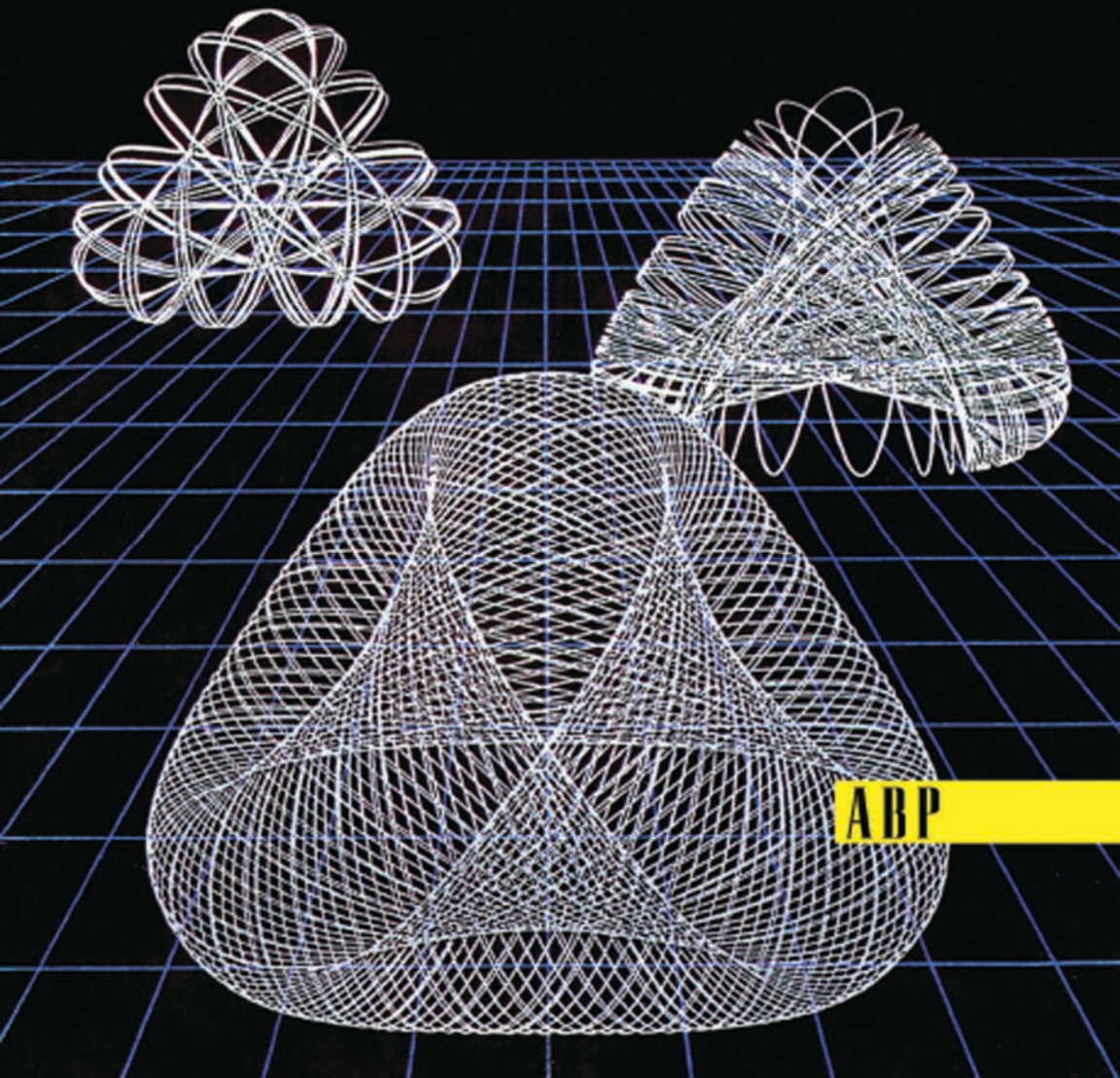
STEVEN E. KOONIN / DAWN C. MEREDITH

COMPUTATIONAL PHYSICS

FORTRAN VERSION



ABP

COMPUTATIONAL PHYSICS

Fortran Version

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

COMPUTATIONAL PHYSICS
Fortran Version

Steven E. Koonin
Professor of Theoretical Physics
California Institute of Technology

Dawn C. Meredith
Assistant Professor of Physics
University of New Hampshire

Advanced Book Program

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business

First published 1990 by Westview Press

Published 2018 by CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

CRC Press is an imprint of the Taylor & Francis Group, an informa business

Visit the Taylor & Francis Web site at
http://www.taylorandfrancis.com

and the CRC Press Web site at
http://www.crcpress.com

This book was typeset by Elizabeth K. Wood using the TEX text-processing system running on a Digital Equipment Corporation VAXStation 2000 computer.

Library of Congress Cataloging-in-Publication Data

Koonin, Steven E.
Computational physics (FORTRAN version) / Steven E. Koonin, Dawn Meredith.
p. cm.
Includes index.
1. Mathematical physics—Data processing. 2. Physics—Computer programs.
3. FORTRAN (computer program language) 4. Numerical analysis. I. Meredith, Dawn.
II. Title.
QC20.7.E4K66 1990 530. 1'5'02855133—dc19 90-129

ISBN 13: 978-0-201-38623-3 (pbk)
ISBN 13: 978-0-201-12779-9 (hbk)

Preface

Computation is an integral part of modern science and the ability to exploit effectively the power offered by computers is therefore essential to a working physicist. The proper application of a computer to modeling physical systems is far more than blind "number crunching," and the successful computational physicist draws on a balanced mix of analytically soluble examples, physical intuition, and numerical work to solve problems that are otherwise intractable.

Unfortunately, the ability "to compute" is seldom cultivated by the standard university-level physics curriculum, as it requires an integration of three disciplines (physics, numerical analysis, and computer programming) covered in disjoint courses. Few physics students finish their undergraduate education knowing how to compute; those that do usually learn a limited set of techniques in the course of independent work, such as a research project or a senior thesis.

The material in this book is aimed at refining computational skills in advanced undergraduate or beginning graduate students by providing direct experience in using a computer to model physical systems. Its scope includes the minimum set of numerical techniques needed to "do physics" on a computer. Each of these is developed in the text, often heuristically, and is then applied to solve non-trivial problems in classical, quantum, and statistical physics. These latter have been chosen to enrich or extend the standard undergraduate physics curriculum, and so have considerable intrinsic interest, quite independent of the computational principles they illustrate.

This book should not be thought of as setting out a rigid or definitive curriculum. I have restricted its scope to calculations that satisfy simultaneously the criteria of illustrating a widely applicable numerical technique, of being tractable on a microcomputer, and of having some particular physics interest. Several important numerical techniques have therefore been omitted, spline interpolation and the Fast Fourier Transform among them. *Computational Physics* is perhaps best thought of as establishing an environment offering opportunities for further exploration. There are many possible extensions and embellishments of the material presented; using one's imagination along these lines is one of the more rewarding parts of working through the book.

Computational Physics is primarily a physics text. For maximum benefit, the student should have taken, or be taking, undergraduate courses in classical mechanics, quantum mechanics, statistical mechanics, and advanced calculus or the mathematical methods of physics. This is *not* a text on numerical analysis, as there has been no attempt at rigor or completeness in any of the expositions of numerical techniques. However, a prior course in that subject is probably not essential; the discussions of numerical techniques should be accessible to a student with the physics background outlined above, perhaps with some reference to any one of the excellent texts on numerical analysis (for example, [Ac70], [Bu81], or [Sh84]). This is also *not* a text on computer programming. Although I have tried to follow the principles of good programming throughout (see Appendix B), there has been no attempt to teach programming *per se*. Indeed, techniques for organizing and writing code are somewhat peripheral to the main goals of the book. Some familiarity with programming, at least to the extent of a one-semester introductory course in any of the standard high-level languages (BASIC, FORTRAN, PASCAL, C), is therefore essential.

The choice of language invariably invokes strong feelings among scientists who use computers. Any language is, after all, only a means of expressing the concepts underlying a program. The contents of this book are therefore relevant no matter what language one works in. However, *some* language had to be chosen to implement the programs, and I have selected the Microsoft dialect of BASIC standard on the IBM PC/XT/AT computers for this purpose. The BASIC language has many well-known deficiencies, foremost among them being a lack of local subroutine variables and an awkwardness in expressing structured code. Nevertheless, I believe that these are more than balanced by the simplicity of the language and the widespread fluency in it, BASIC's almost universal availability on the microcomputers most likely to be used with this book, the existence of both BASIC interpreters convenient for writing and debugging programs and of compilers for producing rapidly executing finished programs, and the powerful graphics and I/O statements in this language. I expect that readers familiar with some other high-level language can learn enough BASIC "on the fly" to be able to use this book. A synopsis of the language is contained in Appendix A to help in this regard, and further information can be found in readily available manuals. The reader may, of course, elect to write the programs suggested in the text in any convenient language.

This book arose out of the Advanced Computational Physics Laboratory taught to third- and fourth-year undergraduate Physics majors

at Caltech during the Winter and Spring of 1984. The content and presentation have benefitted greatly from the many inspired suggestions of M.-C. Chu, V. Pönisch, R. Williams, and D. Meredith. Mrs. Meredith was also of great assistance in producing the final form of the manuscript and programs. I also wish to thank my wife, Laurie, for her extraordinary patience, understanding, and support during my two-year involvement in this project.

Steven E. Koonin
Pasadena
May, 1985

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Preface to the FORTRAN Edition

At the request of the readers of the BASIC edition of *Computational Physics* we offer this FORTRAN version. Although we stand by our original choice of BASIC for the reasons cited in the preface to the BASIC edition it is clear that many of our readers strongly prefer FORTRAN, and so we gladly oblige. The text of the book is essentially unchanged, but all of the codes have been translated into standard FORTRAN-77 and will run (with some modification) on a variety of machines. Although the programs will run significantly faster on mainframe computers than on PC's, we have not increased the scope or complexity of the calculations. Results, therefore, are still produced in "real time", and the codes remain suitable for interactive use.

Another development since the BASIC edition is the publication of an excellent new text on numerical analysis (*Numerical Recipes* [Pr86]) which provides detailed discussions and code for state-of-the-art algorithms. We highly recommend it as a companion to this text.

The FORTRAN versions of the code were written with the help of T. Berke (who designed the menu), G. Buzzell, and J. Farley. We have also profited from the suggestions of many colleagues who have generously tested the new codes or pointed out errors in the BASIC edition. We gratefully acknowledge a grant from the University of New Hampshire DISCovery Computer Aided Instruction Program which provided the initial impetus for the translation. Lastly, our thanks go to E. Wood for typesetting the text in TEX.

Steven E. Koonin
Pasadena, CA
August, 1989

Dawn C. Meredith
Durham, NH

How to Use This Book

This book is organized into chapters, each containing a text section, an example, and a project. Each text section is a brief discussion of one or several related numerical techniques, often illustrated with simple mathematical examples. Throughout the text are a number of exercises, in which the student's understanding of the material is solidified or extended by an analytical derivation or through the writing and running of a simple program. These exercises are indicated by the symbol ■.

Also located throughout the text are tables of numerical errors. Note that the values listed in these tables were taken from runs using BASIC code on an IBM-PC. When the machine precision dominates the error, your values obtained with FORTRAN code may differ.

The example and project in each chapter are applications of the numerical techniques to particular physical problems. Each includes a brief exposition of the physics, followed by a discussion of how the numerical techniques are to be applied. The examples and projects differ only in that the student is expected to use (and perhaps modify) the program that is given for the former in Appendix B, while the book provides guidance in writing programs to treat the latter through a series of steps, also indicated by the symbol ■. However, programs for the projects have also been included in Appendix C; these can serve as models for the student's own program or as a means of investigating the physics without having to write a major program "from scratch". A number of suggested studies accompany each example and project; these guide the student in exploiting the programs and understanding the physical principles and numerical techniques involved.

The programs for both the examples and projects are available either over the Internet network or on diskettes. Appendix E describes how to obtain files over the network; alternatively, there is an order form in the back of the book for IBM-PC or Macintosh formatted diskettes. The codes are suitable for running on any computer that has a FORTRAN-77 standard compiler. (The IBM BASIC versions of the code are also available over the network.) Detailed instructions for revising and running the codes are given in Appendix A.

A "laboratory" format has proved to be one effective mode of presenting this material in a university setting. Students are quite able to

work through the text on their own, with the instructor being available for consultation and to monitor progress through brief personal interviews on each chapter. Three chapters in ten weeks (60 hours) of instruction has proved to be a reasonable pace, with students typically writing two of the projects during this time, and using the "canned" codes to work through the physics of the remaining project and the examples. The eight chapters in this book should therefore be more than sufficient for a one-semester course. Alternatively, this book can be used to provide supplementary material for the usual courses in classical, quantum, and statistical mechanics. Many of the examples and projects are vivid illustrations of basic concepts in these subjects and are therefore suitable for classroom demonstrations or independent study.

Contents

The problem with computers is that they only give answers
—attributed to P. Picasso

Chapter 1

Basic Mathematical Operations

Three numerical operations — differentiation, quadrature, and the finding of roots — are central to most computer modeling of physical systems. Suppose that we have the ability to calculate the value of a function, $f(x)$, at any value of the independent variable x. In differentiation, we seek one of the derivatives of f at a given value of x. Quadrature, roughly the inverse of differentiation, requires us to calculate the definite integral of f between two specified limits (we reserve the term "integration" for the process of solving ordinary differential equations, as discussed in Chapter 2), while in root finding we seek the values of x (there may be several) at which f vanishes.

If f is known analytically, it is almost always possible, with enough fortitude, to derive explicit formulas for the derivatives of f, and it is often possible to do so for its definite integral as well. However, it is often the case that an analytical method cannot be used, even though we can evaluate $f(x)$ itself. This might be either because some very complicated numerical procedure is required to evaluate f and we have no suitable analytical formula upon which to apply the rules of differentiation and quadrature, or, even worse, because the way we can generate f provides us with its values at only a set of discrete abscissae. In these situations, we must employ approximate formulas expressing the derivatives and integral in terms of the values of f we can compute. Moreover, the roots of all but the simplest functions cannot be found analytically, and numerical methods are therefore essential.

