Review of[1]
Classical and Quantum Computing with C++ and Java Simulations
Authors: Yorick Hardy and Willi–Hans Steeb
Publisher: Birkhäuser Verlag, 2001
$77.95, Paperback

Reviewer: Nick Papanikolaou
(Dept. of Computer Science,
University of Warwick, U.K.)

1 Introduction

This textbook is an almost encyclopædic tome on a range of topics mainly within the undergraduate computer science curriculum. Its coverage is very broad, starting from boolean algebra and proceeding all the way to implementations of quantum computers. The selection and ordering of topics is somewhat arbitrary, and a book could be written about the subject matter of nearly every chapter. The most distinctive feature of this text is its coverage of quantum computing. However, while the title may suggest a balanced treatment of classical and quantum computing, only 30% of the book's pages are devoted to the latter. One may wonder what the intended readership is, since there are several good texts devoted to quantum computing, and many many more on the topics covered in the other 400 pages.

2 Coverage

The book is divided into two parts; Part I ('Classical Computing') consists of 15 chapters. Part II, 'Quantum Computing,' consists of 8, generally shorter, chapters. What follows is a summary of the topics covered.

Chapter 1: Algorithms. The first chapter introduces some basic terminology used in the analysis of algorithms, as well as mathematical induction. Euclid's algorithm is used as an example of a classical algorithm; the process of *simulated annealing* is used as an example of a randomized algorithm. Some simple programs in C++ and Java are given.

Chapters 2–7: Boolean Algebra, Sequential and Combinational Circuits. Chapter 2 presents the laws of boolean algebra, De Morgan's theorem, Karnaugh maps and the Quine–McCluskey method for circuit simplification. This is followed by a chapter on number representation, discussing implementations of integers and real numbers. The next two chapters present examples of boolean circuits such as adders and programmable gate arrays, as well as latches. Synchronous circuits take up a chapter of their own.

Chapter 8: Recursion. This chapter discusses recursion in general, and illustrates the technique with examples (Towers of Hanoi, integer powers, quicksort and more) and sample programs. The wavelet transform is discussed; backtracking is illustrated through the 8–queens problem. Implementation of recursion with and without stacks is also covered.

Chapter 9: Abstract Data Types. The book presents the linked list, stack and tree ADTs in a single chapter. Most of the chapter consists of C++ code which implements these ADTs. The explanation of abstract data types is particularly brief, and is only useful as a reference for programmers.

Chapter 10: Error Correction and Detection. The tenth chapter treats Hamming codes and Shannon's theorem. Some useful C++ code for calculating Hamming codes is given, and there is also a Java program for computing weighted checksums.

Chapter 11: Cryptography. The chapter on cryptography is oriented towards programmers, with sample code for transposition and substitution ciphers and the RSA cryptosystem. Relevant background material is discussed only briefly.

Chapter 12: Finite State Machines. Chapter 12 covers finite automata; apart from general background material, attention is paid to Moore machines, Mealy machines and Turing machines. A C++ listing of a simulator for a Turing machine is given.

Chapter 13: Computability. In a mere 10 pages, the authors describe Church's thesis, Gödel's theorem and NP–completeness.

Chapter 14: Neural Networks. The fourteenth chapter is a good, readable presentation of neural networks, detailing the McCulloch–Pitts neuron, the perceptron and its learning algorithm, the multilayer perceptron and back–propagation. Coverage of neural networks is significantly more detailed than that of material in previous chapters. Sample code is provided for simulating networks and back–propagation.

Chapter 15: Genetic Algorithms. This chapter is one of the stronger points of the book, as it deals with an area of considerable academic interest. It discusses the basic, sequential genetic algorithm, the Gray code, Markov chain analysis, and maxima of one- and two–dimensional maps. C++ and Java code for genetic algorithms and related applications (e.g. knapsack and traveling salesman problems) is given.

Part II of the book starts with Chapter 16, which introduces the mathematics of quantum theory.

Chapters 17–18: Basics of Quantum Computing and Quantum Measurement. Both of these chapters tackle fundamental aspects of quantum computation and quantum information. Qubits and quantum registers are introduced, and so are quantum gates and circuits. Quantum measurement is discussed in a chapter of its own, along with certain interpretations (e.g. the Copenhagen interpretation, Everett's many worlds interpretation) of the process. SymbolicC++ code for simulations of quantum circuits are provided, and are generally useful and helpful for practically oriented computer scientists.

Chapters 19–20: Quantum State Machines and Teleportation. By analogy to the material in the chapter on finite state machines, the authors have devoted a few pages (Chapter 19) to quantum automata and quantum Turing machines. The quantum teleportation protocol, which allows the transfer of a single qubit using only a classical channel and a pair of entangled particles, is covered in a short chapter.

Chapter 21: Quantum Algorithms. This chapter presents, in a practical way, the most well–known quantum algorithms: Deutsch's algorithm, Simon's algorithm, Shor's algorithm, and Grover's algorithm. These algorithms are the most important component of the theory of quantum computing –in our view– and deserve a clear, accurate treatment. The account given in this book is readable but brief; the interested reader will have to resort to more specialized texts [6, 2] for explanations and details. The dense coding scheme, and the BB84 protocol for quantum key distribution, are also covered in this chapter. This is misleading as neither of these is regarded as a quantum *algorithm* in the accepted sense of the term.

Chapters 22–23: Quantum Information Theory and Quantum Error Correction Chapter 22 presents a selection of results from quantum information theory, such as measures of entanglement and the Holevo bound. The subsequent chapter deals with the 5–, 7–, and 9–qubit error correction codes. The treatment of all these topics is concise.

Chapter 24: Quantum Hardware This final chapter details proposed physical implementations of quantum computational devices, including cavity QED and NMR spectroscopy.

