Scientific Programming 20 (2012) 355–358
DOI 10.3233/SPR-2012-0354
IOS Press

Book Review

Adrian Brown
SETI Institute
E-mail: abrown@seti.org

Abstract. The second edition of *Computational Physics* by Rubin Landau, Manuel Paez and Cristian C. Bordeianu (published by Wiley) hit bookshelves in 2007, and is steadily making its way into physics classrooms across the United States. Your reviewer first encountered the book at the Stanford Campus Book Store, where it is presumably being put to good use by students and faculty.

The first edition was published in 1997 with Landau and Paez as authors [*Computational Physics*, 1st edn, Wiley, New York, 1997]. Rubin Landau is a very experienced computational physicist and staff member of the Oregon State University in Corvallis, where he directs the *Computational Physics* for Undergraduates course and teaches using this book. Landau is an incredibly active teacher to put it mildly – the code for the book is supplied in an accompanying DVD and he also posts applets and video lectures for his courses on his web page (http://www.physics.orst.edu/~rubin/).

Computational Physics: Problem Solving with Computers, 2nd edition, by Rubin Landau, Manuel Paez and Cristian C. Bordeianu, July 2007, ISBN: 978-3-527-40626-5, 616 pp., paperback.

1. Aim of the book

The subtitle of the book is 'Problem solving with computers' and this really encompasses what this book is all about. Landau has a number of other books to his name at this time, including a truly 'introductory' book (*A First Course in Scientific Computing*, Princeton, NJ, 2005 [2]) but this book aims to take physics students who are interested in computers from zero (how a computer stores digits is covered in Chapter 1) to hero (a method to solve the Lippman–Schwinger equation is presented in Chapter 30).

Your reviewer was very impressed with the quality of code and the subjects covered by the book. The book is really designed with the undergraduate in mind, and I found it was possible to briefly review these topics, but the concluding chapters covered very novel and interesting material even for an experienced programmer.

Unfortunately, there are a large number of typographical errors that are throughout the book – this really does get quite annoying because the frequency is almost 1 per page towards the end of the book. It seems as though they are omitted words chiefly, but they do not hinder the learning of the material – just give an uneasy feeling of incompleteness.

2. Style of the book

The book is written in a very relaxed and conversational style, making it very easy to dive into and turn through the pages. Some Mathematicians may see this as a lack of rigour – I think this is a smart tactic to encourage undergraduates to delve into the book and pick up information quickly and easily.

Code is embedded within the text, which could be criticized as taking up space, but I believe it places the correct amount of emphasis on the importance of algorithms and code in computational physics.

In some cases however, the informal style of writing almost feels like it is taken verbatim from a set of class notes – this is particularly the case when problems are introduced throughout the text, and it is difficult to discern where the instruction stops and the problem set begins. There are great problems that all require numerical analysis throughout each chapter, and the reader is encouraged to 'explore' topics beyond the book's scope, which is a boon for the teaching environment.

Some of the large chapters (e.g., Chapter 15) are broken into Units I and II, presumably for ease of teaching. Frustratingly, on p. 357, a promise is made to cover the Finite Element Method in 'Unit II' however this is not covered anywhere in the book. This is a different mistake from the reference in Chapter 25 to 'Unit II' on p. 379 that clearly refers to Chapter 26 (there is no Unit I or II in Chapter 25). Page 111 says that Chapter 28 covers simulated annealing, but the words 'simulated annealing' do not appear in Chapter 28 although the Metropolis algorithm they describe does appear to be simulated annealing.

The coding style of the examples focuses on Java (which this reader appreciates immensely). The code relies on the standard Java libraries with the addition of JAMA, the matrix manipulation library. Throughout the text the reader is given coding tips that should be invaluable to the novice, and good reminders for those who are using the book to assist them in a coding problem.

The author's attitude towards the *Numerical Recipes* book is interesting. The authors state on page 89 that "Although we prize the book... [6] we cannot recommend taking subroutines from it. They are neither optimized nor documented for easy and stand-alone use".