This chapter deals with the computer realization of these three basic operations. The central technique is to approximate f by a simple function (such as first- or second-degree polynomial) upon which these

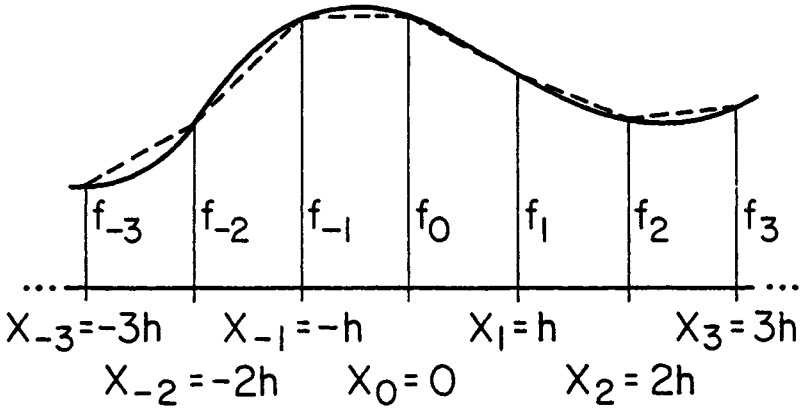


Figure 1.1 Values of f on an equally-spaced lattice. Dashed lines show the linear interpolation.

operations can be performed easily. We will derive only the simplest and most commonly used formulas; fuller treatments can be found in many textbooks on numerical analysis.

1.1 Numerical differentiation

Let us suppose that we are interested in the derivative at $x = 0$, $f'(0)$. (The formulas we will derive can be generalized simply to arbitrary x by translation.) Let us also suppose that we know f on an equally-spaced lattice of x values,

$$f_n = f(x_n)\,; \quad x_n = nh \; (n = 0, \pm 1, \pm 2, \ldots)\,,$$

and that our goal is to compute an approximate value of $f'(0)$ in terms of the f_n (see Figure 1.1).

We begin by using a Taylor series to expand f in the neighborhood of $x = 0$:

$$f(x) = f_0 + xf' + \frac{x^2}{2!}f'' + \frac{x^3}{3!}f''' + \ldots\,, \tag{1.1}$$

where all derivatives are evaluated at $x = 0$. It is then simple to verify that

$$f_{\pm 1} \equiv f(x = \pm h) = f_0 \pm hf' + \frac{h^2}{2}f'' \pm \frac{h^3}{6}f''' + \mathcal{O}(h^4)\,, \tag{1.2a}$$

$$f_{\pm 2} \equiv f(x = \pm 2h) = f_0 \pm 2hf' + 2h^2 f'' \pm \frac{4h^3}{3}f''' + \mathcal{O}(h^4)\,, \tag{1.2b}$$

where $\mathcal{O}(h^4)$ means terms of order h^4 or higher. To estimate the size of such terms, we can assume that f and its derivatives are all of the same order of magnitude, as is the case for many functions of physical relevance.

Upon subtracting f_{-1} from f_1 as given by (1.2a), we find, after a slight rearrangement,

$$f' = \frac{f_1 - f_{-1}}{2h} - \frac{h^2}{6} f''' + \mathcal{O}(h^4) . \tag{1.3a}$$

The term involving f''' vanishes as h becomes small and is the dominant error associated with the finite difference approximation that retains only the first term:

$$f' \approx \frac{f_1 - f_{-1}}{2h} . \tag{1.3b}$$

This "3-point" formula would be exact if f were a second-degree polynomial in the 3-point interval $[-h, +h]$, because the third- and all higher-order derivatives would then vanish. Hence, the essence of Eq. (1.3b) is the assumption that a quadratic polynomial interpolation of f through the three points $x = \pm h$, 0 is valid.

Equation (1.3b) is a very natural result, reminiscent of the formulas used to define the derivative in elementary calculus. The error term (of order h^2) can, in principle, be made as small as is desired by using smaller and smaller values of h. Note also that the symmetric difference about $x = 0$ is used, as it is more accurate (by one order in h) than the forward or backward difference formulas:

$$f' \approx \frac{f_1 - f_0}{h} + \mathcal{O}(h) ; \tag{1.4a}$$

$$f' \approx \frac{f_0 - f_{-1}}{h} + \mathcal{O}(h) . \tag{1.4b}$$

These "2-point" formulas are based on the assumption that f is well approximated by a linear function over the intervals between $x = 0$ and $x = \pm h$.

As a concrete example, consider evaluating $f'(x = 1)$ when $f(x) = \sin x$. The exact answer is, of course, $\cos 1 = 0.540302$. The following FORTRAN program evaluates Eq. (1.3b) in this case for the value of h input:

```
C chap1a.for
      X=1.
      EXACT=COS(X)
```

```
10      PRINT *, 'ENTER VALUE OF H (.LE. 0 TO STOP)'
        READ *,  H
        IF (H .LE. 0) STOP
        FPRIME=(SIN(X+H)-SIN(X-H))/(2*H)
        DIFF=EXACT-FPRIME
        PRINT 20,H,DIFF
20      FORMAT (' H=',E15.8,5X,'ERROR=',E15.8)
        GOTO 10
        END
```

(If you are a beginner in FORTRAN, note the way the value of H is requested from the keyboard, the fact that the code will stop if a non-positive value of H is entered, the natural way in which variable names are chosen and the mathematical formula (1.3b) is transcribed using the SIN function in the sixth line, the way in which the number of significant digits is specified when the result is to be output to the screen in line 20, and the jump in program control at the end of the program.)

Results generated with this program, as well as with similar ones evaluating the forward and backward difference formulas Eqs. (1.4a,b), are shown in Table 1.1. (All of the tables of errors presented in the text were generated from BASIC programs; the numbers may vary from those obtained from FORTRAN code, especially when numerical roundoff dominates.) Note that the result improves as we decrease h, but only up to a point, after which it becomes worse. This is because arithmetic in the computer is performed with only a limited precision (5–6 decimal digits for a single precision BASIC variable), so that when the difference in the numerator of the approximations is formed, it is subject to large "round-off" errors if h is small and f_1 and f_{-1} differ very little. For example, if $h = 10^{-6}$, then

$$f_1 = \sin(1.000001) = 0.841472\ ;\ \ f_{-1} = \sin(0.999999) = 0.841470\ ,$$

so that $f_1 - f_{-1} = 0.000002$ to six significant digits. When substituted into (1.3b) we find $f' \approx 1.000000$, a very poor result. However, if we do the arithmetic with 10 significant digits, then

$$f_1 = 0.8414715251\ ;\ \ f_{-1} = 0.8414704445\ ,$$

which gives a respectable $f' \approx 0.540300$ in Eq. (1.3b). In this sense, numerical differentiation is an intrinsically unstable process (no well-defined limit as $h \to 0$), and so must be carried out with caution.

Table 1.1 Error in evaluating $d \sin x/dx|_{x=1} = 0.540302$

h	Symmetric 3-point Eq. (1.3b)	Forward 2-point Eq. (1.4a)	Backward 2-point Eq. (1.4b)	Symmetric 5-point Eq. (1.5)
0.50000	0.022233	0.228254	−0.183789	0.001092
0.20000	0.003595	0.087461	−0.080272	0.000028
0.10000	0.000899	0.042938	−0.041139	0.000001
0.05000	0.000225	0.021258	−0.020808	0.000000
0.02000	0.000037	0.008453	−0.008380	0.000001
0.01000	0.000010	0.004224	−0.004204	0.000002
0.00500	0.000010	0.002108	−0.002088	0.000006
0.00200	−0.000014	0.000820	−0.000848	−0.000017
0.00100	−0.000014	0.000403	−0.000431	−0.000019
0.00050	0.000105	0.000403	−0.000193	0.000115
0.00020	−0.000163	−0.000014	−0.000312	−0.000188
0.00010	−0.000312	−0.000312	−0.000312	−0.000411
0.00005	0.000284	0.001476	−0.000908	0.000681
0.00002	0.000880	0.000880	0.000880	0.000873
0.00001	0.000880	0.003860	−0.002100	0.000880

It is possible to improve on the 3-point formula (1.3b) by relating f' to lattice points further removed from $x = 0$. For example, using Eqs. (1.2), it is easy to show that the "5-point" formula

$$f' \approx \frac{1}{12h}[f_{-2} - 8f_{-1} + 8f_1 - f_2] + \mathcal{O}(h^4) \tag{1.5}$$

cancels all derivatives in the Taylor series through fourth order. Computing the derivative in this way assumes that f is well-approximated by a fourth-degree polynomial over the 5-point interval $[-2h, 2h]$. Although requiring more computation, this approximation is considerably more accurate, as can be seen from Table 1.1. In fact, an accuracy comparable to Eq. (1.3b) is obtained with a step some 10 times larger. This can be an important consideration when many values of f must be stored in the computer, as the greater accuracy allows a sparser tabulation and so saves storage space. However, because (1.5) requires more mathematical operations than does (1.3b) and there is considerable cancellation among the various terms (they have both positive and negative coefficients), precision problems show up at a larger value of h.

Formulas for higher derivatives can be constructed by taking appro-

Table 1.2 4- and 5-point difference formulas for derivatives

	4-point	5-point
hf'	$\pm\frac{1}{6}(-2f_{\mp1} - 3f_0 + 6f_{\pm1} - f_{\pm2})$	$\frac{1}{12}(f_{-2} - 8f_{-1} + 8f_1 - f_2)$
h^2f''	$f_{-1} - 2f_0 + f_1$	$\frac{1}{12}(-f_{-2} + 16f_{-1} - 30f_0 + 16f_1 - f_2)$
h^3f'''	$\pm(-f_{\mp1} + 3f_0 - 3f_{\pm1} + f_{\pm2})$	$\frac{1}{2}(-f_{-2} + 2f_{-1} - 2f_1 + f_2)$
$h^4f^{(\mathrm{iv})}$	...	$f_{-2} - 4f_{-1} + 6f_0 - 4f_1 + f_2$

priate combinations of Eqs. (1.2). For example, it is easy to see that

$$f_1 - 2f_0 + f_{-1} = h^2 f'' + \mathcal{O}(h^4)\,, \tag{1.6}$$

so that an approximation to the second derivative accurate to order h^2 is

$$f'' \approx \frac{f_1 - 2f_0 + f_{-1}}{h^2}\,. \tag{1.7}$$

Difference formulas for the various derivatives of f that are accurate to a higher order in h can be derived straightforwardly. Table 1.2 is a summary of the 4- and 5-point expressions.

■ **Exercise 1.1** Using any function for which you can evaluate the derivatives analytically, investigate the accuracy of the formulas in Table 1.2 for various values of h.

1.2 Numerical quadrature

In quadrature, we are interested in calculating the definite integral of f between two limits, $a < b$. We can easily arrange for these values to be points of the lattice separated by an even number of lattice spacings; i.e.,

$$N = \frac{(b-a)}{h}$$

is an even integer. It is then sufficient for us to derive a formula for the integral from $-h$ to $+h$, since this formula can be composed many times:

$$\int_a^b f(x)\,dx = \int_a^{a+2h} f(x)\,dx + \int_{a+2h}^{a+4h} f(x)\,dx + \ldots + \int_{b-2h}^{b} f(x)\,dx\,. \tag{1.8}$$

The basic idea behind all of the quadrature formulas we will discuss (technically of the closed Newton-Cotes type) is to approximate f between $-h$ and $+h$ by a function that can be integrated exactly. For example, the simplest approximation can be had by considering the intervals $[-h, 0]$ and $[0, h]$ separately, and assuming that f is linear in each of these intervals (see Figure 1.1). The error made by this interpolation is of order $h^2 f''$, so that the approximate integral is

$$\int_{-h}^{h} f(x)\,dx = \frac{h}{2}(f_{-1} + 2f_0 + f_1) + \mathcal{O}(h^3)\,, \tag{1.9}$$

which is the well-known trapezoidal rule.

A better approximation can be had by realizing that the Taylor series (1.1) can provide an improved interpolation of f. Using the difference formulas (1.3b) and (1.7) for f' and f'', respectively, for $|x| < h$ we can put

$$f(x) = f_0 + \frac{f_1 - f_{-1}}{2h}x + \frac{f_1 - 2f_0 + f_{-1}}{2h^2}x^2 + \mathcal{O}(x^3)\,, \tag{1.10}$$

which can be integrated readily to give

$$\int_{-h}^{h} f(x)\,dx = \frac{h}{3}(f_1 + 4f_0 + f_{-1}) + \mathcal{O}(h^5)\,. \tag{1.11}$$

This is Simpson's rule, which can be seen to be accurate to two orders higher than the trapezoidal rule (1.9). Note that the error is actually better than would be expected naively from (1.10) since the x^3 term gives no contribution to the integral. Composing this formula according to Eq. (1.8) gives

$$\begin{aligned}\int_a^b f(x)\,dx = \frac{h}{3}[f(a) + 4f(a+h) + 2f(a+2h) + 4f(a+3h) \\ \ldots + 4f(b-h) + f(b)]\,.\end{aligned} \tag{1.12}$$

As an example, the following FORTRAN program calculates

$$\int_0^1 e^x\,dx = e - 1 = 1.718282$$

using Simpson's rule for the value of $N = 1/h$ input. [Source code for the longer programs like this that are embedded in the text are contained on the *Computational Physics* diskette or available over Internet (see Appendix E); the shorter codes can be easily entered into the reader's computer from the keyboard.]

```
C chap1b.for
      FUNC(X)=EXP(X)                    !function to integrate
      EXACT=EXP(1.)-1.
30    PRINT *,'ENTER N EVEN (.LT. 2 TO STOP)'
      READ *, N
      IF (N .LT. 2) STOP
      IF (MOD(N,2) .NE. 0) N=N+1
      H=1./N
      SUM=FUNC(0.)                      !contribution from X=0
      FAC=2                             !factor for Simpson's rule
      DO 10 I=1,N-1                     !loop over lattice points
         IF (FAC .EQ. 2.) THEN          !factors alternate
            FAC=4
         ELSE
            FAC=2.
         END IF
         X=I*H                          !X at this point
         SUM=SUM+FAC*FUNC(X)            !contribution to the integral
10    CONTINUE
      SUM=SUM+FUNC(1.)                  !contribution from X=1
      XINT=SUM*H/3.
      DIFF=EXACT-XINT
      PRINT 20,N,DIFF
20    FORMAT (5X,'N=',I5,5X,'ERROR=',E15.8)
      GOTO 30                           !get another value of N
      END
```

Results are shown in Table 1.3 for various values of N, together with the values obtained using the trapezoidal rule. The improvement from the higher-order formula is evident. Note that the results are stable in the sense that a well-defined limit is obtained as N becomes very large and the mesh spacing h becomes small; round-off errors are unimportant because all values of f enter into the quadrature formula with the same sign, in contrast to what happens in numerical differentiation.

