.1 / Generating a Gaussian Distribution of Random Numbers

```
! Algorithm 13.1.
SET WINDOW 0, 1000, -5,5                    ! Determine the window limits.
FOR I = 1 to 1000 step .1                   ! Generate 10,000 numbers.
    LET x = 0                               ! Initialize: x = 0.
    FOR J = 1 to 12                         ! Loop to add 12 random
        LET x = x + rnd                     ! numbers.
    NEXT J
```

```
    LET x = x - 6                                   ! x is the new random
    PLOT I, x                                       ! number.
NEXT I                                              ! Graph number versus counter J.
END
```

PROGRAM 13.7.2 / Finding the Area of a Circle

```
! Algorithm 13.2 and Problem 13.2.
SET WINDOW -1.3*1.5,1.3*1.5,-1.3,1.3                ! The graphics are not
BOX LINES 0,1,0,1                                   ! absolutely necessary,
BOX CIRCLE -1,1,-1,1                                ! but they are nice.
FOR k = 1 to 25                                     ! Make 25 measurements.
    RANDOMIZE                                       ! Start new sequence of random
                                                      numbers with a new seed.
    FOR i = 1 to 1000                               ! Use 1000 points.
        LET x = rnd                                 ! Choose two random
        LET y = rnd                                 ! coordinates.
        PLOT x,y                                    ! Graph the point.
        IF x*x + y*y <= 1 then LET n = n + 1        ! n=n+1 if point is in circle.
    NEXT i                                          ! Continue loop.
    LET f = 4*n/1000                                ! Calculate area.
    PRINT 1000, f                                   ! Print results.
    LET n = 0                                       ! Loop for another
    LET A = A + f                                   ! measurement.
    LET AA = AA + f*f                               ! Find standard deviation.
NEXT k
LET AVG = A/(k-1)
PRINT AVG, sqr ((k-1)*AA-A^2)/(k-1)                 ! Print result of 25
END                                                 ! 1000-point measurements and
                                                      their standard deviation.
```

PROGRAM 13.7.3 / The Mass and Moment of Inertia of a Sphere

```
! Algorithm 13.3, Problem 13.3 and Problem 13.4.
LET M = 1000                                        ! 1000 points in first octant.
LET N = 25                                          ! Make 25 measurements.
LET dV = 1/M                                        ! An element of volume.
FOR k = 1 to N                                      ! Repeat 25 measurements.
    RANDOMIZE                                       ! Sow new random number seed.
    Let V = 0                                       ! Initialize V.
    LET IZ = 0                                      ! Initialize IZ
    FOR J = 1 to M                                  ! Repeat 1000 times.
        LET x = rnd                                 ! Random x.
        LET y = rnd                                 ! Random y.
        LET z =rnd                                  ! Random z.
```

```
        IF x*x + y*y + z*z <= 1 then                    ! Is the point in the sphere?
           LET V = V + dV                               ! Yes. Add volume and
           LET IZ = (x*x + y*y)*dV + IZ                 ! moment of inertia.
        END IF                                          ! No. Add nothing.
    NEXT j
    LET AV = AV + V                                     ! Form an average.
    LET VV = VV + V^2                                   ! Form moment of inertia
    LET I = I + IZ                                      ! average.
    LET II = II + IZ^2                                  ! Also sum squares.
NEXT k
LET AVGV = 8*AV/N
LET AVGI = 8*I/N
PRINT AVGV, 8*sqr (N*VV-AV^2)/N                         ! Print averages and their
PRINT AVGI, 8*sqr(N*II- I^2)/N                          ! standard deviations.
END
```

PROGRAM 13.7.4 / Generating Gaussian Noise: Finding the Mean and Standard Deviation

```
! Algorithm 13.4 and Problem 13.5.
RANDOMIZE                                               ! Plant a new seed.
DO
   CALL Noise (x)                                       ! Get a random number.
   LET s = s + x                                        ! Sum the random numbers.
   LET xx = xx + x^2                                    ! Sum their squares.
   LET k = k + 1                                        ! Count them.
   IF mod(k,1000) = 0 then                              ! Every 1000th number we will
      LET sd = sqr(k*xx - s^2)/k                        ! calculate the standard
                                                          deviation
      LET mean =x/k                                     ! and the mean.
      LET mean = round(mean,3)                          ! Round the mean.
      LET sd = round(sd,4)                              ! Round the standard deviation.
      PRINT k, mean, sd                                 ! Print them.
      RANDOMIZE                                         ! Plant a new seed.
   END IF                                               ! Otherwise we will do nothing.
LOOP until k = 10000                                    ! End of loop.
END
!......................
SUB Noise (x)                                           ! Generate noise with a
    LET x = 0                                           ! Gaussian distribution.
    FOR j = 1 to 12
        LET x = x + rnd
    NEXT j
    LET x= x - 6                                        ! Make mean zero.
END SUB
```

PROGRAM 13.7.5 / Generating and Graphing a Gaussian Distribution of Noise

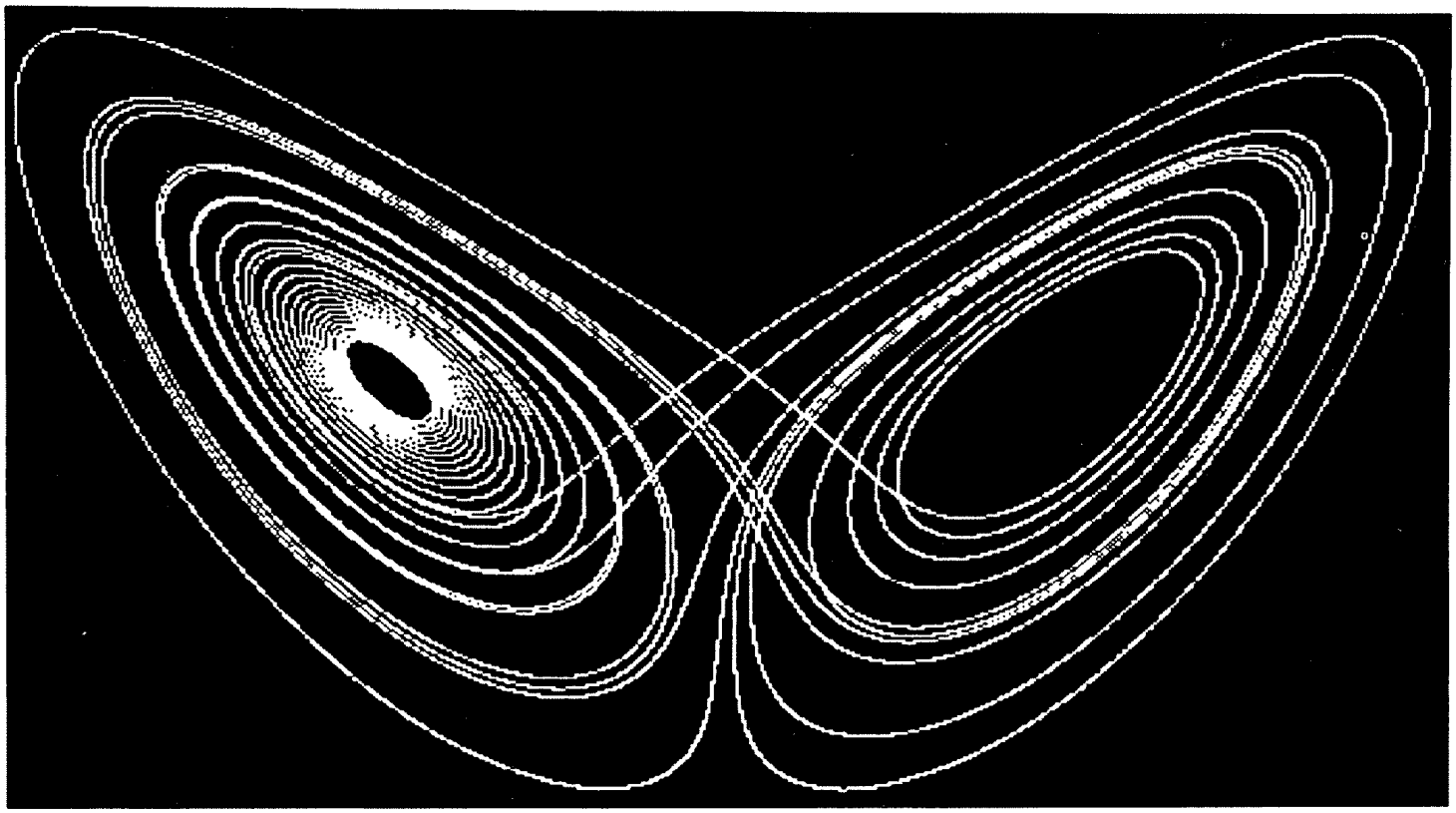
```
! Algorithm 13.5, Problems 13.7, 13.8, and 13.13.
DIM A(25)                                                  ! Set up an array, A(I).
SET WINDOW -5, 30, -300,3000                               ! Our coordinate system.
PLOT 0,-16; 22,-16                                         ! Plot the x-axis.
RANDOMIZE                                                  ! Plant a random number seed.
DO                                                         ! The main loop.
   CALL Noise (x)                                          ! Get a number.
   LET s = s + x                                           ! Add them together.
   LET xx = xx + x*x                                       ! Sum the squares.
   LET k = k + 1                                           ! Increment the loop counter.
   IF mod(k,1000) = 0 then                                 ! Every 1000 noise pulses
      LET sd = sqr(k*y - x*x)/k                            ! update the mean and
      LET mean =s/k                                        ! standard deviation.
      LET mean = round(mean,3)                             ! Round the results.
      LET sd = round(sd,3)
      BOX CLEAR 15,30,2500,2650                            ! Put them on the screen.
      PLOT TEXT, AT 15, 2550:str$(k)&"  mean = "&str$(mean)&" sd = "&str$(sd)
      RANDOMIZE                                            ! Plant another seed.
   ELSE
   END IF                                                  ! End of this calculation.
   IF x => 0 then                                          ! Group positive numbers here.
      LET x = x - 0.2                                      ! Find half of the elements
      IF x <= 0 then                                       ! in A(11).
         LET A(11) = A(11)+1
      ELSE                                                 ! Now do A(12) to A(22).
         LET I = 12
         DO
            LET x = x - 0.4                                ! Find in which interval the
            IF x<= 0 then                                  ! random number lies.
               LET A(I) = A(I) + 1
               EXIT DO                                     ! Once you have found it, quit
            ELSE                                           ! trying.
               LET I = i + 1                               ! Keep trying here.
            END IF
         LOOP until I >22                                  ! Now you're finished for
      END IF                                               ! sure.
   ELSE                                                    ! Group negative numbers here.
      LET x = x + 0.2                                      ! Find other half of the
      IF x>= 0 then                                        ! elements in A(11).
         LET A(11)=A(11) + 1
      ELSE                                                 ! Now do A(10) - A(1).
         LET I = 10
         DO
            LET x = x + .4
            IF x => 0 then                                 ! Find in which interval the
               LET A(I) = A(i) + 1                         ! random number lies.
               EXIT DO                                     ! This is in A(10) interval.
            ELSE                                           ! Or try one of the others
               LET I = i - 1                               ! A(9) to A(1).
            END IF
```

```
        LOOP until I =< 0
      END IF                                          ! Enough of this.
   END IF
FOR I = 1 to 21                                       ! Graph the array A(I).
       IF a(i) <> 0 then PLOT I, A(I)
   NEXT I
LOOP until A(11) > 2500                               ! Quit the main loop when
FOR I = 1 to 21 step .1                               ! A(11) has 2500 elements.
    LET x = -4.4+ i*.4
    LET y = A(11)*exp(-(x^2)/2)                       ! Graph a theoretical
    PLOT I,y;                                         ! Gaussian distribution.
NEXT i
END
SUB Noise (N)                                         ! Generate Gaussian noise
    LET N = 0                                         ! distribution.
    FOR j = 1 to 12
        LET N = N + rnd
    NEXT j
    LET N = N - 6                                     ! Make mean zero.
END SUB
```

