INTRODUCTION TO COMPUTATIONAL PHYSICS

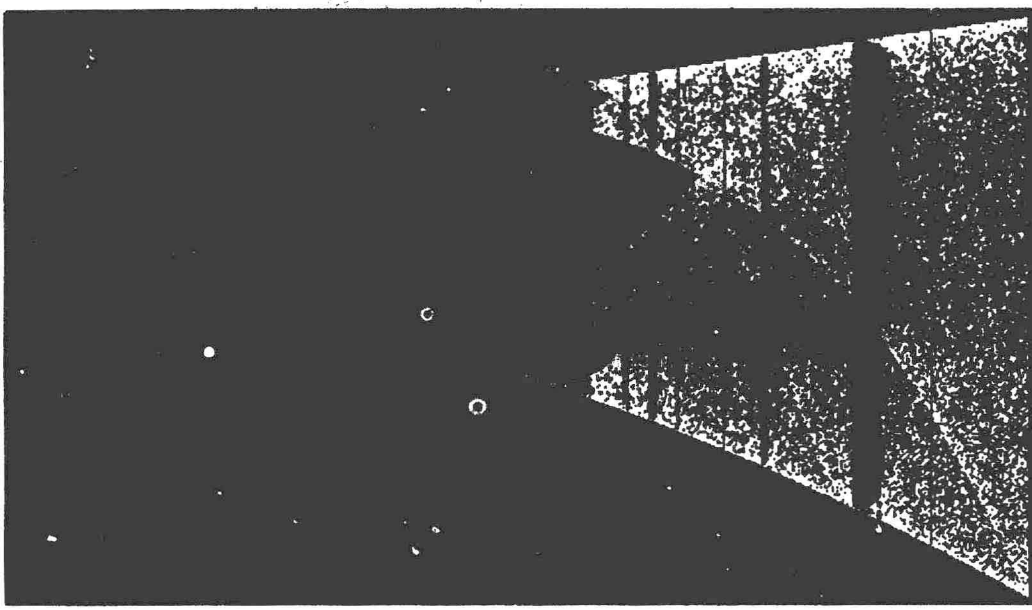
INTRODUCTION TO COMPUTATIONAL PHYSICS

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

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

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-thebullet 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 ≡ 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 (≅),

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

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.

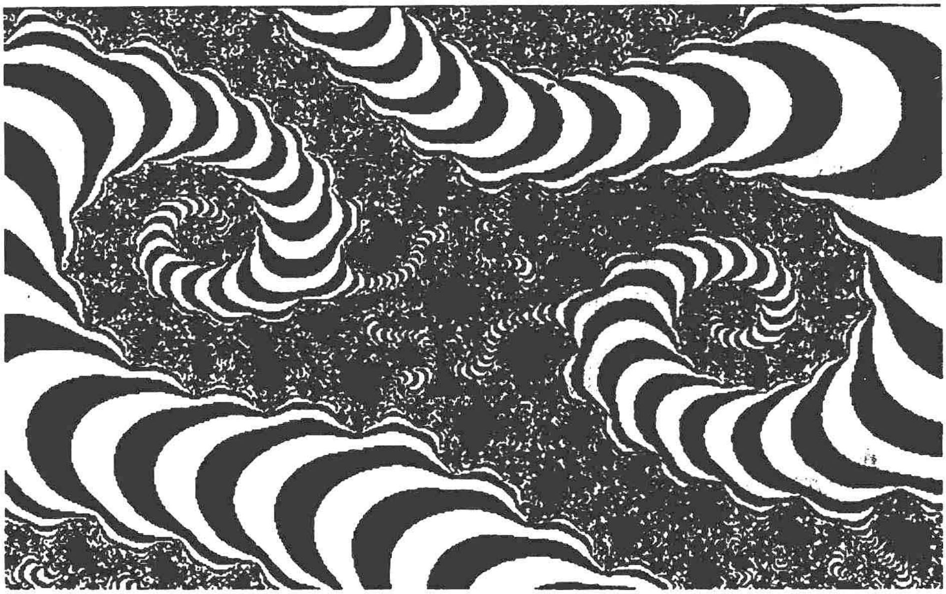
I take some responsibility for his problem, perhaps most of it. I should have asked many more conceptual questions, probing the class members for their understanding of physics. On exams, I should not have provided such large rewards for doing the same kind of problems that the students had done for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

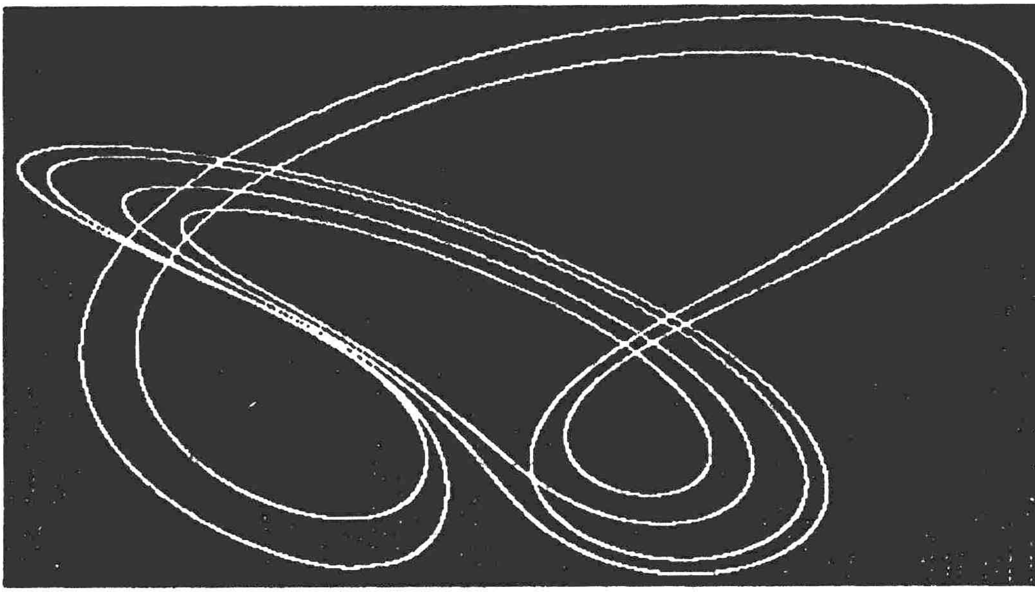
Understanding physics is a shared responsibility and a shared joy. What a professor or a textbook may do is engage you. I hope that I have assumed my share of the responsibility by engaging you and giving you an opportunity to think physics. The rest of the responsibility is yours.

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


Fractal Geometry

Contents

Chapter 1 The Motion of a Falling Object



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