An important issue in quadrature is how small an h is necessary to compute the integral to a given accuracy. Although it is possible to derive rigorous error bounds for the formulas we have discussed, the simplest thing to do in practice is to run the computation again with a smaller h and observe the changes in the results.

Table 1.3 Errors in evaluating $\int_0^1 e^x dx = 1.718282$

N	h	Trapezoidal Eq. (1.9)	Simpson's Eq. (1.12)	Bode's Eq. (1.13b)
4	0.2500000	−0.008940	−0.000037	−0.000001
8	0.1250000	−0.002237	0.000002	0.000000
16	0.0625000	−0.000559	0.000000	0.000000
32	0.0312500	−0.000140	0.000000	0.000000
64	0.0156250	−0.000035	0.000000	0.000000
128	0.0078125	−0.000008	0.000000	0.000000

Higher-order quadrature formulas can be derived by retaining more terms in the Taylor expansion (1.10) used to interpolate f between the mesh points and, of course, using commensurately better finite-difference approximations for the derivatives. The generalizations of Simpson's rule using cubic and quartic polynomials to interpolate (Simpson's $\frac{3}{8}$ and Bode's rule, respectively) are:

$$\int_{x_0}^{x_3} f(x)\,dx = \frac{3h}{8}[f_0 + 3f_1 + 3f_2 + f_3] + \mathcal{O}(h^5)\,; \tag{1.13a}$$

$$\int_{x_0}^{x_4} f(x)\,dx = \frac{2h}{45}[7f_0 + 32f_1 + 12f_2 + 32f_3 + 7f_4] + \mathcal{O}(h^7)\,. \tag{1.13b}$$

The results of applying Bode's rule are also given in Table 1.3, where the improvement is evident, although at the expense of a more involved computation. (Note that for this method to be applicable, N must be a multiple of 4.) Although one might think that formulas based on interpolation using polynomials of a very high degree would be even more suitable, this is not the case; such polynomials tend to oscillate violently and lead to an inaccurate interpolation. Moreover, the coefficients of the values of f at the various lattice points can have both positive and negative signs in higher-order formulas, making round-off error a potential problem. It is therefore usually safer to improve accuracy by using a low-order method and making h smaller rather than by resorting to a higher-order formula. Quadrature formulas accurate to a very high order can be derived if we give up the requirement of equally-spaced abscissae; these are discussed in Chapter 4.

■ **Exercise 1.2** Using any function whose definite integral you can compute analytically, investigate the accuracy of the various quadrature meth-

ods discussed above for different values of h.

Some care and common sense must be exercised in the application of the numerical quadrature formulas discussed above. For example, an integral in which the upper limit is very large is best handled by a change in variable. Thus, the Simpson's rule evaluation of

$$\int_1^b dx\, x^{-2}\, g(x)$$

with $g(x)$ constant at large x, would result in a (finite) sum converging very slowly as b becomes large for fixed h (and taking a very long time to compute!). However, changing variables to $t = x^{-1}$ gives

$$\int_{b^{-1}}^1 g(t^{-1})\, dt\,,$$

which can then be evaluated by any of the formulas we have discussed.

Integrable singularities, which cause the naive formulas to give nonsense, can also be handled in a simple way. For example,

$$\int_0^1 dx(1-x^2)^{-1/2}\, g(x)$$

has an integrable singularity at $x = 1$ (if g is regular there) and is a finite number. However, since $f(x = 1) = \infty$, the quadrature formulas discussed above give an infinite result. An accurate result can be obtained by changing variables to $t = (1-x)^{1/2}$ to obtain

$$2\int_0^1 dt\,(2-t^2)^{-1/2} g(1-t^2)\,,$$

which is then approximated with no trouble.

Integrable singularities can also be handled by deriving quadrature formulas especially adapted to them. Suppose we are interested in

$$\int_0^1 f(x)\, dx = \int_0^h f(x)\, dx + \int_h^1 f(x)\, dx\,,$$

where $f(x)$ behaves as $Cx^{-1/2}$ near $x = 0$, with C a constant. The integral from h to 1 is regular and can be handled easily, while the integral from 0 to h can be approximated as $2Ch^{1/2} = 2hf(h)$.

■ **Exercise 1.3** Write a program to calculate

$$\int_0^1 t^{-2/3}(1-t)^{-1/3}\, dt = 2\pi/3^{1/2}$$

using one of the quadrature formulas discussed above and investigate its accuracy for various values of h. (Hint: Split the range of integration into two parts and make a different change of variable in each integral to handle the singularities.)

1.3 Finding roots

The final elementary operation that is commonly required is to find a root of a function $f(x)$ that we can compute for arbitrary x. One sure-fire method, when the approximate location of a root (say at $x = x_0$) is known, is to guess a trial value of x guaranteed to be less than the root, and then to increase this trial value by small positive steps, backing up and halving the step size every time f changes sign. The values of x generated by this procedure evidently converge to x_0, so that the search can be terminated whenever the step size falls below the required tolerance. Thus, the following FORTRAN program finds the positive root of the function $f(x) = x^2 - 5$, $x_0 = 5^{1/2} = 2.236068$, to a tolerance of 10^{-6} using $x = 1$ as an initial guess and an initial step size of 0.5:

```
C chap1c.for
      FUNC(X)=X*X-5.                !function whose root is sought
      TOLX=1.E-06                   !tolerance for the search
      X=1.                          !initial guess
      FOLD=FUNC(X)                  !initial function
      DX=.5                         !initial step
      ITER=0                        !initialize count
10    CONTINUE
        ITER=ITER+1                     !increment iteration count
        X=X+DX                          !step X
        PRINT *,ITER,X,SQRT(5.)-X       !output current values
        IF ((FOLD*FUNC(X)) .LT. 0) THEN
          X=X-DX                        !if sign change, back up
          DX=DX/2                       ! and halve the step
        END IF
      IF (ABS(DX) .GT. TOLX) GOTO 10
      STOP
      END
```

Results for the sequence of x values are shown in Table 1.4, evidently converging to the correct answer, although only after some 33 iterations. One must be careful when using this method, since if the initial step size

Table 1.4 Error in finding the positive root of $f(x) = x^2 - 5$

Iteration	Search	Newton Eq. (1.14)	Secant Eq. (1.15)
0	1.236076	1.236076	1.236076
1	0.736068	−0.763932	−1.430599
2	0.236068	−0.097265	0.378925
3	−0.263932	−0.002027	0.098137
4	−0.013932	−0.000001	−0.009308
5	0.111068	0.000000	0.000008
6	−0.013932	0.000000	0.000000
⋮	⋮	⋮	⋮
33	0.000001	0.000000	0.000000

is too large, it is possible to step over the root desired when f has several roots.

■ **Exercise 1.4** Run the code above for various tolerances, initial guesses, and initial step sizes. Note that sometimes you might find convergence to the negative root. What happens if you start with an initial guess of -3 with a step size of 6?

A more efficient algorithm, Newton-Raphson, is available if we can evaluate the derivative of f for arbitrary x. This method generates a sequence of values, x^i, converging to x_0 under the assumption that f is locally linear near x_0 (see Figure 1.2). That is,

$$x^{i+1} = x^i - \frac{f(x^i)}{f'(x^i)} . \tag{1.14}$$

The application of this method to finding $5^{1/2}$ is also shown in Table 1.4, where the rapid convergence (5 iterations) is evident. This is the algorithm usually used in the computer evaluation of square roots; a linearization of (1.14) about x_0 shows that the number of significant digits doubles with each iteration, as is evident in Table 1.4.

The secant method provides a happy compromise between the efficiency of Newton-Raphson and the bother of having to evaluate the derivative. If the derivative in Eq. (1.14) is approximated by the differ-

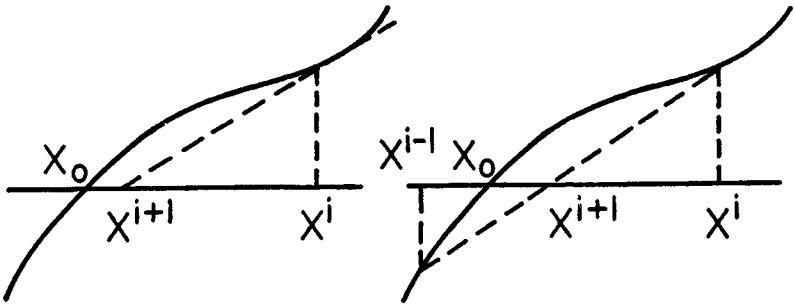


Figure 1.2 Geometrical bases of the Newton-Raphson (left) and secant (right) methods.

ence formula related to (1.4b),

$$f'(x^i) \approx \frac{f(x^i) - f(x^{i-1})}{x^i - x^{i-1}} ,$$

we obtain the following 3-term recursion formula giving x^{i+1} in terms of x^i and x^{i-1} (see Figure 1.2):

$$x^{i+1} = x^i - f(x^i)\frac{(x^i - x^{i-1})}{f(x^i) - f(x^{i-1})} . \tag{1.15}$$

Any two approximate values of x_0 can be used for x^0 and x^1 to start the algorithm, which is terminated when the change in x from one iteration to the next is less than the required tolerance. The results of the secant method for our model problem, starting with values $x^0 = 0.5$ and $x^1 = 1.0$, are also shown in Table 1.4. Provided that the initial guesses are close to the true root, convergence to the exact answer is almost as rapid as that of the Newton-Raphson algorithm.

■ **Exercise 1.5** Write programs to solve for the positive root of $x^2 - 5$ using the Newton-Raphson and secant methods. Investigate the behavior of the latter with changes in the initial guesses for the root.

When the function is badly behaved near its root (e.g., there is an inflection point near x_0) or when there are several roots, the "automatic" Newton-Raphson and secant methods can fail to converge at all or converge to the wrong answer if the initial guess for the root is poor. Hence, a safe and conservative procedure is to use the search algorithm to locate x_0 approximately and then to use one of the automatic methods.

■ **Exercise 1.6** The function $f(x) = \tanh x$ has a root at $x = 0$. Write a program to show that the Newton-Raphson method does not converge for an initial guess of $x \gtrsim 1$. Can you understand what's going wrong by considering a graph of $\tanh x$? From the explicit form of (1.14) for this problem, derive the critical value of the initial guess above which convergence will not occur. Try to solve the problem using the secant method. What happens for various initial guesses if you try to find the $x = 0$ root of $\tan x$ using either method?

1.4 Semiclassical quantization of molecular vibrations

As an example combining several basic mathematical operations, we consider the problem of describing a diatomic molecule such as O_2, which consists of two nuclei bound together by the electrons that orbit about them. Since the nuclei are much heavier than the electrons we can assume that the latter move fast enough to readjust instantaneously to the changing position of the nuclei (Born-Oppenheimer approximation). The problem is therefore reduced to one in which the motion of the two nuclei is governed by a potential, V, depending only upon r, the distance between them. The physical principles responsible for generating V will be discussed in detail in Project VIII, but on general grounds one can say that the potential is attractive at large distances (van der Waals interaction) and repulsive at short distances (Coulomb interaction of the nuclei and Pauli repulsion of the electrons). A commonly used form for V embodying these features is the Lennard-Jones or 6–12 potential,

$$V(r) = 4V_0\left[\left(\frac{a}{r}\right)^{12} - \left(\frac{a}{r}\right)^{6}\right], \tag{1.16}$$

which has the shape shown in the upper portion of Figure 1.3, the minimum occurring at $r_{\min} = 2^{1/6}a$ with a depth V_0. We will assume this form in most of the discussion below. A thorough treatment of the physics of diatomic molecules can be found in [He50] while the Born-Oppenheimer approximation is discussed in [Me68].

The great mass of the nuclei allows the problem to be simplified even further by decoupling the slow rotation of the nuclei from the more rapid changes in their separation. The former is well described by the quantum mechanical rotation of a rigid dumbbell, while the vibrational states of relative motion, with energies E_n, are described by the bound

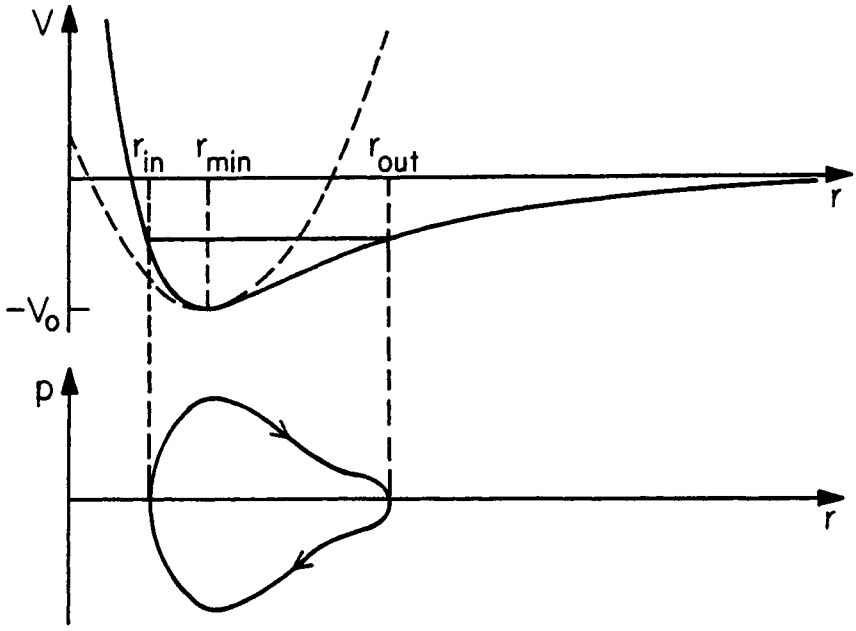


Figure 1.3 (Upper portion) The Lennard-Jones potential and the inner and outer turning points at a negative energy. The dashed line shows the parabolic approximation to the potential. (Lower portion) The corresponding trajectory in phase space.

state solutions, $\psi_n(r)$, of a one-dimensional Schroedinger equation,

$$\left[-\frac{\hbar^2}{2m}\frac{d^2}{dr^2} + V(r)\right]\psi_n = E_n\psi_n \,. \tag{1.17}$$

Here, m is the reduced mass of the two nuclei.