PROGRAM 13.7.6 / Averaging Records to Reduce Noise

```
! Algorithm 13.6, Problem 13.10.
SET WINDOW -30,300,-2,2                               ! Open window to graph signal.
DIM A(500), B(500)                                    ! Set up memory for arrays.
FOR i = 1 to 500                                      ! A(I) and B(I).
    LET x = (i-250.5)/(3*pi)                          ! Generate the signal.
    LET A(i) = sin(x)/(x)                             ! Store it in array A(I).
    PLOT i/2, A(i)
NEXT i
SET WINDOW -30, 300, -4,4                             ! Open a new window.
LET k = 1                                             ! Repeat experiment k times.
DO                                                    ! Start main loop.
   CLEAR
   RANDOMIZE                                          ! Start with new seed.
   FOR I = 1 to 500
       CALL Noise (x)                                 ! Get a noise pulse.
       LET B(i) = B(i) + A(i) + x                     ! Add it to the record.
       PLOT i/2, B(i)/k                               ! Display the average.
   NEXT i
   LET sum = 0                                        ! Calculation of the
   FOR i = 1 to 500                                   ! standard deviation.
       LET Sum = sum + (a(i) - b(i)/k)^2
   NEXT I
   LET dev = round(sqr(sum/500),3)
   PLOT TEXT, AT 200,7:"N = "&str$(k)&" dev = "&str$(dev)   ! plot the results.
   If k = 25 then print "Hi! Time for a coke?"
PAUSE 1                                               ! Read answer.
```

```
   LET k = k + 1
LOOP
END
SUB Noise (x)                                          ! Subroutine to calculate
    LET x = 0                                          ! Gaussian distributed
    FOR j = 1 to 24                                    ! noise.
        LET x = x + rnd
    NEXT j
    LET x =3*(x - 12)                                  ! Make mean zero.
END SUB
```



Butterfly Diagram

Supplement / Graphics

S.1 Introduction

It is unfortunate but true that there are considerable differences in graphics capabilities from one programming language to another and from one version of BASIC to another. Worse, some versions of BASIC have different graphics capability for different computers. QuickBASIC on the Macintosh has different graphics instructions and capabilities than QuickBASIC on an IBM PC. It is fortunate that graphics are quite simple to do in any version of BASIC. In this supplement we will explore two major forms of graphics capabilities, giving you the tools to do black and white graphics on most computers.

There are two types of graphics systems with which we will deal. The first type may be called *coordinate-system graphics* and the second *pixel graphics*. With coordinate-system graphics, the screen of the video monitor or the page on the printer become a Cartesian coordinate system identical to the x-y coordinate systems used in mathematics, the sciences,

and engineering. *You* determine the scale just as you would when you draw a graph on a sheet of paper. In True BASIC, coordinate-system graphics make use of coordinates called *window-coordinates*. In QuickBASIC, these same coordinates are called *world coordinates*. We describe each of these systems in the next two sections. (At the time of this writing, QuickBASIC on the Macintosh does not support world coordinates.)

You will see that an important advantage of coordinate-system graphics is that it equips us with the ability do graphics on any scale. If we are working on the scale of the atom, our length scale would be on the order of 10^{-10} m; if we are working on the scale of the solar system, our length scale would be on the order of 10^{12} m. Another advantage of coordinate-system graphics is that larger is to the right on the x-axis and up on the y-axis, in contrast to pixel graphics in which larger on the y-axis is down. Furthermore, coordinate-system graphics allow the use of positive and negative numbers, whereas pixel graphics do not. Pixel graphics are less convenient; each point with Cartesian coordinates (x, y) must undergo a coordinate transformation to determine the corresponding pixel coordinate pair, (p_x, p_y).

S.2 Coordinate-System Graphics

S.2.1 Coordinate-System Graphics in True BASIC

We begin by describing coordinate-system graphics in True BASIC. With this system, the screen of the video monitor may be thought of as a piece of graph paper with no lines. Users define the scale of the graph that they need to present their data. Suppose that you wish to graph the sine function on the interval $[0, 4\pi]$. Recall that the range of the sine function is $[-1, 1]$. In True BASIC you would begin by setting the x and y scales with a

```
SET WINDOW 0, 4*pi, -1, 1
```

instruction. The graph of the sine function would use the entire screen. To stay away from the edges of the screen, it would be wiser to use the following instruction:

```
SET WINDOW -0.1*pi, 4.1*pi, -1.2, 1.2
```

which would keep the graph of the sine function neatly inside the window. In other words, make the window slightly larger than the graph.

This example should make it clear that the instruction

```
SET WINDOW xmin, xmax, ymin, ymax
```

informs the computer that the smallest x-coordinate you wish to graph is x_{min} and the largest is x_{max}. The quantities y_{min} and y_{max} have the same meaning but for the y-axis. As long as the function you wish to graph has a domain of $[x_{min}, x_{max}]$ and a range of $[y_{min}, y_{max}]$, its graph will appear on the screen. That is, x must satisfy the condition

$$x_{min} \leq x \leq x_{max}$$

and y must satisfy the condition

$$y_{min} \leq y \leq y_{max}.$$

It is important to realize that the coordinate system need not be symmetrical. Thus a problem might require the instruction

```
SET WINDOW 0, 3, -12, 48
```

to set the appropriate boundaries of the window. Finally, to plot a point in True BASIC a

```
PLOT X, Y
```

instruction is used. Example S.1 illustrates the use of both the SET WINDOW instruction and the PLOT X, Y instruction in making a graph of the sine function. Its output is shown in Figure S.1.