3. Outline of the book

Chapters 1–3 of the book are extremely elementary and clearly designed to ease in those first year undergraduates who are not familiar with computers. The topics of machine precision, storage of double precision and single precision numbers, round off errors, propagation of errors in loops. The one piece of novel discussion is the recursion relation for computing spherical Bessel Functions, which is a very nice addition. It acts as a prototype for testing the ideas of numerical round off errors and comparison of upward and downward recursion at the end of the chapter.

Chapter 4 is a very short chapter – it introduces object oriented programming and uses the example of Newton's mechanics as a test subject.

Chapter 5 introduces numerical integration – a standard for these type of texts. The Trapezoidal Rule, Simpson's Rule and Gaussian Quadrature are all introduced. The problem set in this chapter is particularly good because it explores and explains the expected errors for each type of numerical integration. Chapter 6 is a very short introduction to forward, backward and central-difference methods. Chapter 7 introduces the another real physical system (the Quantum states in a square well). The equations are simply stated, for the background the reader is referred to a 'Quantum mechanics' book. Bisection search, Newton–Raphson, and Backtracking are introduced.

Chapter 8 is a large chapter that introduces matrix manipulation with the Physics problem of multidimensional Newton–Raphson searching. Some discussion of publicly available matrix libraries is supplied, and JAMA is used to sole an eigenvalue problem. Chapter 9 continues the exploration of data fitting – Langrange Interpolation polynomials, cubic splines, and least squares fitting are introduced. Tips on using LAPACK with C programs are given.

Chapter 10 gets into the probability side of the house with an immediate introduction to random number generation and methods for assessing the quality of a string of random numbers. Chapter 11 is a natural continuation into Monte Carlo applications. A random walk is implemented, then a radioactive decay problem. Monte Carlo Integration is then introduced, including discussions of multidimensional and nonuniform randomness (exponential and Gaussian weighting algorithms are discussed). The chapter closes with the von Neumann method for arbitrary probability distributions. Chapter 12 moves to thermodynamic simulations, introducing a superficial slice of statistical mechanics to motivate the Metropolis algorithm. The Ising model for ferromagnetism is the supplied physical test case.

Chapter 13 gets back into computer science – memory and CPU terms are introduced at the beginner level. Chapter 14 addresses computer performance and optimization. Some discussion of caching and paging is provided.

Chapter 15 leads in to differential equations. Discretization, Euler's rule, Runge Kutta, choice of rk4 vs. rk45 [6] are discussed. A nonlinear oscillator is used as the physical test case. The addition of friction, time dependent forces and beats are presented as problems. The second unit of Chapter 15 proposes a list of problems without real solutions – projectile motion with drag and planetary motion.

Chapter 16 introduces Quantum Eigenvalues via ODE Matching. This is quite an advanced Physics problem, and although I appreciated it for an examination of how one might solve this problem, it is quite a brief treatment (10 pages, 25% of which is just code). It is hard to imagine the reader of this book remembering this chapter in amongst all the other material.

Chapter 17 introduces Fourier analysis, the DFT and aliasing are introduced. Form factor/structure function is introduced in the Exploration section.

Chapter 18 introduces nonlinear dynamics, starting with the Logistic Map problem (aka Predator–Prey or Lotka–Volterra problems) and discusses attractors, chaos and bifurcation diagrams. Chapter 19 continues the nonlinear theme by looking at the chaotic behavior of a forced pendulum. Mode locking and phase space plots are introduced. Chapter 20 continues the theme by looking at fractals – the Sierpinsky Gasket is implemented, its fractal dimension is estimated, ferns and plants are implemented, the coastline length is estimated, ballistic deposition and globular cluster (diffusion limited aggregation) algorithm are implemented.

Chapter 21 introduces parallel computing terminology. Chapter 22 introduces how to run programs using the message passing interface (MPI) running on a Beowulf cluster. These two chapters occupy 38 pages but there is no physics in them – it is possible they might be of use to students at some stage.

Chapter 23 introduces the finite-difference method for electrostatics. A discussion of elliptic, parabolic and hyperbolic PDEs and Dirichlet, Neumann and Cauchy boundary conditions is given. Relaxation by Jacobi and Gauss–Siedel method and successive over relaxation are introduced. The physical example used throughout the chapter is the Laplace equation governing the charge on two parallel plates.