Our goal in this example is to find the energies E_n, given a particular potential. This can be done exactly by solving the differential eigenvalue equation (1.17); numerical methods for doing so will be discussed in Chapter 3. However, the great mass of the nuclei implies that their motion is nearly classical, so that approximate values of the vibrational energies E_n can be obtained by considering the classical motion of the nuclei in V and then applying "quantization rules" to determine the energies. These quantization rules, originally postulated by N. Bohr and Sommerfeld and Wilson, were the basis of the "old" quantum theory

from which the modern formulation of quantum mechanics arose. However, they can also be obtained by considering the WKB approximation to the wave equation (1.17). (See [Me68] for details.)

Confined classical motion of the internuclear separation in the potential $V(r)$ can occur for energies $-V_0 < E < 0$. The distance between the nuclei oscillates periodically (but not necessarily harmonically) between inner and outer turning points, r_{in} and r_{out}, as shown in Figure 1.3. During these oscillations, energy is exchanged between the kinetic energy of relative motion and the potential energy such that the total energy,

$$E = \frac{p^2}{2m} + V(r)\,, \tag{1.18}$$

is a constant (p is the relative momentum of the nuclei). We can therefore think of the oscillations at any given energy as defining a closed trajectory in phase space (coordinates r and p) along which Eq. (1.18) is satisfied, as shown in the lower portion of Figure 1.3. An explicit equation for this trajectory can be obtained by solving (1.18) for p:

$$p(r) = \pm\big[2m\big(E - V(r)\big)\big]^{1/2}\,. \tag{1.19}$$

The classical motion described above occurs at *any* energy between $-V_0$ and 0. To quantize the motion, and hence obtain approximations to the eigenvalues E_n appearing in (1.17), we consider the dimensionless action at a given energy,

$$S(E) = \oint k(r)\,dr\,, \tag{1.20}$$

where $k(r) = \hbar^{-1}p(r)$ is the local de Broglie wave number and the integral is over one complete cycle of oscillation. This action is just the area (in units of $\hbar$) enclosed by the phase space trajectory. The quantization rules state that, at the allowed energies E_n, the action is a half-integral multiple of 2π. Thus, upon using (1.19) and recalling that the oscillation passes through each value of r twice (once with positive p and once with negative p), we have

$$S(E_n) = 2\left(\frac{2m}{\hbar^2}\right)^{1/2} \int_{r_{\text{in}}}^{r_{\text{out}}} \big[E_n - V(r)\big]^{1/2}\,dr = \big(n + \tfrac{1}{2}\big)2\pi\,, \tag{1.21}$$

where n is a nonnegative integer. At the limits of this integral, the turning points r_{in} and r_{out}, the integrand vanishes.

To specialize the quantization condition to the Lennard-Jones potential (1.16), we define the dimensionless quantities

$$\epsilon = \frac{E}{V_0}\,, \quad x = \frac{r}{a}\,, \quad \gamma = \left(\frac{2ma^2V_0}{\hbar^2}\right)^{1/2}\,,$$

so that (1.21) becomes

$$s(\epsilon_n) \equiv \frac{1}{2}S(\epsilon_n V_0) = \gamma \int_{x_{\rm in}}^{x_{\rm out}} \left[\epsilon_n - v(x)\right]^{1/2} dx = \left(n + \frac{1}{2}\right)\pi\,, \qquad (1.22)$$

where

$$v(x) = 4\left(\frac{1}{x^{12}} - \frac{1}{x^6}\right)$$

is the scaled potential.

The quantity γ is a dimensionless measure of the quantum nature of the problem. In the classical limit ($\hbar$ small or m large), γ becomes large. By knowing the moment of inertia of the molecule (from the energies of its rotational motion) and the dissociation energy (energy required to separate the molecule into its two constituent atoms), it is possible to determine from observation the parameters a and V_0 and hence the quantity γ. For the H_2 molecule, $\gamma = 21.7$, while for the HD molecule, $\gamma = 24.8$ (only m, but not V_0, changes when one of the protons is replaced by a deuteron), and for the much heavier O_2 molecule made of two ^{16}O nuclei, $\gamma = 150$. These rather large values indicate that a semiclassical approximation is a valid description of the vibrational motion.

The FORTRAN program for Example 1, whose source code is contained in Appendix B and in the file EXMPL1.FOR, finds, for the value of γ input, the values of the ϵ_n for which Eq. (1.22) is satisfied. After all of the energies have been found, the corresponding phase space trajectories are drawn. (Before attempting to run this code on your computer system, you should review the material on the programs in "How to use this book" and in Appendix A.)

The following exercises are aimed at increasing your understanding of the physical principles and numerical methods demonstrated in this example.

■ **Exercise 1.7** One of the most important aspects of using a computer as a tool to do physics is knowing when to have confidence that the program is giving the correct answers. In this regard, an essential test is the detailed quantitative comparison of results with what is known in analytically soluble situations. Modify the code to use a parabolic potential

(in subroutine POT, taking care to heed the instructions given there), for which the Bohr-Sommerfeld quantization gives the exact eigenvalues of the Schroedinger equation: a series of equally-spaced energies, with the lowest being one-half of the level spacing above the minimum of the potential. For several values of γ, compare the numerical results for this case with what you obtain by solving Eq. (1.22) analytically. Are the phase space trajectories what you expect?

■ **Exercise 1.8** Another important test of a working code is to compare its results with what is expected on the basis of physical intuition. Restore the code to use the Lennard-Jones potential and run it for $\gamma = 50$. Note that, as in the case of the purely parabolic potential discussed in the previous exercise, the first excited state is roughly three times as high above the bottom of the well as is the ground state and that the spacings between the few lowest states are roughly constant. This is because the Lennard-Jones potential is roughly parabolic about its minimum (see Figure 1.3). By calculating the second derivative of V at the minimum, find the "spring constant" and show that the frequency of small-amplitude motion is expected to be

$$\frac{\hbar\omega}{V_0} = \frac{6 \times 2^{5/6}}{\gamma} \approx \frac{10.691}{\gamma} . \tag{1.23}$$

Verify that this is consistent with the numerical results and explore this agreement for different values of γ. Can you understand why the higher energies are more densely spaced than the lower ones by comparing the Lennard-Jones potential with its parabolic approximation?

■ **Exercise 1.9** Invariance of results under changes in the numerical algorithms or their parameters can give additional confidence in a calculation. Change the tolerances for the turning point and energy searches or the number of Simpson's rule points (this can be done at run-time by choosing menu option 2) and observe the effects on the results. Note that because of the way in which the expected number of bound states is calculated (see the end of subroutine PARAM) this quantity can change if the energy tolerance is varied.

■ **Exercise 1.10** Replace the searches for the inner and outer turning points by the Newton-Raphson method or the secant method. (When ILEVEL$\neq$ 0, the turning points for ILEVEL $-$ 1 are excellent starting values.) Replace the Simpson's rule quadrature for s by a higher-order formula [Eqs. (1.13a) or (1.13b)] and observe the improvement.

Table 1.5 Experimental vibrational energies of the H_2 molecule

n	E_n (eV)	n	E_n (eV)
0	−4.477	8	−1.151
1	−3.962	9	−0.867
2	−3.475	10	−0.615
3	−3.017	11	−0.400
4	−2.587	12	−0.225
5	−2.185	13	−0.094
6	−1.811	14	−0.017
7	−1.466		

■ **Exercise 1.11** Plot the ϵ_n of the Lennard-Jones potential as functions of γ for γ running from 20 to 200 and interpret the results. (As with many other short calculations, you may find it more efficient simply to run the code and plot the results by hand as you go along, rather than trying to automate the plotting operation.)

■ **Exercise 1.12** For the H_2 molecule, observations show that the depth of the potential is $V_0 = 4.747$ eV and the location of the potential minimum is $r_{\min} = 0.74166$ Å. These two quantities, together with Eq. (1.23), imply a vibrational frequency of

$$\hbar\omega = 0.492 V_0 = 2.339 \text{ eV} ,$$

more than four times larger than the experimentally observed energy difference between the ground and first vibrational state, 0.515 eV. The Lennard-Jones shape is therefore not a very good description of the potential of the H_2 molecule. Another defect is that it predicts 6 bound states, while 15 are known to exist. (See Table 1.5, whose entries are derived from the data quoted in [Wa67].) A better analytic form of the potential, with more parameters, is required to reproduce simultaneously the depth and location of the minimum, the frequency of small amplitude vibrations about it, and the total number of bound states. One such form is the Morse potential,

$$V(r) = V_0\left[(1 - e^{-\beta(r-r_{\min})})^2 - 1\right] , \tag{1.24}$$

which also can be solved analytically. The Morse potential has a minimum at the expected location and the parameter β can be adjusted to fit the curvature of the minimum to the observed excitation energy of the first

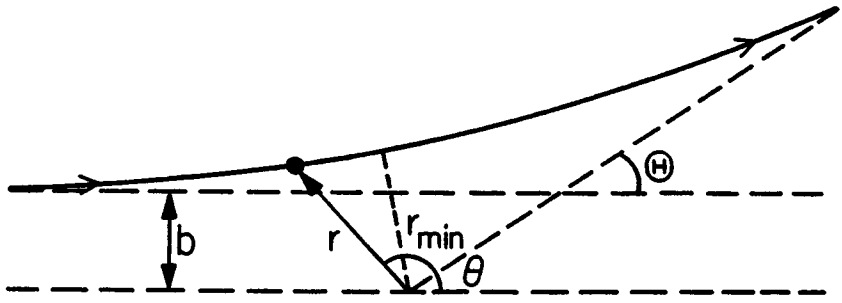


Figure I.1 Quantities involved in the scattering of a particle by a central potential.

vibrational state. Find the value of β appropriate for the H_2 molecule, modify the program above to use the Morse potential, and calculate the spectrum of vibrational states. Show that a much more reasonable number of levels is now obtained. Compare the energies with experiment and with those of the Lennard-Jones potential and interpret the latter differences.

Project I: Scattering by a central potential

In this project, we will investigate the classical scattering of a particle of mass m by a central potential, in particular the Lennard-Jones potential considered in Section 1.4 above. In a scattering event, the particle, with initial kinetic energy E and impact parameter b, approaches the potential from a large distance. It is deflected during its passage near the force center and eventually emerges with the same energy, but moving at an angle Θ with respect to its original direction. Since the potential depends upon only the distance of the particle from the force center, the angular momentum is conserved and the trajectory lies in a plane. The polar coordinates of the particle, (r, θ), are a convenient way to describe its motion, as shown in Figure I.1. (For details, see any textbook on classical mechanics, such as [Go80].)

Of basic interest is the deflection function, $\Theta(b)$, giving the final scattering angle, Θ, as a function of the impact parameter; this function also depends upon the incident energy. The differential cross section for scattering at an angle Θ, $d\sigma/d\Omega$, is an experimental observable that is

related to the deflection function by

$$\frac{d\sigma}{d\Omega} = \frac{b}{\sin\Theta}\left|\frac{db}{d\Theta}\right| . \tag{I.1}$$

Thus, if $d\Theta/db = (db/d\Theta)^{-1}$ can be computed, then the cross section is known.

Expressions for the deflection function can be found analytically for only a very few potentials, so that numerical methods usually must be employed. One way to solve the problem would be to integrate the equations of motion in time (i.e., Newton's law relating the acceleration to the force) to find the trajectories corresponding to various impact parameters and then to tabulate the final directions of the motion (scattering angles). This would involve integrating four coupled first-order differential equations for two coordinates and their velocities in the scattering plane, as discussed in Section 2.5 below. However, since angular momentum is conserved, the evolution of θ is related directly to the radial motion, and the problem can be reduced to a one-dimensional one, which can be solved by quadrature. This latter approach, which is simpler and more accurate, is the one we will pursue here.

To derive an appropriate expression for Θ, we begin with the conservation of angular momentum, which implies that

$$L = mvb = mr^2\frac{d\theta}{dt} , \tag{I.2}$$

is a constant of the motion. Here, $d\theta/dt$ is the angular velocity and v is the asymptotic velocity, related to the bombarding energy by $E = \frac{1}{2}mv^2$. The radial motion occurs in an effective potential that is the sum of V and the centrifugal potential, so that energy conservation implies

$$\frac{1}{2}m\left(\frac{dr}{dt}\right)^2 + \frac{L^2}{2mr^2} + V = E . \tag{I.3}$$

If we use r as the independent variable in (I.2), rather than the time, we can write

$$\frac{d\theta}{dr} = \frac{d\theta}{dt}\left(\frac{dr}{dt}\right)^{-1} = \frac{bv}{r^2}\left(\frac{dr}{dt}\right)^{-1} , \tag{I.4}$$

and solving (I.3) for dr/dt then yields

$$\frac{d\theta}{dr} = \pm\frac{b}{r^2}\left(1 - \frac{b^2}{r^2} - \frac{V}{E}\right)^{-1/2} . \tag{I.5}$$

Recalling that $\theta = \pi$ when $r = \infty$ on the incoming branch of the trajectory and that θ is always decreasing, this equation can be integrated immediately to give the scattering angle,

$$\Theta = \pi - 2\int_{r_{\min}}^{\infty} \frac{b\,dr}{r^2}\left(1 - \frac{b^2}{r^2} - \frac{V}{E}\right)^{-1/2}, \qquad (I.6)$$

where $r_{\min}$ is the distance of closest approach (the turning point, determined by the outermost zero of the argument of the square root) and the factor of 2 in front of the integral accounts for the incoming and outgoing branches of the trajectory, which give equal contributions to the scattering angle.