EXAMPLE S.1 / Graphing the Sine Function: True BASIC

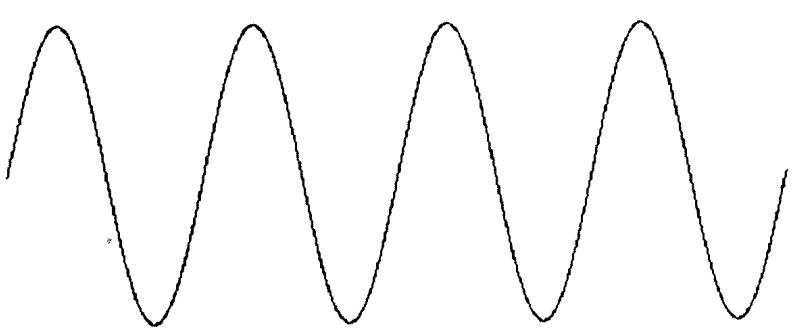
```
! Concepts Illustrated:
   ! Using the SET WINDOW instruction.
   ! Graphing a function in True BASIC.
SET WINDOW -4.1*pi, 4.1*pi, -2, 2          ! Set min and max values of x
                                             and y.
FOR X = -4*pi TO 4*PI STEP 0.01            ! Draw a sine function curve.
   LET y = sin(x)                          ! Define the function to be
                                             graphed.
   PLOT x,y                                ! Graph it one point at a time.
NEXT x                                     ! Loop until the graph is
                                             finished.
END
```

PROBLEM S.1 Graphing a Function in True BASIC • Make a graph of the function defined by

$$y = x^3 - x^2 + x$$

on the interval $[-4, 4]$. From this graph of this cubic function, find approximately where it is zero and find approximate values of the local maxima and minima.

FIGURE S.1 • A Graph of the Sine Function made with the Program in Example S.1.



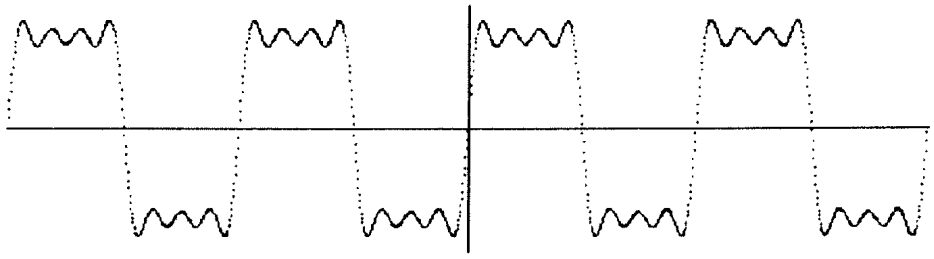


FIGURE S.2 • A Graph of the Function $y = \sin(x) + \sin(3x)/3 + \sin(5x)/5 + \sin(7x)/7$ on Coordinate Axes. This graph is the output of Example S.2.

In addition to the PLOT instruction, which is used to plot points, there are graphics instructions to draw lines. In True BASIC the instruction

```
PLOT x1, y1; x2, y2
```

draws a line from the point (x_1, y_1) to the point (x_2, y_2) on the coordinate system defined by the SET WINDOW instruction. The order of the points is of no consequence. Example S.2 illustrates the use of this instruction to put x and y axes on the screen. The output of this program is shown in Figure S.2. We find that almost all our graphics needs are met by the ability to plot points and lines.

EXAMPLE S.2 / Drawing Coordinate Axes in True BASIC

```
! Concepts Illustrated:
   ! Using the PLOT instruction to draw lines.
   ! Graphing more complex functions.
SET WINDOW -4.1*pi, 4.1*pi, -3, 3                          ! Set min and max values of x
                                                             and y.
PLOT 0, -2; 0, 2                                           ! Draw y-axis.
PLOT -4*pi, 0; 4*pi, 0                                     ! Draw x-axis.
FOR x = -4*pi to 4*pi step 0.02                            ! Make the domain -4π to 4π.
   LET y = sin(x) + sin(3*x)/3 + sin(5*x)/5 + sin(7*x)/7   ! The function to be graphed.
   PLOT x,y                                                ! Graph it one point at a time.
NEXT x                                                     ! Loop until the graph is
                                                             finished.
END
```

PROBLEM S.2 Drawing Coordinate Axes • Modify Problem S.1 to add coordinate axes.

S.2.2 Coordinate-System Graphics in QuickBASIC

In QuickBASIC a coordinate system is established with the instruction

```
WINDOW (xmin, ymin) - (xmax, ymax),
```

defining the diagonal limits of a box on the screen. To be graphed on the screen, the coordinates of a point must satisfy the following conditions:

$$x_{min} \le x \le x_{max}$$

and y must satisfy the condition

$$y_{min} \le y \le y_{max}.$$

To plot points, QuickBASIC uses a

```
PSET (x, y)
```

instruction. A QuickBASIC program to graph the sine function is given in Example S.3.

EXAMPLE S.3 / Graphing the Sine Function in QuickBASIC

```
'Concepts Illustrated:
    'Using the WINDOW instruction.
    'Graphing a function in QuickBASIC.
WINDOW (-13, -2) - (13, 2)                    'Set min and max values of x anc
                                                y.
FOR X = -4*pi to TO 4*PI STEP 0.01            'Draw a sine function curve.
    LET y = sin(x)                            'Define the function to be
                                                graphed.
    PSET (x,y)                                'Graph it one point at a time.
NEXT x                                        'Loop until the graph is
                                                finished.
END
```

QuickBASIC uses a

```
LINE (x1, y1) - (x2, y2)
```

instruction to graph a line from the point with coordinates (x_1, y_1) to the point with coordinates (x_2, y_2). The order of the points is of no consequence. Example S.4 illustrates the use of a LINE instruction to draw the coordinate axes.

EXAMPLE S.4 / Drawing Coordinate Axes in QuickBASIC

```
'Concepts Illustrated:
    'Using the LINE instruction.
    'Graphing a function in QuickBASIC.
WINDOW (-13, -2) - (13, 2)                    'Set min and max values of x and
                                                y.
LINE (-13, 0) - (13, 0)                       'Draw x-axis.
LINE (0, -1.5) - (0, 1.5)                     'Draw y-axis.
```

```
FOR X = -4*pi to TO 4*PI STEP 0.01          'Draw a sine function curve.
    LET y = sin(x) + cos(x)                 'Define the function to be
                                              graphed.
    PSET (x,y)                              'Graph it one point at a time.
NEXT x                                      'Loop until the graph is
                                              finished.
END
```

PROBLEM S.3 Graphing a Function in QuickBASIC • Use QuickBASIC to make a graph of the function

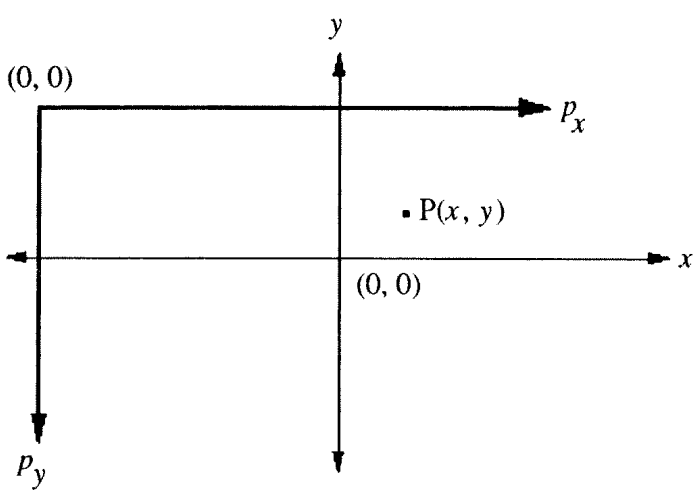
$$y = e^x - 1$$

for x in [–5, 5]. Draw the coordinate axes.

S.3 Pixel Graphics

The screen of your video monitor or the page of your printer is made up of **pixels**, a shortened version of "picture elements." For example, on Macintosh computers the screen measures 512 pixels horizontally and 322 pixels vertically, giving a total of 103,684 pixels, each of which may be made either black or white. The pixels are numbered starting from zero, so the horizontal pixel axis on a Macintosh is from 0 to 511 on the horizontal pixel axis, which we shall call the p_x-axis. The p_y-axis, in this case, extends from 0 to 321. On IBM PCs or compatibles, the number of pixels varies considerably, depending on the particular graphics system that is purchased. You must know the pixel specifications of your graphics system if you wish to do pixel graphics.

