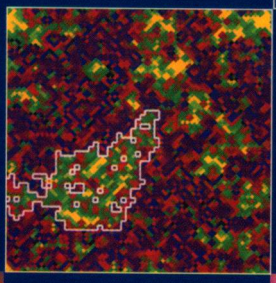
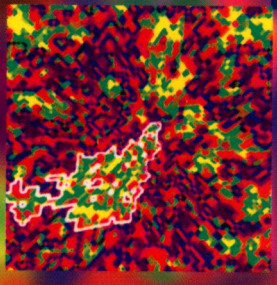
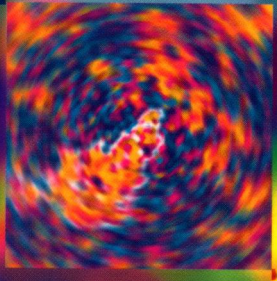
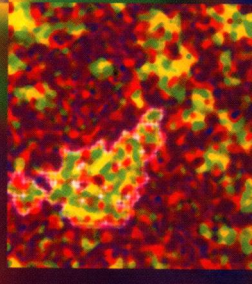
Computational Physics

J. M. THIJSSEN









Computational Physics

J. M. Thijssen
Delft University of Technology

CAMBRIDGE
UNIVERSITY PRESS

PUBLISHED BY THE PRESS SYNDICATE OF THE UNIVERSITY OF CAMBRIDGE
The Pitt Building, Trumpington Street, Cambridge, United Kingdom

CAMBRIDGE UNIVERSITY PRESS
The Edinburgh Building, Cambridge CB2 2RU, UK http://www.cup.cam.ac.uk
40 West 20th Street, New York, NY 10011–4211, USA http://www.cup.org
10 Stamford Road, Oakleigh, Melbourne 3166, Australia

First published 1999

Printed in the United Kingdom at the University Press, Cambridge

Typeset in Times

A catalogue record for this book is available from the British Library

Library of Congress Cataloguing in Publication data
Thijssen, J. M. (Joseph Marie). 1958–
Computational physics / J. M. Thijssen.
p. cm.
Includes bibliographical references and index.
ISBN 0 521 57304 1 – ISBN 0 521 57588 5 (pbk.)
1. Mathematical physics. 2. Condensed matter – Mathematics.
3. Physics – Data processing. I. Title
QC20.T445 1999
530.15–dc21 98–36458 CIP

ISBN 0 521 57304 1 hardback
ISBN 0 521 57588 5 paperback

Computational Physics

This book describes computational methods used in theoretical physics with emphasis on condensed matter applications.

Computational physics involves the use of computer calculations and simulations to solve physical problems. Following an overview of the wide variety of topics and algorithmic approaches studied in this book, the text explores quantum scattering with a spherically symmetric potential as a typical example of a computational physics problem. The next chapters concentrate on electronic structure calculations, presenting the Hartree–Fock and density functional formalisms, and band structure methods. Later chapters discuss molecular dynamics simulations and Monte Carlo methods in classical and quantum physics, with applications to condensed matter and particle field theories. Each chapter begins with an exposition of necessary fundamentals, describes the formation of sample programs and ends with problems addressing related analytical and numerical issues. Useful appendices on numerical methods and random number generators are included and the book contains extensive references.

Suitable for graduate students, this book bridges the gap between undergraduate-level physics and computational research. It is also a valuable reference for researchers in physics, chemistry, computer science, mathematics and biology. Although not essential, some knowledge of numerical analysis would be helpful in reading this book.

JOS THIJSSEN did his physics degree at the University of Nijmegen in the Netherlands. He then taught at secondary schools and developed educational software for university physics courses in Nijmegen. In 1990, he obtained his Ph.D. in statistical physics from the University of Nijmegen. He wrote his thesis on phase transitions in the two-dimensional, frustrated *XY* model. After obtaining his Ph.D., he spent five years as a lecturer in Nijmegen. During this period he set up an undergraduate course on computational physics, which formed the basis of the present book. From the end of 1995 until September 1998 he was a lecturer at the University of Cardiff, Wales. Since September 1998, he has been working on the application of multigrid and parallelisation methods to semiconductor process simulation at the DIMES Institute of the Delft University of Technology, in the Netherlands. His current research interests include polycrystalline growth, polymer folding, and electronic structure methods for large unit cells.