One final transformation is useful before beginning a numerical calculation. Suppose that there exists a distance $r_{\max}$ beyond which we can safely neglect V. In this case, the integrand in (I.6) vanishes as r^{-2} for large r, so that numerical quadrature could be very inefficient. In fact, since the potential has no effect for $r > r_{\max}$, we would just be "wasting time" describing straight-line motion. To handle this situation efficiently, note that since $\Theta = 0$ when $V = 0$, Eq. (I.6) implies that

$$\pi = 2\int_{b}^{\infty} \frac{b\,dr}{r^2}\left(1 - \frac{b^2}{r^2}\right)^{-1/2}, \qquad (I.7)$$

which, when substituted into (I.6), results in

$$\Theta = 2b\left[\int_{b}^{r_{\max}} \frac{dr}{r^2}\left(1 - \frac{b^2}{r^2}\right)^{-1/2} - \int_{r_{\min}}^{r_{\max}} \frac{dr}{r^2}\left(1 - \frac{b^2}{r^2} - \frac{V}{E}\right)^{-1/2}\right]. \qquad (I.8)$$

The integrals here extend only to $r_{\max}$ since the integrands become equal when $r > r_{\max}$.

Our goal will be to study scattering by the Lennard-Jones potential (1.16), which we can safely set to zero beyond $r_{\max} = 3a$ if we are not interested in energies smaller than about

$$V(r = 3a) \approx 5 \times 10^{-3} V_0 .$$

The study is best done in the following sequence of steps:

Step 1 Before beginning *any* numerical computation, it is important to have some idea of what the results should look like. Sketch what you think the deflection function is at relatively low energies, $E \lesssim V_0$, where the peripheral collisions at large $b \leq r_{\max}$ will take place in a predominantly attractive potential and the more central collisions will "bounce" against the repulsive core. What happens at much higher energies, $E \gg V_0$, where

the attractive pocket in V can be neglected? Note that for values of b where the deflection function has a maximum or a minimum, Eq. (I.1) shows that the cross section will be infinite, as occurs in the rainbow formed when light scatters from water drops.

Step 2 To have analytically soluble cases against which to test your program, calculate the deflection function for a square potential, where $V(r) = U_0$ for $r < r_{\max}$ and vanishes for $r > r_{\max}$. What happens when U_0 is negative? What happens when U_0 is positive and $E < U_0$? when $E > U_0$?

Step 3 Write a program that calculates, for a specified energy E, the deflection function by a numerical quadrature to evaluate both integrals in Eq. (I.8) at a number of equally spaced b values between 0 and $r_{\max}$. (Note that the singularities in the integrands require some special treatment.) Check that the program is working properly and is accurate by calculating deflection functions for the square-well potential discussed in Step 2. Compare the accuracy with that of an alternative procedure in which the first integral in (I.8) is evaluated analytically, rather than numerically.

Step 4 Use your program to calculate the deflection function for scattering from the Lennard-Jones potential at selected values of E ranging from $0.1\,V_0$ to $100\,V_0$. Reconcile your answers in Step 1 with the results you obtain. Calculate the differential cross section as a function of Θ at these energies.

Step 5 If your program is working correctly, you should observe, for energies $E \lesssim V_0$, a singularity in the deflection function where Θ appears to approach $-\infty$ at some critical value of b, b_{crit}, that depends on E. This singularity, which disappears when E becomes larger than about V_0, is characteristic of "orbiting." In this phenomenon, the integrand in Eq. (I.6) has a linear, rather than a square root, singularity at the turning point, so that the scattering angle becomes logarithmically infinite. That is, the effective potential,

$$V + E\left(\frac{b}{r}\right)^2,$$

has a parabolic maximum and, when $b = b_{\text{crit}}$, the peak of this parabola is equal to the incident energy. The trajectory thus spends a very long time at the radius where this parabola peaks and the particle spirals many times around the force center. By tracing b_{crit} as a function of energy

and by plotting a few of the effective potentials involved, convince yourself that this is indeed what's happening. Determine the maximum energy for which the Lennard-Jones potential exhibits orbiting, either by a solution of an appropriate set of equations involving V and its derivatives or by a systematic numerical investigation of the deflection function. If you pursue the latter approach, you might have to reconsider the treatment of the singularities in the numerical quadratures.

Chapter 2

Ordinary Differential Equations

Many of the laws of physics are most conveniently formulated in terms of differential equations. It is therefore not surprising that the numerical solution of differential equations is one of the most common tasks in modeling physical systems. The most general form of an ordinary differential equation is a set of M coupled first-order equations

$$\frac{d\mathbf{y}}{dx} = \mathbf{f}(x,\mathbf{y}) \,, \tag{2.1}$$

where x is the independent variable and $\mathbf{y}$ is a set of M dependent variables ($\mathbf{f}$ is thus an M-component vector). Differential equations of higher order can be written in this first-order form by introducing auxiliary functions. For example, the one-dimensional motion of a particle of mass m under a force field $F(z)$ is described by the second-order equation

$$m\frac{d^2z}{dt^2} = F(z) \,. \tag{2.2}$$

If we define the momentum

$$p(t) = m\frac{dz}{dt} \,,$$

then (2.2) becomes the two coupled first-order (Hamilton's) equations

$$\frac{dz}{dt} = \frac{p}{m} \;\; ; \;\; \frac{dp}{dt} = F(z) \,, \tag{2.3}$$

which are in the form of (2.1). It is therefore sufficient to consider in detail only methods for first-order equations. Since the matrix structure

of coupled differential equations is of the most natural form, our discussion of the case where there is only one independent variable can be generalized readily. Thus, we need be concerned only with solving

$$\frac{dy}{dx} = f(x, y) \tag{2.4}$$

for a single dependent variable $y(x)$.

In this chapter, we will discuss several methods for solving ordinary differential equations, with emphasis on the initial value problem. That is, find $y(x)$ given the value of y at some initial point, say $y(x = 0) = y_0$. This kind of problem occurs, for example, when we are given the initial position and momentum of a particle and we wish to find its subsequent motion using Eqs. (2.3). In Chapter 3, we will discuss the equally important boundary value and eigenvalue problems.

2.1 Simple methods

To repeat the basic problem, we are interested in the solution of the differential equation (2.4) with the initial condition $y(x = 0) = y_0$. More specifically, we are usually interested in the value of y at a particular value of x, say $x = 1$. The general strategy is to divide the interval $[0, 1]$ into a large number, N, of equally spaced subintervals of length $h = 1/N$ and then to develop a recursion formula relating y_n to $\{y_{n-1}, y_{n-2}, \ldots\}$, where y_n is our approximation to $y(x_n = nh)$. Such a recursion relation will then allow a step-by-step integration of the differential equation from $x = 0$ to $x = 1$.

One of the simplest algorithms is Euler's method, in which we consider Eq. (2.4) at the point x_n and replace the derivative on the left-hand side by its forward difference approximation (1.4a). Thus,

$$\frac{y_{n+1} - y_n}{h} + \mathcal{O}(h) = f(x_n, y_n)\,, \tag{2.5}$$

so that the recursion relation expressing y_{n+1} in terms of y_n is

$$y_{n+1} = y_n + hf(x_n, y_n) + \mathcal{O}(h^2)\,. \tag{2.6}$$

This formula has a local error (that made in taking the single step from y_n to y_{n+1}) that is $\mathcal{O}(h^2)$ since the error in (1.4a) is $\mathcal{O}(h)$. The "global" error made in finding $y(1)$ by taking N such steps in integrating from $x = 0$ to $x = 1$ is then $N\mathcal{O}(h^2) \approx \mathcal{O}(h)$. This error decreases only

linearly with decreasing step size so that half as large an h (and thus twice as many steps) is required to halve the inaccuracy in the final answer. The numerical work for each of these steps is essentially a single evaluation of f.

As an example, consider the differential equation and boundary condition

$$\frac{dy}{dx} = -xy\,;\ y(0) = 1\,, \tag{2.7}$$

whose solution is

$$y = e^{-x^2/2}\,.$$

The following FORTRAN program integrates forward from $x = 0$ to $x = 3$ using Eq. (2.6) with the step size input, printing the result and its error as it goes along.

```
C chap2a.for
      FUNC(X,Y)=-X*Y                  !dy/dx
20    PRINT *, ' Enter step size ( .le. 0 to stop)'
      READ *, H
      IF (H .LE. 0.) STOP
      NSTEP=3./H                      !number of steps to reach X=3
      Y=1.                            !y(0)=1
      DO 10 IX=0,NSTEP-1              !loop over steps
         X=IX*H                       !last X value
         Y=Y+H*FUNC(X,Y)              !new Y value from Eq 2.6
         DIFF=EXP(-0.5*(X+H)**2)-Y !compare with exact value
         PRINT *, IX,X+H,Y,DIFF
10    CONTINUE
      GOTO 20                         !start again with new value of H
      END
```

Errors in the results obtained for

$$y(1) = e^{-1/2} = 0.606531\,,\ \ y(3) = e^{-9/2} = 0.011109$$

with various step sizes are shown in the first two columns of Table 2.1. As expected from (2.6), the errors decrease linearly with smaller h. However, the fractional error (error divided by y) increases with x as more steps are taken in the integration and y becomes smaller.

Table 2.1 Error in integrating $dy/dx = -xy$ with $y(0) = 1$

	Euler's method Eq. (2.6)		Taylor series Eq. (2.10)		Implicit method Eq. (2.18)	
h	$y(1)$	$y(3)$	$y(1)$	$y(3)$	$y(1)$	$y(3)$
0.500	−.143469	.011109	.032312	−.006660	−.015691	.001785
0.200	−.046330	.006519	.005126	−.000712	−.002525	.000255
0.100	−.021625	.003318	.001273	−.000149	−.000631	.000063
0.050	−.010453	.001665	.000317	−.000034	−.000157	.000016
0.020	−.004098	.000666	.000051	−.000005	−.000025	.000003
0.010	−.002035	.000333	.000013	−.000001	−.000006	.000001
0.005	−.001014	.000167	.000003	.000000	−.000001	.000000
0.002	−.000405	.000067	.000001	.000000	.000000	.000000
0.001	−.000203	.000033	.000000	.000000	.000000	.000000

■ **Exercise 2.1** A simple and often stringent test of an accurate numerical integration is to use the final value of y obtained as the initial condition to integrate backward from the final value of x to the starting point. The extent to which the resulting value of y differs from the original initial condition is then a measure of the inaccuracy. Apply this test to the example above.

Although Euler's method seems to work quite well, it is generally unsatisfactory because of its low-order accuracy. This prevents us from reducing the numerical work by using a larger value of h and so taking a smaller number of steps. This deficiency becomes apparent in the example above as we attempt to integrate to larger values of x, where some thought shows that we obtain the absurd result that $y = 0$ (exactly) for $x > h^{-1}$. One simple solution is to change the step size as we go along, making h smaller as x becomes larger, but this soon becomes quite inefficient.

Integration methods with a higher-order accuracy are usually preferable to Euler's method. They offer a much more rapid increase of accuracy with decreasing step size and hence greater accuracy for a fixed amount of numerical effort. One class of simple higher order methods can be derived from a Taylor series expansion for y_{n+1} about y_n:

$$y_{n+1} = y(x_n + h) = y_n + hy'_n + \frac{1}{2}h^2 y''_n + \mathcal{O}(h^3) . \tag{2.8}$$

From (2.4), we have

$$y'_n = f(x_n, y_n) , \tag{2.9a}$$

and

$$y_n'' = \frac{df}{dx}(x_n, y_n) = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}\frac{dy}{dx} = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} f \,, \tag{2.9b}$$

which, when substituted into (2.8), results in

$$y_{n+1} = y_n + hf + \frac{1}{2}h^2\left[\frac{\partial f}{\partial x} + f\frac{\partial f}{\partial y}\right] + \mathcal{O}(h^3) \,, \tag{2.10}$$

where f and its derivatives are to be evaluated at (x_n, y_n). This recursion relation has a local error $\mathcal{O}(h^3)$ and hence a global error $\mathcal{O}(h^2)$, one order more accurate than Euler's method (2.6). It is most useful when f is known analytically and is simple enough to differentiate. If we apply Eq. (2.10) to the example (2.7), we obtain the results shown in the middle two columns of Table 2.1; the improvement over Euler's method is clear. Algorithms with an even greater accuracy can be obtained by retaining more terms in the Taylor expansion (2.8), but the algebra soon becomes prohibitive in all but the simplest cases.

2.2 Multistep and implicit methods

Another way of achieving higher accuracy is to use recursion relations that relate y_{n+1} not just to y_n, but also to points further "in the past," say y_{n-1}, y_{n-2}, ... To derive such formulas, we can integrate one step of the differential equation (2.4) *exactly* to obtain

$$y_{n+1} = y_n + \int_{x_n}^{x_{n+1}} f(x, y)\, dx \,. \tag{2.11}$$

The problem, of course, is that we don't know f over the interval of integration. However, we can use the values of y at x_n and x_{n-1} to provide a linear extrapolation of f over the required interval:

$$f \approx \frac{(x - x_{n-1})}{h} f_n - \frac{(x - x_n)}{h} f_{n-1} + \mathcal{O}(h^2) \,, \tag{2.12}$$

where $f_i \equiv f(x_i, y_i)$. Inserting this into (2.11) and doing the x integral then results in the Adams-Bashforth two-step method,

$$y_{n+1} = y_n + h\left(\frac{3}{2} f_n - \frac{1}{2} f_{n-1}\right) + \mathcal{O}(h^3) \,. \tag{2.13}$$

Related higher-order methods can be derived by extrapolating with higher-degree polynomials. For example, if f is extrapolated by a cubic polynomial fitted to f_n, f_{n-1}, f_{n-2}, and f_{n-3}, the Adams-Bashforth four-step method results:

$$y_{n+1} = y_n + \frac{h}{24}(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}) + \mathcal{O}(h^5) . \quad (2.14)$$

Note that because the recursion relations (2.13) and (2.14) involve several previous steps, the value of y_0 alone is not sufficient information to get them started, and so the values of y at the first few lattice points must be obtained from some other procedure, such as the Taylor series (2.8) or the Runge-Kutta methods discussed below.