FIGURE S.3 • Pixel Coordinates and Cartesian Coordinates. The origin of the pixel coordinate system is in the upper-left-hand corner, while the Cartesian system has its origin at the center.



The origin of the pixel coordinate system is in the upper-left-hand corner of the screen. The p_x-axis is positive to the right and the p_y-axis is positive downward. No negative numbers are allowed. In some versions of BASIC, Applesoft for example, selecting a pixel outside of the range of allowed pixels produces an error. A pixel-coordinate system and a Cartesian coordinate system are superimposed in Figure S.3. We will show how to convert the Cartesian coordinates of a point, (x, y), into pixel coordinates, (p_x, p_y), which the computer can plot.

We assume that there is a linear transformation between the pixel coordinates, (p_x, p_y) of point P and the Cartesian coordinates, (x, y) of point P, as follows:

$$p_x = k_1 x + k_2,$$

and

$$p_y = k_3 y + k_4.$$

Requiring that the transformation be linear means that the variables p_x, p_y, x, and y, do not have exponents other than one. This ensures that straight lines in Cartesian coordinates graph into straight lines in pixel coordinates, circles graph into circles, and in general, the graph of a function in pixel coordinates is similar to the graph of the same function in Cartesian coordinates. Our task is to find the constants k_1, k_2, k_3, and k_4 that will allow us to calculate p_x and p_y given x and y.

Assume that the x-coordinates of the graph or trajectory we wish to plot lie in the interval $[x_{min}, x_{max}]$ and its y-coordinates lie in the interval $[y_{min}, y_{max}]$. Assume that the horizontal pixels are numbered 0 to a, and assume that the vertical pixels are numbered 0 to b. Thus, if there are 512 horizontal pixels, then $a = 511$, and if there are 322 vertical pixels, then $b = 321$. Here are the demands we place on the k constants:

- When $x = x_{min}$, $p_x = 0$.
- When $x = x_{max}$, $p_x = a$.
- When $y = y_{min}$, $p_y = b$.
- When $y = y_{max}$, $p_y = 0$.

Substitution of these four demands into the transformation equations produces four equations that may be solved for the four unknowns, k_1, k_2, k_3, and k_4.

We find:

$$k_1 = a\frac{1}{x_{max} - x_{min}},$$

$$k_2 = a\frac{x_{min}}{x_{min} - x_{max}},$$

$$k_3 = b\frac{1}{y_{min} - y_{max}},$$

and

$$k_4 = b\frac{y_{max}}{y_{max} - y_{min}}.$$

In a computer program the calculations for the k's are done once, *but each point requires the linear transformation given above*. Example S.5 illustrates how to make a graph of the hyperbolic cosine function, the shape taken by a cable hanging under its own weight. It is sometimes wise to make a and b somewhat smaller than the window dimensions to keep the graph clearly visible on the screen. Thus you might choose $a = 500$ and $b = 300$, but give (0,0) - (512, 322) as the window dimensions. Example S.6 illustrates how to draw coordinate axes using pixel coordinates.

PROBLEM S.4 Transforming Cartesian Coordinates into Pixel Coordinates • Substitute our demands into the pixel equations and verify the expressions for k_1, k_2, k_3, and k_4 given above.

EXAMPLE S.5 / Using Pixel Graphics in QuickBASIC to Graph the Hyperbolic Cosine Function

```
'Concepts Illustrated:
    'Using the WINDOW instruction.
    'Graphing a function using pixel coordinates.
LET a = 500: LET b = 310                                  'Screen is a x b in pixels.
WINDOW 1,"graphics",(0,0)-(511,321)                       'Window named 'graphics' is
                                                            established
LET XMIN = -5: LET xmax = 5: LET ymin = 0: LET ymax = 75  'for pixel coordinates.
LET k1 = a/(xmax - xmin)                                  'Calculate k1.
LET k2 = a*xmin/(xmin - xmax)                             'Calculate k2.
LET k3 = b/(ymin - ymax)                                  'Calculate k3.
LET k4 = b*ymax/(ymax - ymin)                             'Calculate k4.
FOR X = -4 TO 4 STEP 0.05                                 'Vary x over the domain [-5, 5].
    LET y = 0.5*(exp(x) + exp(-x))                        'Calculate the hyperbolic
                                                            cosine.
    PSET (x*k1+k2, y*k3+k4)                               'Graph it one point at a time.
NEXT x                                                    'Loop until the graph is
                                                            finished.
END
```

Some versions of BASIC on older computers stop executing a program and print an error message if the points you try to plot are not in the intervals $[0, a]$ for the p_x-coordinate or $[0, b]$ for the p_y-coordinate. If that is the case, you will need to provide IF... THEN... instructions

to branch around the PLOT or PSET instruction. Here is a simple instruction sequence that will take care of this problem:

```
100 IF px < 0 OR px > a THEN 120
110 IF py => 0 and py =< b THEN PLOT px, py
120 ......
```

where we have used line numbers because these restrictions are usually found in early versions of BASIC.

EXAMPLE S.6 / Graphing (sinx)/x and Coordinate Axes with Pixel Graphics: QuickBASIC

```
LET a = 500: LET b = 300                                    'Define pixel boundaries.
WINDOW 1,"graphics",(0,0)-(511,321)                         'Define the window.
LET xmin = -12: LET xmax = 12: LET ymin = -.25: LET ymax = 1.2
LET k1 = a/(xmax - xmin)                                    'Find the transformation
                                                              equations.
LET k2 = a*xmin/(xmin - xmax)
LET k3 = b/(ymin - ymax)                                    'Transform axes endpoints into
LET k4 = b*ymax/(ymax - ymin)                               'pixel coordinates.
LET px1 = k1*xmin + k2: LET py1 = k3*0 + k4                 'Prepare to draw x-axis.
LET px2=k1*xmax + k2
LINE (px1, py1) - (px2, py1)                                'Draw x-axis.
LET px = 0*k1 + k2:LET py1 = ymin*k3 + k4: LET py2 = ymax*k3 + k4
LINE (px, py1)-(px,py2)                                     'Draw y axis.
FOR x = -10  TO 10 STEP .005                                'Now plot the function.
    LET y = ((SIN(x)/x*COS(5*x))^2)
    LET px = k1*x + k2                                      'Transform axes endpoints into
    LET py = k3*y + k4                                      'pixel coordinates.
    PSET (px, py)                                           'Plot using pixel coordinates.
NEXT x
END
```

PROBLEM S.5 Using Pixel Coordinates to Graph the Hyperbolic Sine Function

• Assume that the screen is 256 x 256 pixels. Write a program to graph the hyperbolic sine function

$$y = \frac{1}{2}(e^x - e^{-x}).$$

Choose the interval [–4, 4] for the domain. With this domain, the range of this function is approximately [–30, 30]. Try adding axes to your graph. Refer to Example S.6 to see how axes may be drawn in pixel graphics.

S.4 Curves Graphed Parametrically

In physics you encounter vectors and they are usually treated in terms of their x and y components. Thus it is commonplace to see graphs of $y(t)$ versus $x(t)$ as well as the graphs we have described so far, which are graphs of functions, $y = f(x)$. The trajectory of a particle is a case in point. To graph the path of a projectile, we graph $x(t)$ and $y(t)$ where

$$x(t) = x_o + v_{x0}t,$$

and

$$y(t) = y_0 + v_{y0}t + \frac{1}{2}a_y t^2.$$

Curves described by equations for x and y in terms of a third variable (parameter), t, are said to be described parametrically.

We will give an example of how to graph curves described parametrically. Because of its interesting properties, we choose to graph a cycloid. A point on a wheel rolling without slipping traces out a cycloid. The parametric form of its equation follows:

$$x(t) = r(1 - \sin t),$$

and

$$y(t) = r(1 - \cos t).$$

Example S.7 illustrates the computer program, and Figure S.4 is the output of this program.