A list of websites related to quantum computation and quantum information is provided in the form of Chapter 25, and this is followed by bibliography and index.

3 Strengths

This book is essentially a compendium of techniques and computer programs usually dealt with in undergraduate computing courses. It is handy as a reference, since it covers a wide range of topics concisely. Its coverage of genetic algorithms, and gene expression programming in particular, is novel and useful for beginning researchers. The material on quantum computing has never appeared in a general computer science text before, and serves as a convenient summary of the basics; it is quite brief compared to other texts. The C++ and Java code provided is useful in practice, and helps to illustrate the techniques discussed.

4 Weaknesses

This textbook has two particular weaknesses, in our view. On one hand, the book covers far too many topics, and does little justice to most. On the other hand, while the title may suggest a balanced treatment of classical and quantum computation theory, the book devotes much less space to quantum computing and quantum information.

This book scores very highly on breadth of coverage, but at the cost of depth. Furthermore, the amount of space devoted to each topic is disproportionate. Topics of especial importance in computer science, such as computability and complexity, boolean algebra and cryptography (discussed in 8 pages, 23 pages and 6 pages respectively) receive far less attention than neural networks (35 pp.) and genetic algorithms (89 pp.). Clearly, an element of balance is missing.

The bibliography contains significantly more items related to quantum computation and quantum information than items related to classical computer science. This mismatches the proportion of material in the book on classical and quantum computing.

To address the question of the book's intended audience, we resort to the book's preface, where we are told that:

> "Scientific computing is not numerical analysis, the analysis of algorithms, high performance computing or computer graphics. It consists instead of the application of these fields and others to craft solution strategies for applied problems. It is the original application area of computers and remains the most important. [...] More and more universities introduce a Department of Scientific Computing or a Department of Computational Science. [...] This book can serve as a text book in Scientific Computing. It contains all the techniques (including quantum computing)."

The above fragment contains a couple of fallacies, if we are to accept the conventional usage of the term *scientific computing.* Textbooks and university curricula for the subject known as scientific computing, almost without exception, contain material traditionally associated with numerical analysis. What is more, the last sentence in the fragment seems to suggest that quantum computing is a 'technique' which belongs to this subject; again, this depends on your definition of 'scientific computing.'

So let's make this clear: *scientific computing* is a term that normally refers to the study of such topics as linear and nonlinear equations, least eigenvalues, optimization, integration, interpolation, ordinary and partial differential equations, and fast Fourier transforms. This is not the subject of Hardy and Steeb's text at all. To drive this point home, I will quote from [4], a highly regarded text on scientific computing:

> "The subject of this book is traditionally called numerical analysis. Numerical analysis is concerned with the design and analysis of algorithms for solving mathematical problems that arise in many fields, especially science and engineering. For this reason, numerical analysis has more recently also become known as scientific computing."

The authors seem to suggest that a 'Department of Scientific Computing' and a 'Department of Computational Science' are one and the same. Computer Science departments generally do not confine themselves to the study of numerical analysis, no matter what they are called. The distinction has been made particularly clear here at the University of Warwick, which actually has both a 'Department of Computer Science'[2] *and* a 'Centre for Scientific Computing'[3].

Putting the semantics of the term 'scientific computing' aside, we return to the question of this book's intended audience. This book is very usable as a reference on a number of topics, and someone with a mild interest in quantum computing may consider it a handy addition to their bookshelf. Apart from quantum computing, the book has good chapters on neural networks and genetic algorithms, but researchers in any one of these fields will only use it to recap on the basics, while referring to more specialized texts for details. Undergraduates may find some use for the C++ and Java code, but the book does not help as a text for a specific course. That is why, in our view, the readership for this book is difficult to determine.

---

[2] See `http://www.dcs.warwick.ac.uk/`

[3] See `http://www2.warwick.ac.uk/fac/sci/csc/`

Finally, we should point out that C++ and Java code listings are often several pages long; we feel that printing code fully in the book is not necessary, since anyone who is interested in the code can download it off the web and peruse it on his/her own machine.

5 Conclusions

Our criticisms of this book do not imply that its authors have not made a considerable effort; indeed, to cover such a wide range of topics, especially in a short amount of space, requires a lot of skill. We express the hope that a future edition of the book will address some of the issues raised here, and reach a wider audience; also, we welcome the initiative to present the subject of quantum computation and quantum information in a non–specialized text. Hardy and Steeb have already produced a brilliant little book [8] with solved problems in quantum computing and quantum information, which we highly recommend.

References

[1] Gay, S., Nagarajan, R., and Papanikolaou, N. Probabilistic model–checking of quantum protocols. Quantum Physics Repository Preprint quant-ph/0504007, available at `www.arxiv.org`.

[2] Gruska, J. *Quantum Computing.* McGraw–Hill International, 1999.

[3] Hardy, Y., and Steeb, W.-H. *Classical and Quantum Computing With C++ and Java Simulations.* Birkhäuser Verlag, 2001.

[4] Heath, M. *Scientific Computing*, 2nd ed. McGraw–Hill, 2001.

[5] Nagarajan, R., Papanikolaou, N., Bowen, G., and Gay, S. An automated analysis of the security of quantum key distribution. CoRR Preprint cs.CR/0502048, available at `www.arxiv.org`.

[6] Nielsen, M. A., and Chuang, I. L. *Quantum Computation and Quantum Information.* Cambridge University Press, 2000.

[7] Papanikolaou, N. Introduction to quantum cryptography. *ACM Crossroads Magazine 11.3* (Spring 2005 Issue).

[8] Steeb, W.-H., and Hardy, Y. *Problems & Solutions in Quantum Computing & Quantum Information.* World Scientific, 2004.