■ **Exercise 2.2** Apply the Adams-Bashforth two- and four-step algorithms to the example defined by Eq. (2.7) using Euler's method (2.6) to generate the values of y needed to start the recursion relation. Investigate the accuracy of $y(x)$ for various values of h by comparing with the analytical results and by applying the reversibility test described in Exercise 2.1.

The methods we have discussed so far are all "explicit" in that the y_{n+1} is given directly in terms of the already known value of y_n. "Implicit" methods, in which an equation must be solved to determine y_{n+1}, offer yet another means of achieving higher accuracy. Suppose we consider Eq. (2.4) at a point $x_{n+1/2} \equiv (n + \frac{1}{2})h$ mid-way between two lattice points:

$$\left.\frac{dy}{dx}\right|_{x_{n+1/2}} = f(x_{n+1/2}\ ,\ y_{n+1/2}) . \quad (2.15)$$

If we then use the symmetric difference approximation for the derivative (the analog of (1.3b) with $h \to \frac{1}{2}h$) and replace $f_{n+1/2}$ by the average of its values at the two adjacent lattice points [the error in this replacement is $\mathcal{O}(h^2)$], we can write

$$\frac{y_{n+1} - y_n}{h} + \mathcal{O}(h^2) = \frac{1}{2}[f_n + f_{n+1}] + \mathcal{O}(h^2) , \quad (2.16)$$

which corresponds to the recursion relation

$$y_{n+1} = y_n + \frac{1}{2}h[f(x_n, y_n) + f(x_{n+1}, y_{n+1})] + \mathcal{O}(h^3) . \quad (2.17)$$

This is all well and good, but the appearance of y_{n+1} on both sides of this equation (an implicit equation) means that, in general, we must solve a non-trivial equation (for example, by the Newton-Raphson method discussed in Section 1.3) at each integration step; this can be very time consuming. A particular simplification occurs if f is linear in y, say $f(x,y) = g(x)y$, in which case (2.17) can be solved to give

$$y_{n+1} = \left[\frac{1 + \frac{1}{2}g(x_n)h}{1 - \frac{1}{2}g(x_{n+1})h}\right] y_n \,. \tag{2.18}$$

When applied to the problem (2.7), where $g(x) = -x$, this method gives the results shown in the last two columns of Table 2.1; the quadratic behavior of the error with h is clear.

■ **Exercise 2.3** Apply the Taylor series method (2.10) and the implicit method (2.18) to the example of Eq. (2.7) and obtain the results shown in Table 2.1. Investigate the accuracy of integration to larger values of x.

The Adams-Moulton methods are both multistep and implicit. For example, the Adams-Moulton two-step method can be derived from Eq. (2.11) by using a quadratic polynomial passing through f_{n-1}, f_n, and f_{n+1},

$$f \approx \frac{(x - x_n)(x - x_{n-1})}{2h^2} f_{n+1} - \frac{(x - x_{n+1})(x - x_{n-1})}{h^2} f_n + \frac{(x - x_{n+1})(x - x_n)}{2h^2} f_{n-1} + \mathcal{O}(h^3) \,,$$

to interpolate f over the region from x_n to x_{n+1}. The implicit recursion relation that results is

$$y_{n+1} = y_n + \frac{h}{12}(5f_{n+1} + 8f_n - f_{n-1}) + \mathcal{O}(h^4) \,. \tag{2.19}$$

The corresponding three-step formula, obtained with a cubic polynomial interpolation, is

$$y_{n+1} = y_n + \frac{h}{24}(9f_{n+1} + 19f_n - 5f_{n-1} + f_{n-2}) + \mathcal{O}(h^5) \,. \tag{2.20}$$

Implicit methods are rarely used by solving the implicit equation to take a step. Rather, they serve as bases for "predictor-corrector" algorithms, in which a "prediction" for y_{n+1} based only on an explicit method

is then "corrected" to give a better value by using this prediction in an implicit method. Such algorithms have the advantage of allowing a continuous monitoring of the accuracy of the integration, for example by making sure that the correction is small. A commonly used predictor-corrector algorithm with local error $\mathcal{O}(h^5)$ is obtained by using the explicit Adams-Bashforth four-step method (2.14) to make the prediction, and then calculating the correction with the Adams-Moulton three-step method (2.20), using the predicted value of y_{n+1} to evaluate f_{n+1} on the right-hand side.

2.3 Runge-Kutta methods

As you might gather from the preceding section, there is quite a bit of freedom in writing down algorithms for integrating differential equations and, in fact, a large number of them exist, each having it own peculiarities and advantages. One very convenient and widely used class of methods are the Runge-Kutta algorithms, which come in varying orders of accuracy. We derive here a second-order version to give the spirit of the approach and then simply state the equations for the third- and commonly used fourth-order methods.

To derive a second-order Runge-Kutta algorithm (there are actually a whole family of them characterized by a continuous parameter), we approximate f in the integral of (2.11) by its Taylor series expansion about the mid-point of the integration interval. Thus,

$$y_{n+1} = y_n + hf(x_{n+1/2}\ ,\ y_{n+1/2}) + \mathcal{O}(h^3)\ , \tag{2.21}$$

where the error arises from the quadratic term in the Taylor series, as the linear term integrates to zero. Although it seems as if we need to know the value of $y_{n+1/2}$ appearing in f in the right-hand side of this equation for it to be of any use, this is not quite true. Since the error term is already $\mathcal{O}(h^3)$, an approximation to y_{n+1} whose error is $\mathcal{O}(h^2)$ is good enough. This is just what is provided by the simple Euler's method, Eq. (2.6). Thus, if we define k to be an intermediate approximation to twice the difference between $y_{n+1/2}$ and y_n, the following two-step procedure gives y_{n+1} in terms of y_n:

$$k = hf(x_n, y_n)\ ; \tag{2.22a}$$

$$y_{n+1} = y_n + hf(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k) + \mathcal{O}(h^3)\ . \tag{2.22b}$$

This is a second-order Runge-Kutta algorithm. It embodies the general idea of substituting approximations for the values of y into the right-hand

side of implicit expressions involving f. It is as accurate as the Taylor series or implicit methods (2.10) or (2.17), respectively, but places no special constraints on f, such as easy differentiability or linearity in y. It also uses the value of y at only one previous point, in contrast to the multipoint methods discussed above. However, (2.22) does require the evaluation of f twice for each step along the lattice.

Runge-Kutta schemes of higher-order can be derived in a relatively straightforward way. Any of the quadrature formulas discussed in Chapter 1 can be used to approximate the integral (2.11) by a finite sum of f values. For example, Simpson's rule yields

$$y_{n+1} = y_n + \frac{h}{6}\left[f(x_n, y_n) + 4f(x_{n+1/2}\ ,\ y_{n+1/2}) + f(x_{n+1}\ ,\ y_{n+1})\right] + \mathcal{O}(h^5)\ . \tag{2.23}$$

Schemes for generating successive approximations to the y's appearing in the right-hand side of a commensurate accuracy then complete the algorithms. A third-order algorithm with a local error $\mathcal{O}(h^4)$ is

$$\begin{aligned} k_1 &= hf(x_n, y_n)\ ; \\ k_2 &= hf(x_n + \tfrac{1}{2}h, y_n + \tfrac{1}{2}k_1)\ ; \\ k_3 &= hf(x_n + h, y_n - k_1 + 2k_2)\ ; \\ y_{n+1} &= y_n + \tfrac{1}{6}(k_1 + 4k_2 + k_3) + \mathcal{O}(h^4)\ . \end{aligned} \tag{2.24}$$

It is based on (2.23) and requires three evaluations of f per step. A fourth-order algorithm, which requires f to be evaluated four times for each integration step and has a local accuracy of $\mathcal{O}(h^5)$, has been found by experience to give the best balance between accuracy and computational effort. It can be written as follows, with the k_i as intermediate variables:

$$\begin{aligned} k_1 &= hf(x_n, y_n)\ ; \\ k_2 &= hf(x_n + \tfrac{1}{2}h, y_n + \tfrac{1}{2}k_1)\ ; \\ k_3 &= hf(x_n + \tfrac{1}{2}h, y_n + \tfrac{1}{2}k_2)\ ; \\ k_4 &= hf(x_n + h, y_n + k_3)\ ; \\ y_{n+1} &= y_n + \tfrac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) + \mathcal{O}(h^5)\ . \end{aligned} \tag{2.25}$$

■ **Exercise 2.4** Try out the second-, third-, and fourth-order Runge-Kutta methods discussed above on the problem defined by Eq. (2.7). Compare the computational effort for a given accuracy with that of other methods.

■ **Exercise 2.5** The two coupled first-order equations

$$\frac{dy}{dt} = p\,; \quad \frac{dp}{dt} = -4\pi^2 y \tag{2.26}$$

define simple harmonic motion with period 1. By generalizing one of the single-variable formulas given above to this two-variable case, integrate these equations with any particular initial conditions you choose and investigate the accuracy with which the system returns to its initial state at integral values of t.

2.4 Stability

A major consideration in integrating differential equations is the numerical stability of the algorithm used; i.e., the extent to which round-off or other errors in the numerical computation can be amplified, in many cases enough for this "noise" to dominate the results. To illustrate the problem, let us attempt to improve the accuracy of Euler's method and approximate the derivative in (2.4) directly by the symmetric difference approximation (1.3b). We thereby obtain the three-term recursion relation

$$y_{n+1} = y_{n-1} + 2hf(x_n, y_n) + \mathcal{O}(h^3)\,, \tag{2.27}$$

which superficially looks about as useful as either of the third-order formulas (2.10) or (2.18). However, consider what happens when this method is applied to the problem

$$\frac{dy}{dx} = -y\,; \quad y(x=0) = 1\,, \tag{2.28}$$

whose solution is $y = e^{-x}$. To start the recursion relation (2.27), we need the value of y_1 as well as $y_0 = 1$. This can be obtained by using (2.10) to get

$$y_1 = 1 - h + \frac{1}{2}h^2 + \mathcal{O}(h^3)\,.$$

(This is just the Taylor series for e^{-h}.) The following FORTRAN program then uses the method (2.27) to find y for values of x up to 6 using the value of h input:

Table 2.2 Integration of $dy/dx = -y$ with $y(0) = 1$

x	Exact	Error	x	Exact	Error	x	Exact	Error
0.2	.818731	−.000269	3.3	.036883	−.000369	5.5	.004087	−.001533
0.3	.740818	−.000382	3.4	.033373	−.000005	5.6	.003698	.001618
0.4	.670320	−.000440	3.5	.030197	−.000380	5.7	.003346	−.001858
0.5	.606531	−.000517	3.6	.027324	.000061	5.8	.003028	.001989
0.6	.548812	−.000538	3.7	.024724	−.000400	5.9	.002739	−.002257
			3.8	.022371	.000133	6.0	.002479	.002439

```
C chap2b.for
10      PRINT *,' Enter value of step size ( .le. 0 to stop)'
        READ *, H
        IF (H .LE. 0) STOP
        YMINUS=1                          !Y(0)
        YZERO=1.-H+H**2/2                 !Y(H)
        NSTEP=6./H
        DO 20 IX=2,NSTEP                  !loop over X values
           X=IX*H                         !X at this step
           YPLUS=YMINUS-2*H*YZERO         !Y from Eq. 2.27
           YMINUS=YZERO                   !roll values
           YZERO=YPLUS
           EXACT=EXP(-X)                  !analytic value at this point
           PRINT *, X,EXACT,EXACT-YZERO
20      CONTINUE
        GOTO 10                           !get another H value
        END
```

Note how the three-term recursion is implemented by keeping track of only three local variables, YPLUS, YZERO, and YMINUS.

A portion of the output of this code run for $h = 0.1$ is shown in Table 2.2. For small values of x, the numerical solution is only slightly larger than the exact value, the error being consistent with the $\mathcal{O}(h^3)$ estimate. Then, near $x = 3.5$, an oscillation begins to develop in the numerical solution, which becomes alternately higher and lower than the exact values lattice point by lattice point. This oscillation grows larger as the equation is integrated further (see values near $x = 6$), eventually overwhelming the exponentially decreasing behavior expected.

The phenomenon observed above is a symptom of an instability in the algorithm (2.27). It can be understood as follows. For the problem

(2.28), the recursion relation (2.27) reads

$$y_{n+1} = y_{n-1} - 2hy_n \,. \tag{2.29}$$

We can solve this equation by assuming an exponential solution of the form $y_n = Ar^n$ where A and r are constants. Substituting into (2.29) then results in an equation for r,

$$r^2 + 2hr - 1 = 0 \,,$$

the constant A being unimportant since the recursion relation is linear. The solutions of this equation are

$$r_+ = (1+h^2)^{1/2} - h \approx 1 - h \,; \; r_- = -(1+h^2)^{1/2} - h \approx -(1+h) \,,$$

where we have indicated approximations valid for $h \ll 1$. The positive root is slightly less than one and corresponds to the exponentially decreasing solution we are after. However, the negative root is slightly less than -1, and so corresponds to a spurious solution

$$y_n \sim (-)^n (1+h)^n \,,$$

whose magnitude increases with n and which oscillates from lattice point to lattice point.