EXAMPLE S.7 / Graphing Parametric Equations: True BASIC

```
! Concept Illustrated: Graphing parametric equations.
SET WINDOW -15, 15, -3, 3                          ! Set min and max values of x
                                                     and y.
PLOT 0, 0; 0, 2                                    ! Draw y-axis.
PLOT -4*pi, 0; 4*pi, 0                             ! Draw x-axis.
FOR t = -4*pi to 4*pi step 0.04                    ! Make the domain -4π to 4π.
    LET x = t - sin(t)                             ! Determine x parametrically.
    LET y = 1 - cos(t)                             ! Determine y parametrically.
    PLOT x,y                                       ! Graph it one point at a time.
NEXT t                                             ! Loop until the graph is
                                                     finished.
END
```

PROBLEM S.6 Graphing a Four-Cusped Hypocycloid • Graph the curve described parametrically by

$$x(t) = r\cos^3 t,$$

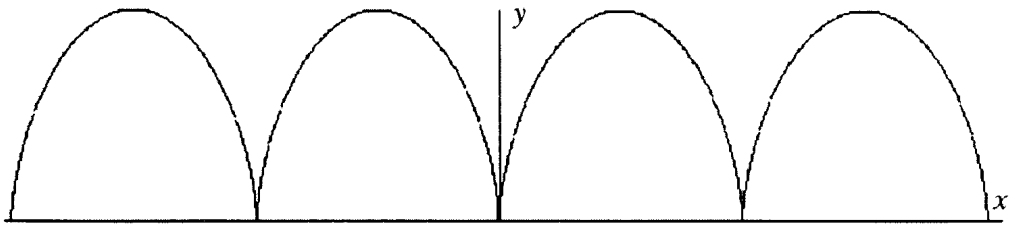


FIGURE S.4 • Graphing the Parametric Equations of a Cycloid.

and

$$y(t) = r\sin^3 t.$$

S.5 Aspect Ratio

In many instances it is important to have identical x and y scales on the video monitor, when drawing a circle for example. The x and y scales on a video monitor are not necessarily the same, and the ratio of the x-scale to the y-scale is the aspect ratio. To find the aspect ratio, write a program that draws a circle on the screen. If the figure is indeed a circle, there is nothing more to do. If the figure is an ellipse, the ratio of the length of the horizontal axis to the vertical axis of the ellipse is the aspect ratio. To correct the scales to obtain a circle, multiply the window size in the x-direction by the aspect ratio, using the set window instruction. For example, on the Macintosh, the instruction

```
SET WINDOW -1, 1, -1, 1
```

produces a window in which a circle appears as an ellipse. This is corrected by using a

```
SET WINDOW -1.5, 1.5, -1,1
```

instruction.

Suppose that our task is to graph the curve described parametrically with the equations

$$x(t) = \sin(mt),$$

and

$$y(t) = \sin(nt),$$

where m and n are integers. We wish to have identical x and y scales. Example S.8 illustrates how to graph these equations. The curve so obtained is called a *Lissajous* figure and appears in Figure S.5. Example S.9 is another illustration of how to correct for the aspect ratio.

EXAMPLE S.8 / Graphing Lissajous Figures

```
SET WINDOW -1.4*1.5, 1.4*1.5,-1.4,1.4          ! Open window.
PLOT -1.1*1.5 ,0; 1.1*5.,0                     ! Plot x-axis.
PLOT 0, -1.1; 0,1.1                            ! Plot y-axis.
LET t = 0                                      ! Initial value of t.
LET r = 1                                      ! Simplicity.
DO until t > 2*pi                              ! Vary t
    LET x = r*cos(3*t)                         ! This determines x.
    LET y = r*sin(5*t)                         ! This determines y.
    LET t = t + .005
    PLOT x,y;                                  ! Plot instruction.
LOOP                                           ! End of loop.
END
```

Many polar graphs may be plotted using parametric equations. Example S.9 illustrates how to make a graph of the polar equation $r = \sin(5\theta)$. We make use of the coordinate transformation equations,

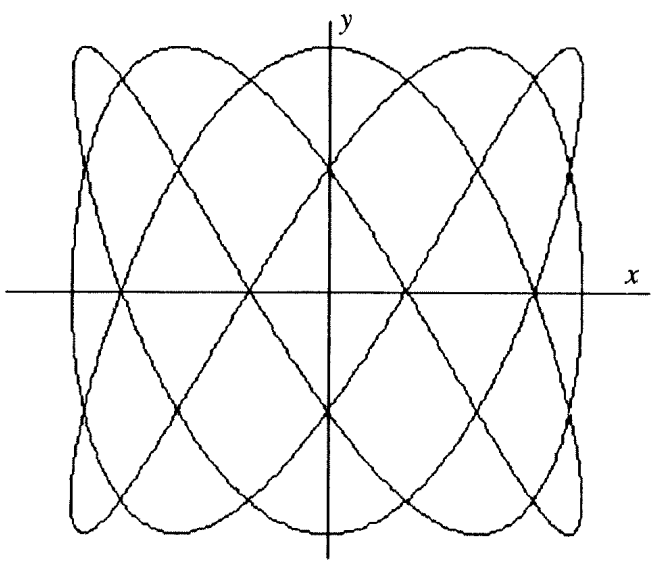
$$x = r\cos\theta$$

and

$$y = r\sin\theta,$$

to put the polar equation in parametric form.

FIGURE S.5 • Output of Example S.8.



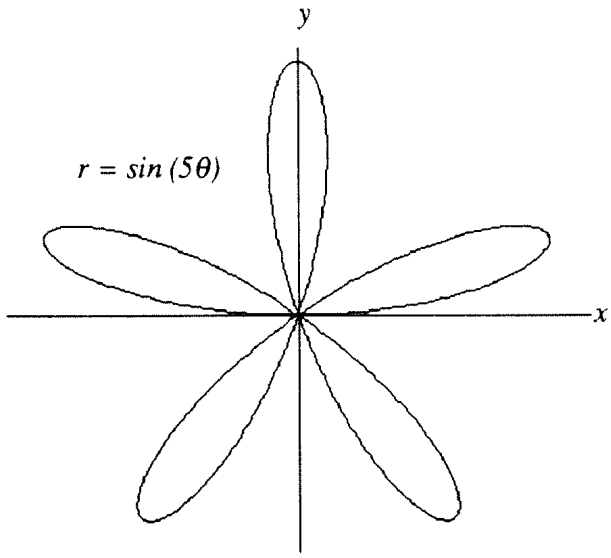


FIGURE S.6 • Graph of the Polar Equation $r = \sin(5\theta)$. Polar equations are usually handled by putting them in parametric form.

PROBLEM S.7 Graphing the Polar Curve $r = \sin(5\theta)$ • Make a graph of the polar curve $r = \sin(5\theta)$. To do this, you will need to transform polar coordinates into two parametric equations with the equations

$$x = r\cos\theta = \sin(5\theta)\cos\theta,$$

and

$$y = r\sin\theta = \sin(5\theta)\sin\theta.$$

Example S.9 illustrates the programming necessary to complete the task, and Figure S.6 shows the output of this program.

EXAMPLE S.9 / Graphing in Polar Coordinates

```
SET WINDOW -1.4, 1.4,-1.4,1.4
PLOT -1.1,0; 1.1,0                    ! Plot x-axis.
PLOT 0, -1.1; 0,1.1                   ! Plot y-axis.
LET t = 0                             ! Initial value of t.
DO until t > pi                       ! Vary t
    LET x = sin(5*t)*cos(t)           ! This determines x.
    LET y = sin(5*t)*sin(t)           ! This determines y.
    LET t = t + .005
    PLOT x,y;                         ! Plot instruction.
LOOP                                  ! End of loop.
END
```

PROBLEM S.7 Graphing Polar Equations • Make a graph of the polar equation $r = 1 - \cos\theta$.

S.6 Answers to the Programming Problems in this Supplement

PROBLEM S.1 AND S.2 / Graphing a Function in True BASIC

```
SET WINDOW -5.1, 5.1, -125, 125                    ! Set up coordinate system.
PLOT -5,0;5,0                                      ! Plot x-axis.
PLOT 0, -100; 0, 100                               ! Plot y-axis.
LET x = -5                                         ! Starting value of x.
DO until x > 5
   LET y = x^3 - x^2 + x                           ! Calculate y.
   PLOT x, y;                                      ! Plot points.
   LET x = x + .1                                  ! Increment x.
LOOP
END
```

PROBLEM S.3 / Graphing a Function in QuickBASIC

```
WINDOW (-6, -160) - (160, 2)                       'Define coordinate system.
LINE (-5, 0) - (5, 0)                              'Draw x-axis.
LINE (0, -150) - (150, 1.5)                        'Draw y-axis.
LET x = - 5                                        'Starting value of x.
WHILE x <= 5
    LET y = exp(x) - 1                             'Define the function
    PSET (x,y)                                     'Graph it.
    LET X = X + 1                                  'Increment x.
WEND                                               'End of loop.
END
```