Preface

This is a book on computational methods used in theoretical physics research, with an emphasis on condensed matter applications.

Computational physics is concerned with performing computer calculations and simulations for solving physical problems. Although computer memory and processor performance have increased dramatically over the last two decades, most physical problems are too complicated to be solved without approximations to the physics, quite apart from the approximations inherent in any numerical method. Therefore, most calculations done in computational physics involve some degree of approximation. In this book, emphasis is on the derivation of algorithms and the implementation of these: it is a book which tells you how methods work, why they work, and what the approximations are. It does not contain extensive discussions on results obtained for large classes of different physical systems.

This book is not elementary: the reader should have a background in basic undergraduate physics and in computing. Some background in numerical analysis is also helpful. On the other hand, the topics discussed are not treated in a comprehensive way; rather, this book hopefully bridges the gap between more elementary texts[1–3] and specialised monographs and review papers on the applications described. The fact that a wide range of topics is included has the advantage that the many similarities in the methods used in seemingly very different fields could be highlighted. Many important topics and applications are however not considered in this book – the material presented obviously reflects my own expertise and interest.

I hope that this book will be useful as a source for intermediate and advanced courses on the subject. I furthermore hope that it will be helpful for graduates and researchers who want to increase their knowledge of the field.

Some variation in the degree of difficulty is inherent to the topics addressed in this book. For example, in molecular dynamics, the equations of motion of a collection of particles are solved numerically, and as such it is a rather elementary subject. However, a careful analysis of the integration algorithms used, the problem of performing these simulations in different statistical ensembles, and the problem of treating long range forces with periodic

boundary conditions, are much more difficult. Therefore, sections addressing advanced material are marked with an asterisk (*) – they can be skipped at first reading. Also, extensive theoretical derivations are sometimes moved to sections with asterisks, so that the reader who wants to write programs rather than go into the theory may use the results, taking the derivations for granted.

Aside from theoretical sections, implementations of algorithms are discussed, often in a step-by-step fashion, so that the reader can program the algorithms him- or herself. Suggestions for checking the program are included. In the exercises after each chapter, additional suggestions for programs are given, but there are also exercises in which the computer is not used. The computer exercises are marked by the symbol [**C**]; if the exercise is divided up into parts, this sign occurs before the parts in which a computer program is to be written (a problem marked with [**C**] may contain major parts which are to be done analytically). The programs are not easy to write – most of them took me a long time to complete! Some data-files and numerical routines can be found on `http://ectm.et.tudelft.nl/www/homepage/Thijssen/`.

The first person who suggested that I should write this book was Aloysio Janner. Thanks to the support and enthusiasm of my colleague and friend John Inglesfield in Nijmegen, I then started writing a proposal containing a draft of the first hundred pages. After we both moved to the University of Cardiff (UK), he also checked many chapters with painstaking precision, correcting the numerous errors, both in the physics and in the English; without his support, this book would probably never have been completed.

Bill Smith, from Daresbury Laboratories (UK), has checked the chapters on classical many particle systems and Prof. Konrad Singer those on quantum simulation methods. Simon Hands from the University of Swansea (UK) has read the chapter on lattice field theories, and Hubert Knops (University of Nijmegen, The Netherlands) those on statistical mechanics and transfer matrix calculations. Maziar Nekovee (Imperial College, London, UK) commented on the chapter on quantum Monte Carlo methods. I am very grateful for the numerous suggestions and corrections from them all. I am also indebted to Paul Hayman for helping me correcting the final version of the manuscript.

In writing this book, I have discovered that the acknowledgements to the author's family, often expressed in an apologetic tone as a result of the disruption caused by the writing process to family life, are too real to be disqualified as a cliché. My sons Maurice, Boudewijn and Arthur have in turn disrupted the process of writing in the most pleasant way possible, regularly asking me to show growing trees or fireworks on the screen of my PC, instead of the dull black-on-white text windows. Boudewijn and Maurice's professional imitation of their dad, tapping on the keyboard, and sideways reading formulae,

is promising for the future.