The general solution to the linear difference equation (2.27) is a linear combination of these two exponential solutions. Even though we might carefully arrange the initial values y_0 and y_1 so that only the decreasing solution is present for small x, numerical round-off during the recursion relation [Eq. (2.29) shows that two positive quantities are subtracted to obtain a smaller one] will introduce a small admixture of the "bad" solution that will eventually grow to dominate the results. This instability is clearly associated with the three-term nature of the recursion relation (2.29). A good rule of thumb is that instabilities and round-off problems should be watched for whenever integrating a solution that decreases strongly as the iteration proceeds; such a situation should therefore be avoided, if possible. We will see the same sort of instability phenomenon again in our discussion of second-order differential equations in Chapter 3.

■ **Exercise 2.6** Investigate the stability of several other integration methods discussed in this chapter by applying them to the problem (2.28). Can you give analytical arguments to explain the results you obtain?

2.5 Order and chaos in two-dimensional motion

A fundamental advantage of using computers in physics is the ability to treat systems that cannot be solved analytically. In the usual situation, the numerical results generated agree qualitatively with the intuition we have developed by studying soluble models and it is the quantitative values that are of real interest. However, in a few cases computer results defy our intuition (and thereby reshape it) and numerical work is then essential for a proper understanding. Surprisingly, such cases include the dynamics of simple classical systems, where the generic behavior differs *qualitatively* from that of the models covered in a traditional Mechanics course. In this example, we will study some of this surprising behavior by integrating numerically the trajectories of a particle moving in two dimensions. General discussions of these systems can be found in [He80], [Ri80], and [Ab78].

We consider a particle of unit mass moving in a potential, V, in two dimensions and assume that V is such that the particle remains confined for all times if its energy is low enough. If the momenta conjugate to the two coordinates (x, y) are (p_x, p_y), then the Hamiltonian takes the form

$$H = \frac{1}{2}(p_x^2 + p_y^2) + V(x, y) \,. \tag{2.30}$$

Given any particular initial values of the coordinates and momenta, the particle's trajectory is specified by their time evolution, which is governed by four coupled first-order differential equations (Hamilton's equations):

$$\begin{aligned} \frac{dx}{dt} &= \frac{\partial H}{\partial p_x} = p_x \,, \quad \frac{dy}{dt} = \frac{\partial H}{\partial p_y} = p_y \,; \\ \frac{dp_x}{dt} &= -\frac{\partial H}{\partial x} = -\frac{\partial V}{\partial x} \,, \quad \frac{dp_y}{dt} = \frac{\partial H}{\partial y} = -\frac{\partial V}{\partial y} \,. \end{aligned} \tag{2.31}$$

For any V, these equations conserve the energy, E, so that the constraint

$$H(x, y, p_x, p_y) = E$$

restricts the trajectory to lie in a three-dimensional manifold embedded in the four-dimensional phase space. Apart from this, there are very few other general statements that can be made about the evolution of the system.

One important class of two-dimensional Hamiltonians for which additional statements about the trajectories *can* be made are those that are

integrable. For these potentials, there is a second function of the coordinates and momenta, apart from the energy, that is a constant of the motion; the trajectory is thus constrained to a two-dimensional manifold of the phase space. Two familiar kinds of integrable systems are separable and central potentials. In the separable case,

$$V(x,y) = V_x(x) + V_y(y) \,, \tag{2.32}$$

where the $V_{x,y}$ are two independent functions, so that the Hamiltonian separates into two parts, each involving only one coordinate and its conjugate momentum,

$$H = H_x + H_y \,;\; H_{x,y} = \frac{1}{2}p_{x,y}^2 + V_{x,y} \,.$$

The motions in x and y therefore decouple from each other and each of the Hamiltonians $H_{x,y}$ is separately a constant of the motion. (Equivalently, $H_x - H_y$ is the second quantity conserved in addition to $E = H_x + H_y$.) In the case of a central potential,

$$V(x,y) = V(r) \,;\; r = (x^2 + y^2)^{1/2} \,, \tag{2.33}$$

so that the angular momentum, $p_\theta = xp_y - yp_x$, is the second constant of the motion and the Hamiltonian can be written as

$$H = \frac{1}{2}p_r^2 + V(r) + \frac{p_\theta^2}{2r^2} \,,$$

where p_r is the momentum conjugate to r. The additional constraint on the trajectory present in integrable systems allows the equations of motion to be "solved" by reducing the problem to one of evaluating certain integrals, much as we did for one-dimensional motion in Chapter 1. All of the familiar analytically soluble problems of classical mechanics are those that are integrable.

Although the dynamics of integrable systems are simple, it is often not at all easy to make this simplicity apparent. There is no general analytical method for deciding if there is a second constant of the motion in an arbitrary potential or for finding it if there is one. Numerical calculations are not obviously any better, as these supply only the trajectory for given initial conditions and this trajectory can be quite complicated in even familiar cases, as can be seen by recalling the Lissajous patterns

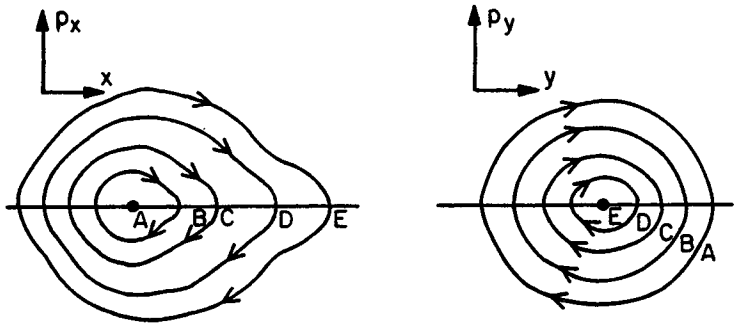


Figure 2.1 Trajectories of a particle in a two-dimensional separable potential as they appear in the (x, p_x) and (y, p_y) planes. Several trajectories corresponding to the same energy but different initial conditions are shown. Trajectories A and E are the limiting ones having vanishing E_x and E_y, respectively.

of the (x, y) trajectories that arise when the motion in both coordinates is harmonic,

$$V_x = \frac{1}{2}\omega_x^2 x^2 \ , \ V_y = \frac{1}{2}\omega_y^2 y^2 \ . \tag{2.34}$$

An analysis in phase space suggests one way to detect integrability from the trajectory alone. Consider, for example, the case of a separable potential. Because the motions of each of the two coordinates are independent, plots of a trajectory in the (x, p_x) and (y, p_y) planes might look as shown in Figure 2.1. Here, we have assumed that each potential $V_{x,y}$ has a single minimum value of 0 at particular values of x and y, respectively. The particle moves on a closed contour in each of these two-dimensional projections of the four-dimensional phase space, each contour looking like that for ordinary one-dimensional motion shown in Figure 1.3. The areas of these contours depend upon how much energy is associated with each coordinate (i.e., E_x and E_y) and, as we consider trajectories with the same energy but with different initial conditions, the area of one contour will shrink as the other grows. In each plane there is a limiting contour that is approached when all of the energy is in that particular coordinate. These contours are the intersections of the two-dimensional phase-space manifolds containing each of the trajectories and the (y, p_y) or (x, p_x) plot is therefore termed a "surface of section." The existence of these closed contours signals the integrability of the system.

Although we are able to construct Figure 2.1 only because we understand the integrable motion involved, a similar plot can be obtained

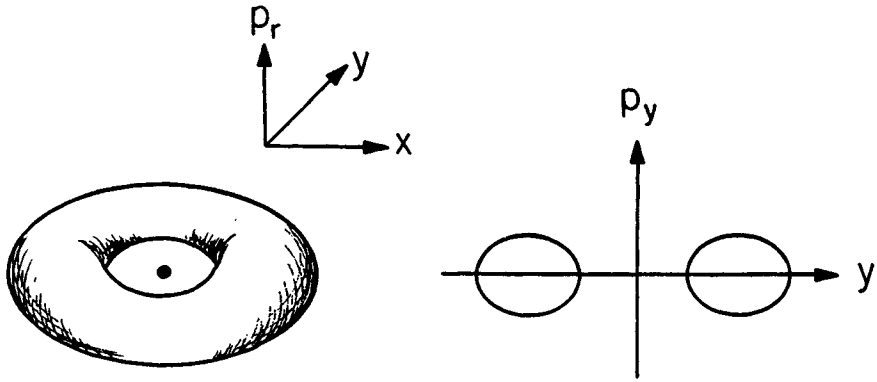


Figure 2.2 (Left) Toroidal manifold containing the trajectory of a particle in a central potential. (Right) Surface of section through this manifold at $x = 0$.

from the trajectory alone. Suppose that every time we observe one of the coordinates, say x, to pass through zero with $p_x > 0$, we plot the location of the particle on the (y, p_y) plane. In other words, crossing through the $x = 0$ plane in a positive sense triggers a "stroboscope" with which we observe the (y, p_y) variables. If the periods of the x and y motions are incommensurate (i.e., their ratio is an irrational number), then, as the trajectory proceeds, these observations will gradually trace out the full (y, p_y) contour; if the periods are commensurate (i.e., a rational ratio), then a series of discrete points around the contour will result. In this way, we can study the topology of the phase space associated with any given Hamiltonian just from the trajectories alone.

The general topology of the phase space for an integrable Hamiltonian can be illustrated by considering motion in a central potential. For fixed values of the energy and angular momentum, the radial motion is bounded between two turning points, r_{in} and r_{out}, which are the solutions of the equation

$$E - V(r) - \frac{p_\theta^2}{2r^2} = 0 \,.$$

These two radii define an annulus in the (x, y) plane to which the trajectory is confined, as shown in the left-hand side of Figure 2.2. Furthermore,

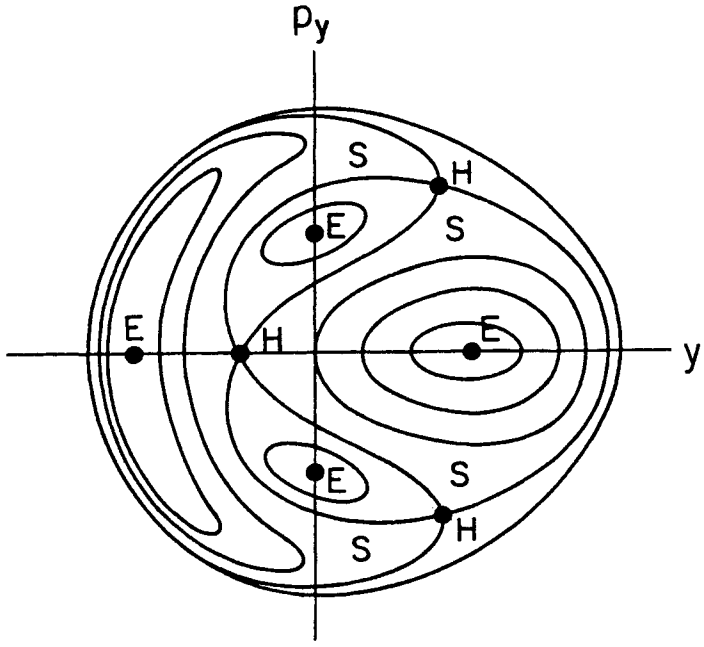


Figure 2.3 Possible features of the surface of section of a general integrable system. E and H label elliptic and hyperbolic fixed points, respectively, while the curve labeled S is a separatrix.

for a fixed value of r, energy conservation permits the radial momentum to take on only one of two values,

$$p_r = \pm\left(2E - 2V(r) - \frac{p_\theta^2}{r^2}\right)^{1/2}$$

These momenta define the two-dimensional manifold in the (x, y, p_r) space that contains the trajectory; it clearly has the topology of a torus, as shown in the left-hand side of Figure 2.2. If we were to construct a (y, p_y) surface of section by considering the $x = 0$ plane, we would obtain two closed contours, as shown in the right-hand side of Figure 2.2. (Note that $y = r$ when $x = 0$.) If the energy is fixed but the angular momentum is changed by varying the initial conditions, the dimensions of this torus change, as does the area of the contour in the surface of section.

The toroidal topology of the phase space of a central potential can be shown to be common to all integrable systems. The manifold on which

the trajectory lies for given values of the constants of the motion is called an "invariant torus," and there are many of them for a given energy. The general surface of section of such tori looks like that shown in Figure 2.3. There are certain fixed points associated with trajectories that repeat themselves exactly after some period. The elliptic fixed points correspond to trajectories that are stable under small perturbations. Around each of them is a family of tori, bounded by a separatrix. The hyperbolic fixed points occur at the intersections of separatrices and are stable under perturbations along one of the axes of the hyperbola, but unstable along the other.

An interesting question is *"What happens to the tori of an integrable system under a perturbation that destroys the integrability of the Hamiltonian?"*. For small perturbations, "most" of the tori about an elliptic fixed point become slightly distorted but retain their topology (the KAM theorem due to Kolmogorov, Arnold, and Moser, [Ar68]). However, adjacent regions of phase space become "chaotic", giving surfaces of section that are a seemingly random splatter of points. Within these chaotic regions are nested yet other elliptic fixed points and other chaotic regions in a fantastic hierarchy. (See Figure 2.4.)

Large deviations from integrability must be investigated numerically. One convenient case for doing so is the potential

$$V(x,y) = \frac{1}{2}(x^2 + y^2) + x^2 y - \frac{1}{3}y^3 \,, \tag{2.35}$$

which was originally introduced by Hénon and Heiles in the study of stellar orbits through a galaxy [He64]. This potential can be thought of as a perturbed harmonic oscillator potential (a small constant multiplying the cubic terms can be absorbed through a rescaling of the coordinates and energy, so that the magnitude of the energy becomes a measure of the deviation from integrability) and has the three-fold symmetry shown in Figure 2.5. The potential is zero at the origin and becomes unbounded for large values of the coordinates. However, for energies less than 1/6, the trajectories remain confined within the equilateral triangle shown.