Chapter 24 warms up with a look at the heat equation. Time is introduced into the problem, and the leap frog method is used to handle the extra variable. The von Neumann stability assessment is introduced.

Chapter 25 moves to the wave equation – again the leap-frog technique is used and the von Neumann stability assessment is used and the Courant condition is also mentioned. Extensions are made into including friction, variable tension and density. Two-dimensional waves are then examined in some detail – an oscillating membrane is given as the physical example.

Chapter 26 moves to one of the more novel contributions of the book – a discussion of solitons and the Kortweg–de Vries (KdV) equation. The chapter begins with modifying the pendulum to become nonlinear by including a dispersion term – envisioned as a number of pendulums connected via a torsion bar. The Sine two-dimensional Gordon equation is introduced and shown to describe soliton waves on a membrane – a possible model for an elementary particle. The second half of the chapter addresses one dimensional soliton waves described by the KdeV equation. The problem is physically motivated by shallow water waves in canals. The analytic solution and finite-difference numerical solution are presented and an extension to crossing solitons is discussed.

The final four chapters of the book discuss quantum physics motivated problems. Chapter 27 introduces the time dependent Schroedinger equation, and presents a numerical algorithm using staggered real/imaginary leap-frog technique. This is used to resolve the motion of a wave packet inside a harmonic oscillator potential n one-dimension. The chapter closes with extensions to two dimensions by using a two-interval leap frog technique, however the rationale for the different technique is left in an external reference.

Chapter 28 introduces quantum paths for Functional integration using Green's functions or propagators (Feynman's Space–Time technique). This chapter absolutely flies through the background material (perhaps a book could be devoted to this) and the solution is provided by a Monte Carlo technique coupled with a Boltzmann distribution of energy. This chapter is really too brief to expect that the undergraduate will retain any of the explanations, however it is nice to have the code presented and explained even at a perfunctory level.

Chapter 29 is a very short chapter (7 pages) on solving quantum bound states via integral equations. This really does not give enough space to get the physics across. The Schrödinger equation is put into an integro-differential form and then linearized using Gauss quadrature and solved by using an eigenvalue matrix technique. The analytic solution of the delta-shell potential is given for comparison to the numerical code.

The last chapter, Chapter 30, addresses the solution of the Lippmann–Schwinger quantum scattering equation via integral equation techniques. There is a well-detailed discussion of avoiding the singularity in the equation using the Cauchy principal value prescription that could serve as a model for further additions to Chapters 28 and 29. The delta shell potential implementation is used again for the comparable analytic solution.

An appendix is given that briefly introduces the reader to the PtPlot graphing package [7].

4. Who will this book appeal to and why?

This book is clearly written with the undergraduate audience in mind, and undergraduate computational science, physics and mathematics students are all likely to benefit from reading this book. Because of the introductory nature of Chapters 1–3 some more experienced readers may feel the book is too basic, however the chapters on solving w wide range of differential equations may be of use to the more experienced practitioner.

5. Conclusions

Rubin Landau and his co-authors have put together a nice introduction to programming and computer science with a strong emphasis on underlying physics problems in their latest book *Computational Physics*. The writing is brisk and will be appreciated by those who are interested in quickly bringing numerical analysis into their domain of expertise. For those requiring more detailed mathematical expositions, this book may serve as a stepping stone to more complex texts such as Golub and Van Loan [1] or (despite the wishes of Landau) the Numerical Recipes text [6].

References

[1] G.H. Golub and C.F. Van Loan, *Matrix Computations*, Johns Hopkins Univ. Press, Baltimore, MD, 1996.

[2] R. Landau, *A First Course in Scientific Computing*, Princeton Univ. Press, Princeton, NJ, 2005.

[3] R. Landau and M. Paez, *Computational Physics*, 1st edn, Wiley, New York, 1997.

[4] R. Landau, M. Paez and C.C. Bordeianu, *Computational Physics*, 2nd edn, Wiley, Weinheim, 2007.

[5] R. Landau webpage, http://www.physics.orst.edu/~rubin/.

[6] W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery, *Numerical Recipes*, 3rd edn, Cambridge Univ. Press, Cambridge, 2007.

[7] PtPlot webpage, ptolemy.eecs.berkeley.edu/java/ptplot/.