It is to my wife Ellen that I dedicate this book, with gratitude for her patience, strength and everlasting support during the long, and sometimes difficult time in which the book came into being.

Contents

1

Introduction

1.1 Physics and computational physics

Solving a physical problem often amounts to solving an ordinary or partial differential equation. This is the case in classical mechanics, electrodynamics, quantum mechanics, fluid dynamics and so on. In statistical physics we must calculate sums or integrals over large numbers of degrees of freedom. Whatever type of problem we attack, it is only in very few cases that analytical solutions are possible. In most cases we therefore resort to numerical calculations to obtain useful results. Computer performance has increased dramatically over the last few decades (see also chapter 14) and we can solve complicated equations and evaluate large integrals in a reasonable amount of time.

Often we can apply numerical routines (found in software libraries for example) directly to the physical equations and obtain a solution. We shall see, however, that although computers have become very powerful, they are still unable to provide a solution to most problems without approximations to the physical equations. In this book, we shall focus on these approximations, that is, we shall concentrate on the development of computational methods (and also on their implementation into computer programs). In this introductory chapter we give a bird's-eye perspective of different fields of physics and the computational methods used to solve problems in these areas. We give examples of direct application of numerical methods but we also give brief and heuristic descriptions of the additional theoretical analysis and approximations necessary to obtain workable methods for more complicated problems which are described in more detail in the remainder of this book. The order adopted in the following sections differs somewhat from the order in which the material is treated in this book.

1.2 Classical mechanics and statistical mechanics

The motion of a point particle in one dimension subject to a force F, depending on the particle's position x, and perhaps on the velocity $\dot{x}$ and on time t, is determined by Newton's equation of motion:

$$m\ddot{x}(t) = F[x(t), \dot{x}(t), t]. \tag{1.1}$$

The (double) dot denotes a (double) derivative with respect to time. A solution can be found for each set of initial conditions $x(t_0)$ and $\dot{x}(t_0)$ given at some time t_0. Analytical solutions exist for constant force, for the harmonic oscillator ($F = \kappa x^2/2$), and for a number of other cases. In section A.7.1.3 a simple numerical method for solving this equation is described and this can be applied straightforwardly to arbitrary forces and initial conditions.

Interesting and sometimes surprising physical phenomena can now be studied. As an example, consider the Duffing oscillator,[4] with a force given by

$$F = -\gamma\dot{x} + 2ax - 4bx^3 + F_0\cos(\omega t). \tag{1.2}$$

The first term on the right hand side represents a velocity-dependent friction; the second and third terms are the force a particle feels when it moves in a double potential well $bx^4 - ax^2$, and the last term is an external periodic force. An experimental realisation is a pendulum consisting of an iron ball suspended by a thin string, with two magnets below it. The pendulum and the magnets are placed on a table which is moved back and forth with frequency ω. The string and the air provide the frictional force, the two magnets together with gravity form some kind of double well, and, in the reference frame in which the pendulum is at rest, the periodic motion of the table is felt as a periodic force. It turns out that the Duffing oscillator exhibits *chaotic behaviour* for particular values of the parameters γ, a, b, F_0 and ω. This means that the motion itself looks irregular and that a very small change in the initial conditions will grow and result in completely different motion. Figure 1.1 shows the behaviour of the Duffing oscillator for two nearly equal initial conditions, showing the sensitivity to these conditions. Over the last few decades, chaotic systems have been studied extensively. A system which often behaves chaotically is the weather: the difficulty in predicting the evolution of chaotic systems causes weather forecasts to be increasingly unreliable as they look further into the future, and occasionally to be drastically wrong.