PROBLEM S.5 / Graphing the Hyperbolic Sine Function with Pixels: QuickBASIC

```
LET a = 255: LET b = 255                           'Screen is a x b in pixels.
WINDOW 1,"graphics",(0,0)-(a,b)                    'Window named 'graphics' is
                                                     established
LET XMIN = -4:LET xmax = 4:LET ymin = -30:LET ymax = 30   'for pixel coordinates.
LET k1 = a/(xmax - xmin)                           'Calculate k1.
LET k2 = a*xmin/(xmin - xmax)                      'Calculate k2.
LET k3 = b/(ymin - ymax)                           'Calculate k3.
LET k4 = b*ymax/(ymax - ymin)                      'Calculate k4.
LET px1 = k1*xmin + k2: LET py1 = k3*0 + k4        'Prepare to draw x-axis.
LET px2=k1*xmax + k2
LINE (px1, py1) - (px2, py1)                       'Draw x-axis.
LET px = 0*k1 + k2:LET py1 = ymin*k3 + k4: LET py2 = ymax*k3 + k4
LINE (px, py1)-(px,py2)                            'Draw y axis.
```

```
FOR X = -4 TO 4 STEP 0.05                    'Vary x over the domain [-5, 5].
    LET y = 0.5*(exp(x) - exp(-x))           'Calculate the hyperbolic
                                               cosine.
    LET px = k1*x + k2                       'Do the coordinate
                                             transformation for x.
    LET py = k3*y + k4                       'Do the coordinate
                                               transformation for y.
    PSET (px, py)                            'Graph it one point at a time.
NEXT x                                       'Loop until the graph is
                                               finished.
END
```

PROBLEM S.6 / Graphing a Four-Cusped Hypocycloid

```
SET WINDOW -1.2, 1.2,-1.2,1.2
PLOT -1,0; 1,0                               ! Plot x-axis.
PLOT 0, -1; 0,1                              ! Plot y-axis.
LET t = 0                                    ! Initial value of t.
LET r = 1                                    ! Simplicity.
DO until t > 2*pi                            ! Vary t
    LET x = r*(cos(t)^3)                     ! This determines x.
    LET y = r*(sin(t)^3)                     ! This determines y.
    LET t = t + .01
    PLOT x,y                                 ! Plot instruction.
LOOP                                         ! End of loop.
END
```

PROBLEM S.7 / Graphing in Polar Coordinates

```
SET WINDOW -2.1, 2.1,-1.4,1.4
PLOT -2,0; 2,0                               ! Plot x-axis.
PLOT 0, -1.1; 0,1.1                          ! Plot y-axis.
LET t = 0                                    ! Initial value of t.
DO until t > pi                              ! Vary t
    LET x = (1-cos(t))*cos(t)                ! This determines x.
    LET y = (1-cos(t))*sin(t)                ! This determines y.
    LET t = t + .005
    PLOT x,y;                                ! Plot instruction.
LOOP                                         ! End of loop.
END
```

S.7 Library ComPhys

In this section we list the graphics subroutines used throughout this book. These routines are to be placed in a library called ComPhys. Many of the programs in the book call these routines. One of the first instructions which should be put in a program that calls these routines is the

```
LIBRARY "ComPhys"
```

instruction. This instruction loads the routines into the True BASIC environment, where they may be called by the program. There is no particular need to study these routines, simply use them. With the exception of SUBROUTINE LABEL, they will run on any machine that is supported by True BASIC SUBROUTINE LABEL uses mouse commands, and cannot be used if there is no mouse.

LISTING S.1 / True BASIC Graphics Routines, COMPHYS, Used in this Book

```
EXTERNAL!..........
SUB OPEN (xmin,xmax,ymin,ymax,wxmin,wxmax,wymin,wymax)      ! Opens graphics window.
    PRINT "Please enter the limits of the x-axis, xmin and xmax" ! x boundaries.
    PRINT "respectively, separated with commas. Then press RETURN."
    INPUT xmin, xmax
    PRINT "Please enter the range limits, ymin and ymax"    ! y boundaries.
    PRINT "respectively, separated with commas. Then press RETURN."
    INPUT ymin, ymax
    LET xint = 0.02*(xmax - xmin)                           ! Make window size in x
                                                              direction a few% larger than
    LET wxmin = xmin - xint                                 ! coordinate system.
    LET wxmax = xmax + xint
    LET inty = 0.02*(ymax - ymin)                           ! Make window size in y
                                                              direction
    LET wymin = ymin - inty                                 ! a few% larger on each end.
    LET wymax = ymax + inty
    SET WINDOW wxmin, wxmax, wymin, wymax                   ! Open the graphics window.
    CLEAR                                                   ! Clear the screen.
    CALL AXES(xmin, xmax, ymin, ymax,xaxis, yaxis)          ! Subroutine to draw axes.
    CALL TICK_Marks(xmin, xmax, ymin, ymax,xaxis,yaxis,dx,dy) ! Routine to add tick marks.
    CALL HASH_MARKS (xmin, xmax, ymin, ymax,dx,dy)          ! Subroutine to add hash marks.
END SUB
!..........
SUB AXES (xmin, xmax, ymin, ymax,xaxis, yaxis)              ! Subroutine to draw axes.
    IF xmin*xmax < 0 then                                   ! Determine where to put y-axis
       LET yaxis = 0                                        ! if omain goes from - to +.
       PLOT xmin,ymin;xmin,ymax                             ! Draw left wall.
       PLOT xmax,ymin;xmax,ymax                             ! Draw right wall.
       PLOT yaxis,ymin-.04*ymin;yaxis, ymax -.04*ymax       ! Draw y-axis.
    ELSE                                                    ! If domain has one sign,
       LET yaxis = xmin                                     ! put x axis at y = 0,
       PLOT xmax,ymin;xmax,ymax                             ! Draw right side of box.
       PLOT yaxis,ymin; yaxis,ymax                          ! Draw left side of box.
```

```
    END IF                                                  ! otherwise put it at xmin.
    IF ymin*ymax<0 then                                     ! Determine where to put x-axis
       LET xaxis = 0
       PLOT xmin,ymin;xmax,ymin                             ! Draw bottom wall.
       PLOT xmin,ymax;xmax,ymax                             ! Draw top wall.
       PLOT xmin -.04*xmin,xaxis;xmax-.04*xmax,xaxis        ! Plot the x axis.
    ELSE                                                    ! If range goes from - to +
       LET xaxis = ymin                                     ! put x-axis at y = 0.
       PLOT xmin ,ymax;xmax,ymax                            ! Draw top wall
       PLOT xmin, xaxis; xmax, xaxis                        ! Draw x-axis.
    END IF
END SUB
!..........
SUB TICK_MARKS (xmin, xmax, ymin, ymax,xaxis,yaxis,dx,dy)
    ! Draw nine tick marks on each axis.
    LET dx=(xmax - xmin)/10
    LET dy = (ymax-ymin)/10
    LET y = xaxis
    FOR x = xmin +dx to xmax-dx/10 step dx                  ! Draw tick marks on x axis
        PLOT x,ymin;x,ymin + dy/8                           ! bottom wall
        PLOT x,ymax;x,ymax-dy/8                             ! top wall
        IF ymax*ymin <0 then
           PLOT x, y - dy/10; x, y + dy/10                  ! at intervals dx apart.
        ELSE
           PLOT x,0;x,dy/10                                 ! x axis.
        END IF
    NEXT x
    LET x = yaxis
    FOR y = ymin+ dy to ymax-dy/10 step dy                  ! Draw tick marks on y axis
        PLOT xmax,y;xmax -dx/10,y                           ! right wall
        PLOT xmin,y;xmin+dx/10,y                            ! left wall
        IF xmax*xmin <0 then
           PLOT x-dx/10,y;x+dx/10,y                         ! y axis
        ELSE
           PLOT 0,y;dx/10,y                                 ! y axis
        END IF                                              ! at intervals dx apart
    NEXT y
END SUB
!..........
SUB HASH_MARKS (xmin, xmax, ymin, ymax, dx, dy)
    LET x = 0                                               ! Put hash marks on graph.
    FOR y = ymin to ymax+dy/100 step dy/2
        FOR x = xmin to xmax+dx/100 step dx/2               ! Marks spacing is same as tick
                                                              spacing.
            PLOT x,y                                        ! Hash mark is one point.
        NEXT x
    NEXT y
END SUB
! ..........
SUB SAVE (wxmin, wxmax, wymin, wymax, Keep$)                ! Subroutine to save a graphics
                                                              window
    BOX KEEP wxmin, wxmax, wymin, wymax in Keep$            ! as a string (Keep$) in memory.
```

```
END SUB
! ....................
SUB SHOW (wxmin, wymin, Keep$)                          ! Subroutine to restore string
                                                          Keep$
    BOX SHOW Keep$ at wxmin, wymin                      ! to the graphics window.
END SUB
!...........
SUB LABEL (wxmin, wxmax, wymin, wymax)                  ! Start of subroutine.
    CALL SAVE (wxmin, wxmax, wymin, wymax, keep$)       ! Save the screen momentarily
    DO                                                  ! in a file named keep$.
       PRINT "Type any text you would like to add to the graph.  Otherwise"
       PRINT "type no."
       INPUT Words$                                     ! Input the text.
       IF Words$ = "no" then                            ! Type "no" for no more text.
       CLEAR
       CALL SHOW (wxmin, wymin, keep$)                  ! Put graph back on screen
       EXIT DO                                          ! and get out of the loop.
    ELSE
       CLEAR                                            ! Clear the screen.
       CALL SHOW (wxmin, wymin, keep$)                  ! Show it.
       GET POINT x,y                                    ! Find out from mouse where
       PLOT TEXT, AT x,y: Words$                        ! to plot the text. Plot it.
       CALL SAVE (wxmin, wxmax, wymin, wymax , keep$)   ! Save screen.
    END IF
 LOOP                                                   ! Get more text.
END SUB                                                 ! End of the subroutine.
!...........
SUB PLOT_DATA_A (wxmin,wxmax,wymin, wymax, t,v)         ! Plots crosses at experimental
                                                          (t, v) points.
 LET deltx =0.75*((wxmax - wxmin)/50)                   ! Determine size of "+".
 LET delty = (wymax - wymin)/50                         ! 1.5 is aspect ratio
                                                          correction.
 PLOT t,v-delty;t,v+delty                               ! Graph a "+" symbol at the
                                                          point.
 PLOT  t-deltx,v; t+deltx,v
END SUB
SUB PLOT_DATA_B(wxmin,wxmax,wymin,wymax, t,v)           ! Plots boxes at (t, s) points.
 LET deltx = 0.75*((wxmax - wxmin)/80)                  ! Determine size of box.
                                                          Approximately
 LET delty =(wymax - wymin)/80                          ! 1/80 of the window dimensions.
 PLOT t-deltx,v-delty;t-deltx,v+delty;t+deltx,v+delty;t+deltx,v-delty;t-deltx,v-delty ! Draw box.
END SUB
!................
SUB OPENB (xmin,xmax,ymin,ymax,wxmin,wxmax,wymin,wymax)  ! Corrects for aspect ratio.
 PRINT "Please enter the range limits, ymin and ymax"
 PRINT "respectively, separated with commas. Then press RETURN."
 INPUT ymin, ymax
 LET xmin = 1.5*ymin
 LET xmax = 1.5*ymax
 LET inty = 0.03*(ymax - ymin)                          ! Make window size in y
                                                          direction
 LET wymin = ymin - inty                                ! few% larger on each end.
```

```
 LET wymax = ymax + inty
 LET wxmin = 1.5*wymin
 LET wxmax = 1.5*wymax
 SET WINDOW wxmin,wxmax, wymin, wymax                    ! Open a graphics window.
 CLEAR                                                   ! Clear the screen.
 CALL AXESB(xmin, xmax, ymin, ymax,xaxis, yaxis)         ! Subroutine to draw axes.
 CALL TICK_MarksB (xmin, xmax, ymin, ymax,xaxis,yaxis,dx,dy)
END SUB
SUB AXESB (xmin, xmax, ymin, ymax,xaxis, yaxis)          ! Subroutine to draw axes.
 IF xmin*xmax < 0 then                                   ! Determine where to put y-axis.
    LET yaxis = 0                                        ! Domain goes from - to +.
    PLOT xmin,ymin;xmin,ymax                             ! Draw left wall.
    PLOT xmax,ymin;xmax,ymax                             ! Draw right wall.
 ELSE                                                    ! If domain has one sign,
    LET yaxis = xmin                                     ! put x axis at y = 0,
    PLOT xmax,ymin;xmax,ymax                             ! Draw right side of box.
    PLOT yaxis,ymin; yaxis,ymax                          ! Draw left side of box.
 END IF                                                  ! otherwise put it at xmin.
 IF ymin*ymax<0 then                                     ! Determine where to put x-axis
    LET xaxis = 0                                        ! the x-axis.
    PLOT xmin,ymin;xmax,ymin                             ! Draw bottom wall.
    PLOT xmin,ymax;xmax,ymax                             ! Draw top wall.

 ELSE                                                    ! If range goes from - to +
    LET xaxis = ymin                                     ! put x-axis at y = 0.
    PLOT xmin ,ymax;xmax,ymax                            ! Draw top wall
    PLOT xmin, xaxis; xmax, xaxis                        ! Draw x-axis.
 END IF
END SUB
!..........
SUB TICK_MARKSB (xmin, xmax, ymin, ymax,xaxis,yaxis,dx,dy)
 ! Draw nine tick marks on each axis.
 LET dx=(xmax - xmin)/10
 LET dy = (ymax-ymin)/10
 LET y = xaxis
 FOR x = xmin +dx to xmax-dx/10 step dx                  ! Draw tick marks on x axis
     PLOT x,ymin;x,ymin + dy/10                          ! bottom wall
     PLOT x,ymax;x,ymax-dy/10                            ! top wall
 NEXT x
 LET x = yaxis
 FOR y = ymin+ dy to ymax-dy/10 step dy                  ! Draw tick marks on y axis
     PLOT xmax,y;xmax -dx/10,y                           ! right wall
     PLOT xmin,y;xmin+dx/10,y                            ! left wall
 NEXT y
END SUB
```