The FORTRAN program for Example 2, whose source code is contained in Appendix B and in the file EXMPL2.FOR, constructs surfaces of section for the Hénon-Heiles potential. The method used is to integrate the equations of motion (2.31) using the fourth-order Runge-Kutta algorithm (2.25). Initial conditions are specified by putting $x = 0$ and by giving the energy, y, and p_y; p_x is then fixed by energy conservation. As the integration proceeds, the (x,y) trajectory and the (y,p_y) surface

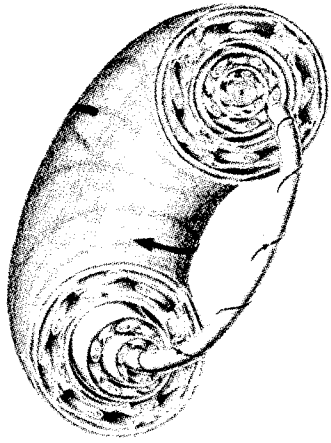


Figure 2.4 Nested tori for a slightly perturbed integrable system. Note the hierarchy of elliptic orbits interspersed with chaotic regions. A magnification of this hierarchy would show the same pattern repeated on a smaller scale and so on, *ad infinitum*. (Reproduced from [Ab78].)

of section are displayed. Points on the latter are calculated by watching for a time step during which x changes sign. When this happens, the precise location of the point on the surface of section plot is determined by switching to x as the independent variable, so that the equations of motion (2.31) become

$$\frac{dx}{dx} = 1\,, \quad \frac{dy}{dx} = \frac{1}{p_x} p_y\,;$$
$$\frac{dp_x}{dx} = -\frac{1}{p_x}\frac{\partial V}{\partial x}\,, \quad \frac{dp_y}{dx} = -\frac{1}{p_x}\frac{\partial V}{\partial y}\,,$$

and then integrating one step backward in x from its value after the time step to 0 [He82]. If the value of the energy is not changed when new initial conditions are specified, all previous surface of section points can be plotted, so that plots like those in Figures 2.3 and 2.4 can be built up after some time.

The program exploits a symmetry of the Hénon-Heiles equations of motion to obtain two trajectories from one set of initial conditions. This symmetry can be seen from the equations of motion which are invariant

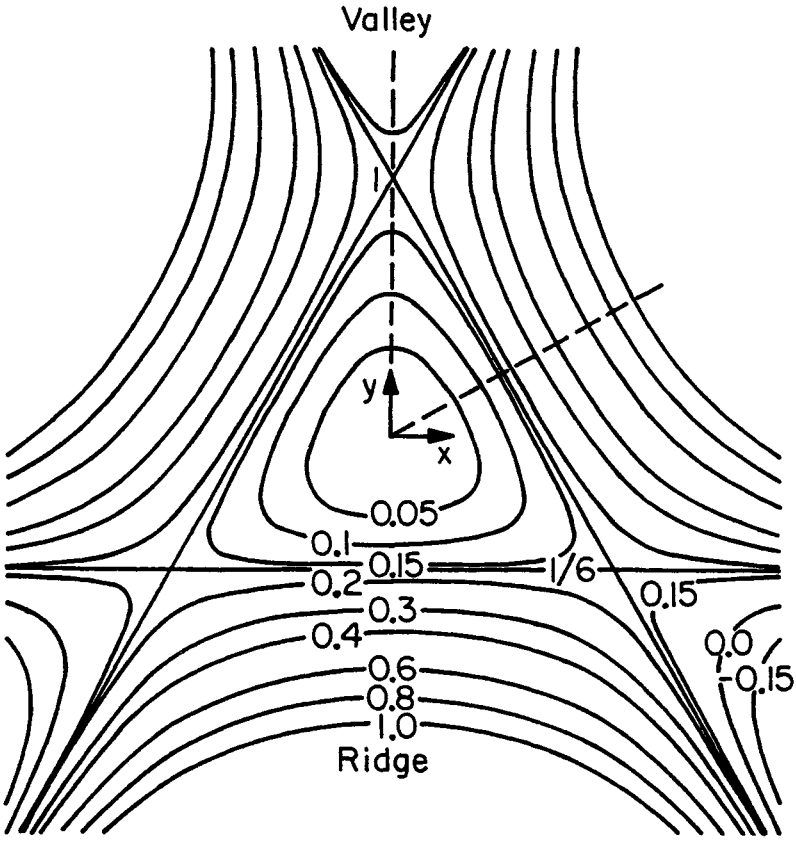


Figure 2.5 Equipotential contours of the Hénon-Heiles potential, Eq. (2.35).

if $x \to -x$ and $p_x \to -p_x$. Therefore from one trajectory we can trivially obtain a "reflected" trajectory by flipping the sign of both x and p_x at each point along the original trajectory. More specifically, if we are interested in creating the surface of section on the $x = 0$ plane we already have $x = -x$, and only the sign of p_x must be changed. Practically this means that every time we cross the $x = 0$ plane we take the corresponding y and p_y values to be on the surface of section, without consideration for the sign of p_x. If p_x is greater than zero it is a point from the initial trajectory, while if p_x is less than zero then that point belongs to the reflected trajectory.

The following exercises are aimed at improving your understanding of the physical principles and numerical methods demonstrated in this example.

■ **Exercise 2.7** One necessary (but not sufficient) check of the accuracy of the integration in Hamiltonian systems is the conservation of energy. Change the time step used and observe how this affects the accuracy. Replace the integration method by one of the other algorithms discussed in this chapter and observe the change in accuracy and efficiency. Note that we require an algorithm that is both accurate and efficient because of the very long integration times that must be considered.

■ **Exercise 2.8** Change the potential to a central one (function subroutine V) and observe the character of the (x, y) trajectories and surfaces of section for various initial conditions. Compare the results with Figure 2.2. and verify that the qualitative features don't change if you use a different central potential. Note that you will have to adjust the energy and scale of the potential so that the (x, y) trajectory does not go beyond the equilateral triangle of the Hénon-Heiles potential (Figure 2.5) or else the graphics subroutine will fail.

■ **Exercise 2.9** If the sign of the y^3 term in the Hénon-Heiles potential (2.35) is reversed, the Hamiltonian becomes integrable. Verify analytically that this is so by making a canonical transformation to the variables $x \pm y$ and showing that the potential is separable in these variables. Make the corresponding change in sign in the code and observe the character of the surfaces of section that result for various initial conditions at a given energy. (You should keep the energy below 1/12 to avoid having the trajectory become unbounded.) Verify that there are no qualitative differences if you use a different separable potential, say the harmonic one of Eq. (2.34).

■ **Exercise 2.10** Use the code to construct surfaces of section for the Hénon-Heiles potential (2.35) at energies ranging from 0.025 to 0.15 in steps of 0.025. For each energy, consider various initial conditions and integrate each trajectory long enough in time (some will require going to $t \approx 1000$) to map out the surface-of-section adequately. For each energy, see if you can find the elliptic fixed points, the tori (and tori of tori) around them, and the chaotic regions of phase space and observe how the relative proportions of each change with increasing energy. With some patience and practice, you should be able to generate a plot that resembles the schematic representation of Figure 2.4 around each elliptic trajectory.

Project II: The structure of white dwarf stars

White dwarf stars are cold objects composed largely of heavy nuclei and their associated electrons. These stars are one possible end result of the conventional nuclear processes that build the elements by binding nucleons into nuclei. They are often composed of the most stable nucleus, ^{56}Fe, with 26 protons and 30 neutrons, but, if the nucleosynthesis terminates prematurely, ^{12}C nuclei might predominate. The structure of a white dwarf star is determined by the interplay between gravity, which acts to compress the star, and the degeneracy pressure of the electrons, which tends to resist compression. In this project, we will investigate the structure of a white dwarf by integrating the equations defining this equilibrium. We will determine, among other things, the relation between the star's mass and its radius, quantities that can be determined from observation. Discussions of the physics of white dwarfs can be found in [Ch57], [Ch84], and [Sh83].

II.1 The equations of equilibrium

We assume that the star is spherically symmetric (i.e., the state of matter at any given point in the star depends only upon the distance of that point from the star's center), that it is not rotating, and that the effects of magnetic fields are not important. If the star is in mechanical (hydrostatic) equilibrium, the gravitational force on each bit of matter is balanced by the force due to spatial variation of the pressure, P. The gravitational force acting on a unit volume of matter at a radius r is

$$F_{\text{grav}} = -\frac{Gm}{r^2}\rho\,, \tag{II.1}$$

where G is the gravitational constant, $\rho(r)$ is the mass density, and $m(r)$ is the mass of the star interior to the radius r:

$$m(r) = 4\pi \int_0^r \rho(r')r'^2 dr'\,. \tag{II.2}$$

The force per unit volume of matter due to the changing pressure is $-dP/dr$. When the star is in equilibrium, the net force (gravitational plus pressure) on each bit of matter vanishes, so that, using (II.1), we have

$$\frac{dP}{dr} = -\frac{Gm(r)}{r^2}\rho(r)\,. \tag{II.3}$$

A differential relation between the mass and the density can be obtained by differentiating (II.2):

$$\frac{dm}{dr} = 4\pi r^2 \rho(r) \,. \tag{II.4}$$

The description is completed by specifying the "equation of state," an intrinsic property of the matter giving the pressure, $P(\rho)$, required to maintain it at a given density. Upon using the identity

$$\frac{dP}{dr} = \left(\frac{d\rho}{dr}\right)\left(\frac{dP}{d\rho}\right) ,$$

Eq. (II.3) can be written as

$$\frac{d\rho}{dr} = -\left(\frac{dP}{d\rho}\right)^{-1} \frac{Gm}{r^2}\rho \,. \tag{II.5}$$

Equations (II.4) and (II.5) are two coupled first-order differential equations that determine the structure of the star for a given equation of state. The values of the dependent variables at $r = 0$ are $\rho = \rho_c$, the central density, and $m = 0$. Integration outward in r then gives the density profile, the radius of the star, R, being determined by the point at which ρ vanishes. (On very general grounds, we expect the density to decrease with increasing distance from the center.) The total mass of the star is then $M = m(R)$. Since both R and M depend upon ρ_c, variation of this parameter allows stars of different mass to be studied.

II.2 The equation of state

We must now determine the equation of state appropriate for a white dwarf. As mentioned above, we will assume that the matter consists of large nuclei and their electrons. The nuclei, being heavy, contribute nearly all of the mass but make almost no contribution to the pressure, since they hardly move about at all. The electrons, however, contribute virtually all of the pressure but essentially none of the mass. We will be interested in densities far greater than that of ordinary matter, where the electrons are no longer bound to individual nuclei, but rather move freely through the material. A good model is then a free Fermi gas of electrons at zero temperature, treated with relativistic kinematics.

For matter at a given mass density, the number density of electrons is

$$n = Y_e \frac{\rho}{M_p} \,, \tag{II.6}$$

where M_p is the proton mass (we neglect the small difference between the neutron and proton masses) and Y_e is the number of electrons per nucleon. If the nuclei are all ^{56}Fe, then

$$Y_e = \frac{26}{56} = 0.464 ,$$

while $Y_e = \frac{1}{2}$ if the nuclei are ^{12}C; electrical neutrality of the matter requires one electron for every proton.

The free Fermi gas is studied by considering a large volume V containing N electrons that occupy the lowest energy plane-wave states with momentum $p < p_f$. Remembering the two-fold spin degeneracy of each plane wave, we have

$$N = 2V \int_0^{p_f} \frac{d^3p}{(2\pi)^3} , \qquad (II.7)$$

which leads to

$$p_f = (3\pi^2 n)^{1/3} , \qquad (II.8)$$

where $n = N/V$. (We put $\hbar = c = 1$ unless indicated explicitly.) The total energy density of these electrons is

$$\frac{E}{V} = 2 \int_0^{p_f} \frac{d^3p}{(2\pi)^3} (p^2 + m_e^2)^{1/2} , \qquad (II.9)$$

which can be integrated to give

$$\frac{E}{V} = n_0 m_e x^3 \epsilon(x) ; \qquad (II.10a)$$

$$\epsilon(x) = \frac{3}{8x^3}\left[x(1+2x^2)(1+x^2)^{1/2} - \log\left[x + (1+x^2)^{1/2}\right]\right] \qquad (II.10b)$$

where

$$x \equiv \frac{p_f}{m_e} = \left(\frac{n}{n_0}\right)^{1/3} ; \quad n_0 = \frac{m_e^3}{3\pi^2} . \qquad (II.10c)$$

The variable x characterizes the electron density in terms of

$$n_0 = 5.89 \times 10^{29} \text{ cm}^{-3} ,$$

the density at which the Fermi momentum is equal to the electron mass, m_e.

In the usual thermodynamic manner, the pressure is related to how the energy changes with volume at fixed N:

$$P = -\frac{\partial E}{\partial V} = -\frac{\partial E}{\partial x}\frac{\partial x}{\partial V} . \qquad (II.11)$$

Using (II.10c) to find

$$\frac{\partial x}{\partial V} = -\frac{x}{3V}$$

and differentiating (II.10a,b) results in

$$P = \frac{1}{3} n_0 m_e x^4 \epsilon' , \qquad (II.12)$$

where $\epsilon' = d\epsilon/dx$.

It is now straightforward to calculate $dP/d\rho$. Since P is most naturally expressed in terms of x, we must relate x to ρ. This is most easily done using (II.10c) and (II.6):

$$x = \left(\frac{n}{n_0}\right)^{1/3} = \left(\frac{\rho}{\rho_0}\right)^{1/3} ; \qquad (II.13a)$$

$$\rho_0 = \frac{M_p n_0}{Y_e} = 9.79 \times 10^5 Y_e^{-1} \text{ gm cm}^{-3} . \qquad (II.13b)$$

Thus, ρ_0 is the mass density of matter in which the electron density is n_0. Differentiating the pressure and using (II.12, II.13a) then yields, after some algebra,

$$\frac{dP}{d\rho} = \frac{dP}{dx}\frac{dx}{d\rho} = Y_e \frac{m_e}{M_p}\gamma(x) ;$$

$$\gamma(x) = \frac{1}{9x^2}\frac{d}{dx}(x^4\epsilon') = \frac{x^2}{3(1+x^2)^{1/2}} . \qquad (II.14)$$