Another interesting problem is that of several particles, moving in three dimensions and subject to each other's gravitational interaction. Our solar

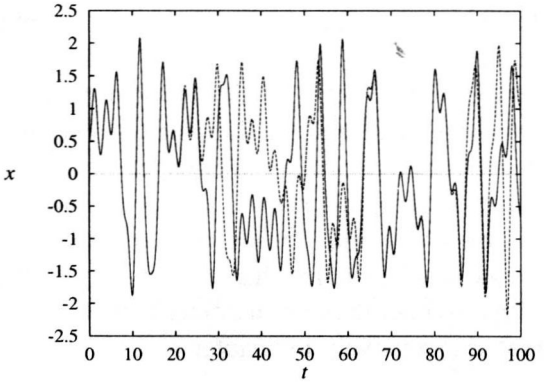


Figure 1.1: Solution of the Duffing oscillator. Parameters are $m = 1$, $a = {}^1/_4$, $b = {}^1/_2$, $F_0 = 2.0$, $\omega = 2.4$, $\gamma = 0.1$. Two solutions are shown: one with initial position $x_0 = 0.5$, the other with $x_0 = 0.5001$ ($\dot{x}_0 = 0$ in both cases). For these nearly equal initial conditions, the solutions soon become uncorrelated, showing the difficulty in predicting the time evolution of a chaotic system.

system is an example. For the simplest nontrivial case of three particles (for two particles, Newton has given the analytical solution), analytical solutions exist for particular configurations, but the general problem can only be solved numerically. This problem is called the *three-body problem* (N-body problem in general). The motion of satellites orbiting in space is calculated numerically using programs for the N-body problem, and the evolution of galaxies is calculated with similar programs using a large number of test particles (representing the stars) – millions of particles can be treated using a combination of high-end computers and clever computational methods which will be considered in chapter 8. Electrostatic forces are related to gravitational forces, as both the gravitational and the electrostatic (Coulomb) potential have a $1/r$ form. The difference between the two is that electrostatic forces can be repulsive and attractive, whereas gravitational forces are always attractive.

Neutral atoms interact via a different potential: they attract each other weakly unless they come too close – then they start repelling each other. The problem of many interacting atoms and molecules is a very important subfield of computational physics: it is called *molecular dynamics*. In molecular dynamics, the equations of motion for the particles are solved straightforwardly using numerical algorithms similar to those with which a Duffing oscillator is analysed, the main difference being the larger number of degrees of freedom in molecular dynamics. The aim of molecular dynamics simulations is to extract

data for gases, liquids and solids (and systems in other phases, like liquid crystals). An important result is the equation of state: this is the relation between temperature, number of particles, pressure and volume. Also, the microscopic structure as exhibited by the pair correlation function, which is experimentally accessible via neutron scattering, is an interesting property which can be determined in simulations. There are, however, many problems and pitfalls associated with computer simulations: the systems which can be simulated are always much smaller than realistic systems, and simulating a system at a predefined temperature or chemical potential is nontrivial. All these aspects will be treated extensively in chapter 8.

1.3 Stochastic simulations

In the previous section we have explained how numerical algorithms for solving Newton's equations of motion can be used to simulate liquids. The particles are moved around according to their mechanical trajectories which are governed by the forces they exert on each other. Another way of moving them around is to displace them in a random fashion. Of course this must be done in a controlled way, and not every move should be allowed, but we shall see in chapter 10 that it is possible to obtain information in this way similar to that obtained from molecular dynamics. This is an example of a *Monte Carlo* method – procedures in which random numbers play an essential role. The Monte Carlo method is not suitable for studying dynamical physical quantities such as transport coefficients, as this method uses an artificial dynamics to simulate many-particle systems.

Random number generators can also be used in direct simulations: some process of which we do not know the details is replaced by a random generator. If you simulate a card game for example, the distribution of the cards among the players is done using random numbers. An example of a direct simulation in physics is diffusion limited aggregation (DLA), which describes the growth of dendritic clusters (see figure 1.2). Consider a square lattice in two dimensions. The sites of the lattice are either occupied or unoccupied. Initially, only one site in the centre is occupied. We release a random walker from the boundary of the lattice. The walker moves over the lattice in a stepwise fashion. At each step, the walker moves from a site to one of its neighbour sites, which is chosen at random (there are four neighbours for each site in the interior of the lattice, and two or three at the angles and at the sides respectively). If the walker arrives at a neighbouring site of the central site which is occupied, it sticks there, so that a two-site cluster is formed. Then a new walker is released from the boundary.