Index

A
Activity, 197
Air resistance, 2, 3, 32, 34, 40, 49, 50, 68–71, 74, 85, 106, 193
Alpha particle, 164–169, 171–173
Altitude, 76
Analog, 139
Analytic solution, 6, 25–26, 31–33, 38, 68, 128, 134
Angular acceleration, 32
Angular displacement, 32
Angular momentum, 105
Angular velocity, 32, 51, 55
Apogee, 79, 85, 104
Approximation, 4, 5, 16
Area, 238
Area of a circle, 254
Aspect ratio, 79, 269
Attractor, 36, 141, 207, 208, 216, 223, 225–227
Average, 246, 252
Averaging, 257

B
Baseball, 2, 8, 15, 69–71, 73, 85, 101
Beta particle, 193, 194
Binary stars, 186
Binary system, 186
Bohr model, 83
Bohr, Niels, 185
Bonding region, 184–185
Butterfly diagram, 259

C
Calculus, 24
Capacitor, 124, 128–131, 137, 140, 142
Cartesian coordinates, 264
Centrifugal, 76
Chaotic attractor, 208, 225
Chaotic dynamical system, 205–206
Charge, 128
Charging, 124
Circular orbits, 80
Coefficient of friction, 53–54
COMPHYS, 20, 21, 41, 274
Computer model, 142–143
Conditional equation, 11
Conservation of energy (COE), 13, 33, 36, 84, 92, 95, 96, 98, 101, 104, 128–136
Conservation of momentum, 105, 171
Coordinate axes, 262
Coordinate-system graphics, 259–262
Coriolis, 76
Coulomb's Law, 83, 109, 110, 164, 167, 178, 180, 183
Covalent bond, 178, 183–185
Critically damped, 141
Cross product, 105
Current, 124, 125, 128

D
Damped, 105, 215, 217
Damped oscillations, 35, 38, 141
Damped pendulum, 35
Damping, 34, 35, 37, 39, 209, 212, 216, 223
Damping coefficient, 35, 40, 92, 95
Damping force, 92–93
Daughter, 194, 197–199, 203
Decay, 195
Definition of a derivative, 1, 3, 15, 18, 24, 31
Density of air, 74
Deterministic chaos, 206
Deterministic system, 207
Difference equation, 227
Difference quotient, 1, 4, 24
Differential equation, 33, 34, 130, 206, 209–210, 222, 227
Discharge, 128
Discharging, 124
Displacement, 24
Dissipative, 216
Domain, 14
Drag, 48, 51, 101
Drag coefficient, 2, 8, 49, 50, 51, 52, 56, 59, 66, 69, 71, 74, 101, 133

Driven oscillations, 39, 105, 217
Driven and damped pendulum, 206
Driving, 212
Driving force, 105, 106, 223
Duffing hard–spring oscillator, 229
Duffing oscillator, 206, 209, 212, 213, 214, 215, 216, 217, 220, 225, 232
Dynamical system, 207

E

E-field, 112, 113, 114, 115, 116, 117, 118, 119, 120, 165, 167
E-vector, 120
Earth, 81, 82
Elastic potential energy, 93, 97
Electric compasses, 113
Electric field, 109–115, 117, 119, 120, 164–165
Electric field lines, 119
Electrical potential energy, 117, 129, 131
Electron, 81, 83, 117, 178–185, 188
Electron density, 189
Electron volts, 183
Electrostatic potential energy, 183
Ellipsis, 81
Elliptical orbit, 81, 104–105
Energy, 91, 92, 94, 101, 107, 135, 142
Envelope, 39
Equals sign, 11
Equation of motion, 209–210, 212, 216, 217
Equipotential curve, 116, 117, 118, 119
Equipotential line segments, 118
Equipotential lines, 109, 118, 121
Equipotential surface, 109, 117
Euler's method, 34, 38, 65, 125, 126, 228, 130
Exponential function, 198

F

Falling marble, 14
Falling object, 1, 3, 8, 12, 15, 16, 17, 18, 133
Families of curves, 128
Feynman-Newton method, 38, 44, 47, 65, 68, 91, 93, 104, 138, 164, 178, 210
Field lines, 112
First-order differential equation, 33–34
Fixed point, 207, 215–216
Fly by, 187
Formula, 12
Force law, 171
Fractal, 208, 225–227
Function notation, 3, 11, 12
Fundamental theorem of calculus (FTC), 99

G

Gamma rays, 193
Gaussian distribution, 236, 237, 249, 253, 256
Gaussian noise, 244, 246, 249, 255
Geiger, H., 164, 169, 171
Grain of sand, 15
Graphics, 259
Graphics subroutines, 20
Graphing Lissajous figures, 270
Graphing parametric equations, 268
Graphing polar curves, 271
Gravitational potential energy, 101, 102, 103

H

H_2, 178, 184
H_2 ion, 177–183, 185, 188
H_2^+ ion, 185,
Half-life, 199–201
Half-step method, 38
Halley's Comet, 186
Helium atom, 186
Hitting angle, 73
Hooke's Law, 27
Horsepower, 45, 46, 48, 51, 52, 53, 55, 56, 59
Hydrogen atom, 83, 84
Hydrogen molecule, 181
Hyperbolic orbit, 81
Hyperbolic orbits, 105

I

Identity, 11
Impact parameter, 165, 169–170
Inductor, 131–134, 137, 140, 142
Initial condition, 6, 11, 28, 33
Instantaneous velocity, 16
Integral, 24, 237, 243
Integral definition, 19
Internal thermal energy, 95, 101, 131

Interplanetary space probe, 187
Ionized hydrogen molecule, 178

K
Kinetic energy, 49–51, 92–96, 101, 103–104, 164, 171, 183–184, 214
Kirchhoff's Loop Rule, 124, 125, 132, 137, 138, 139, 143

L
Limit, 28
Limit cycle, 207, 218
Limit definitions, 28
Linear transformation, 265, 266
Log-log graph, 252
Logarithmic graphs, 201
Logistic equation, 227
Logistic function, 227
Lord Rutherford, 164
Lorenz equations, 228
LR circuit, 131, 132, 133, 134, 135, 137, 140
LRC circuit, 123, 137–143

M
Magnetic field, 134, 142
Magnetic potential energy, 134, 135
Marsden, E., 164, 169–171
Mascon, 186
Maximum range, 68
Mean, 243, 244, 245, 255
Mechanical energy, 46, 85, 92, 93, 95, 99, 100, 101, 103
Missile, 171
Moment of inertia, 32, 235, 240, 242, 243, 253
Moment of inertia of a sphere, 254
Momentum, 91
Monte Carlo techniques, 235, 240, 241, 242

N
Natural frequency of vibration, 39
Newton's Second Law, 1–3, 5, 9, 18, 20, 25, 27, 32–34, 39, 38, 43, 65, 67, 69, 74, 77, 80, 83, 91, 94, 97, 98, 138, 139, 164, 167, 178, 206, 209–211
Newton's Third Law, 166
Noise, 243, 249–252, 257
Noise distribution, 247
Non-linear differential equation, 209
Non-linear dynamical system, 207, 209
Nonconservative forces, 98
Notation, 3, 10, 12, 15
Notation conventions, 13
Nuclei, 197, 198
Nucleus, 170, 171, 195, 197
Numerical analysis, 19, 26, 34
Numerical integration, 19, 99
Numerical solution, 6–8, 10, 13, 17–19, 25, 26, 28, 31–33, 37–38, 66, 68, 126, 128, 132–134, 138, 210, 211
Numerical techniques, 32

O
Object-spring system, 26, 28–32, 35–36, 38, 94, 106, 109, 205
Ohm's Law, 124
Orbit, 207
Orbital motion, 65, 103, 104, 105
Overdamped, 141

P
Parabolic, 81, 105
Parent, 194, 197–200
Pendulum, 205, 208–209, 212–222, 224, 229, 232
Perigee, 79, 85, 104
Period, 219, 220
Period doubling, 220, 221
Period of the pendulum, 32
Periodic driving force, 39
Periodic orbit, 207
Periodic potential, 214
Phase curve, 214
Phase flow, 214
Phase portrait, 23, 211, 214–218, 222, 231
Phase portrait, 214
Phase space, 35–36, 39, 205, 207, 214, 216, 218–222, 225
Phase-space trajectory, 36
Ping-pong ball, 15, 71, 72
Pixel coordinates, 264, 265, 266
Pixel graphics, 259–260, 264, 266, 267
Pixels, 264

Planet, 79
Poincaré section, 222–225, 233
Polar coordinates, 271
Potential, 117, 118, 121, 122, 124
Potential difference, 116, 129, 131
Potential energy, 92, 95, 101, 102, 104, 116, 117, 164, 183, 184, 212, 214
Potential well, 214
Power, 46, 47, 48, 49, 51, 52, 53, 54, 55, 134, 135
Predictability, 208
Probability, 194, 195, 197, 200, 236
Probe, 187
Projectile, 19, 26, 40, 65, 68, 69, 70, 72, 73, 74, 75, 76, 77, 78, 79, 80, 84, 85, 94
Projectile motion, 68, 69, 70, 101
Proton, 81, 83, 84, 117, 178, 179, 180, 181, 182, 183, 185

Q

Quadratic potential, 213
Quantum mechanics, 171, 181, 185
Quartic potentials, 213

R

Radioactive, 164, 193–194, 201
Radioactive daughter, 200
radioactive decay, 194, 196, 197, 198, 199, 201
radioactive nuclei, 196, 200
Radioactive parent, 196, 200, 202, 203
Radioactive series, 199
Radioactivity, 193
Raindrops, 15
Random, 206
Random number, 193–197, 200, 206, 235, 236, 237, 239, 240, 244, 245, 246, 249, 253
Range, 14
RC circuit, 124–128, 131, 132, 142
Replacement, 12
Resistance, 124
Resistor, 124, 128–136, 142
Resonance frequency, 106, 140
RND, 195, 235, 239
Rolling friction, 48, 49, 50, 51
Rotational kinetic energy, 53
Round-off errors, 13
Runge-Kutta, 210, 211
Running sum, 10, 19, 29, 106, 130
Rutherford, 170, 171
Rutherford scattering, 163, 164–165, 167, 169

S

Satellite, 65, 77, 80–81, 84–85, 91, 94, 101–107, 186
Scintillations, 170, 171
Second-order differential equation, 33
Seed, 200, 236
Self-inductance, 123, 132
Self-similarity, 226
SET WINDOW, 260, 261, 262
Shock absorbers, 35
Simple pendulum, 32, 35, 37
Sirius A, 186
Sirius B, 186
Softball, 74
Space probe, 187
Space-time trajectory, 36
Spinning magnet, 229
Spreadsheet, 8, 10, 12–13, 29, 30, 99, 235, 239
Spreadsheet solution, 10
Spring, 27
Spring constant, 92, 94
Stable attractor, 36
Stable daughter, 196, 197, 202
Standard deviation, 238–239, 242–246, 250, 252, 255
Strange attractor, 205
Styrofoam ball, 18
Subscript notation, 10–11, 16
Surface-friction damping, 39

T

Tacoma Narrows Bridge, 40
Target nucleus, 170
Tennis ball, 15
Terminal value, 127
Terminal velocity, 8, 9, 14, 15, 20, 48, 57, 59
Thermal energy, 49, 50, 51, 98, 96, 99, 100, 129, 131, 134, 135, 136
Three-body problem, 177, 178

Thomson, J. J., 165
Torque, 32, 208
Total energy, 103, 104, 105, 106
Tractive resistance, 49, 50, 56, 57
Trajectory in phase space, 35
Transfer orbit, 187
Transmutation, 196
Truncation errors, 13
Two-well potential, 214

U
Umbrella, 18
Underdamped, 141
Uniform noise, 244, 253
Universal Law of Gravitation, 74
Unpredictable, 206, 208
Uranium, 194

V
Velocity, 24
Velocity space, 182–183
Venus, 187
Vibrating system, 26–27
Viscous damping, 208
Volume of a sphere, 242
Volume of an ellipsoid, 242

W
Window, 79, 262
Window-coordinates, 260
Work, 101, 102, 106, 135, 164
Work-kinetic energy theorem (WKET), 43, 45–46, 49–51, 95, 97–98
World coordinates, 260