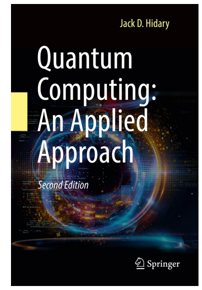
Review of [1]



Quantum Computing: An Applied Approach, 2nd ed.
by **Jack D. Hidary**

Springer Nature Switzerland, 2021
422 pages, Hardcover, $37.99, eBook, $29.99

Review by **James V. Rauff**
(`jrauff@millikin.edu`)
Department of Mathematics and Computational Sciences
Millikin University

1 Overview

Quantum Computing: An Applied Approach is an introductory textbook on quantum computing. The book features an extensive review of the mathematics underpinning quantum computing, a discussion of the classic algorithms of quantum computing, and implementations of the algorithms using Cirq, an open-source Python framework for Noisy Intermediate Scale Quantum (NISQ) algorithms. The mathematical chapters contain intertextual exercises with solutions. The book relies upon an associated GitHub site for updated code, additional exercises, and other resources.

2 Summary of Contents

The text consists of three parts comprising 15 chapters. Some of these chapters contain intertextual exercises with solutions.

Part I. Foundations

Chapter 1. Superposition, Entanglement and Reversibility.
A short outline of the basic elements of quantum physics relevant to quantum computing is given in this chapter. The topics herein include superposition, entanglement, the Born Rule (relating the amplitude of a quantum state to the probability of that state resulting after measurement), Schrödinger's equation, and reversibility. There are no exercises in the textbook, but some problems are posed in the GitHub resources.

Chapter 2. A Brief History of Quantum Computing.
This very short chapter highlights the contributions of Feynman, Deutsch, Vazirani, Bernstein, Simon, Shor, and Grover. It concludes with DiVincenzo's criteria for a quantum computer. Some problems for this chapter may be found in the GitHub resources.

Chapter 3. Qubits, Operators and Measurement.

This chapter lays the groundwork for the quantum algorithms to be presented later. Here the author defines a qubit and introduces Dirac notation along with quantum circuit diagrams. The fundamental quantum operators/gates (X, Y, Z, $R\varphi$, H, CNOT, CZ, Fredkin, and Toffoli) are explained and given in matrix form and circuit diagrams. A short description of the Bloch sphere is included. No exercises are given in the textbook, but there are several in the GitHub resources.

Chapter 4. Complexity Theory.

This very short chapter sketches the differences between the classical complexity classes P, NP, BPP and the quantum classes BQP, EQP, and QMA. A few problems are available in the GitHub resources.

Part II. Hardware and Applications

Chapter 5. Building a Quantum Computer.

This chapter discusses the leading paradigms of quantum computing hardware. The author views quantum processing units (QPU) as being used in combination with classical CPUs. Nuclear magnetic resonance (NMR) devices, nitrogen-vacancy (NV) center-in-diamond, photonics, and trapped ion approaches are included. There are two problems available in the GitHub resources.

Chapter 6. Development Libraries for Quantum Computer Programming.

A brief presentation of some quantum computing development libraries is the subject of this chapter. Included are Cirq (Google), Qiskit (IBM), Forest (Rigetti), and QDK (Microsoft). Code snippets for each library are presented. There are no exercises provided for this chapter.

Chapter 7. Teleportation, Superdense Coding and Bell's Inequality.

The eponymous quantum circuits are explained and implemented in Cirq. There are no exercises provided for this chapter.

Chapter 8. The Canon: Code Walkthrough.

The content of this chapter would probably form the heart of any beginning quantum computing course. It includes a description and Cirq implementation of the Deutsch-Jozsa algorithm, the Bernstein-Vazirani algorithm, Simon's problem (code on GitHub, but not in the text), the quantum Fourier transform, Shor's algorithm, and Grover's algorithm. The most detailed discussion is focused on Shor's algorithm, during which some exercises (with solutions) are given. Additional exercises are provided in the GitHub resources.

Chapter 9. Quantum Computing Methods.

Several quantum computing programs that can be run on NISQ processors are discussed in this chapter. Included are the variational quantum eigensolver, an application to quantum chemistry, the quantum approximate optimization algorithm (applied to the problem of computing the expectation of the cost Hamiltonian), quantum neural networks, quantum phase estimation, the HHL algorithm for solving linear systems of equations, a quantum random number generator, and quantum walks. All are implemented in Cirq. Some exercises are provided in the GitHub resources.

Chapter 10. Applications and Quantum Supremacy.

Quantum supremacy refers to a computational task that can be efficiently performed on a quantum computer beyond the capabilities of a classical supercomputer. This chapter discusses some of the tasks used in demonstrating quantum supremacy. These include random circuit sampling and quantum error correction. There are no exercises provided for this chapter.

Part III. Toolkit

Chapters 11-14 of Part III provide a review of the mathematics of quantum computing. Each chapter has intertextual exercises. These chapters are referred to throughout the first ten chapters of the text. They are designed to provide the definitions, theorems, and techniques needed to write quantum computing algorithms. Here I will simply list the topics appearing in each chapter.

Chapter 11. Mathematical Tools for Quantum Computing I.
Basic vector operation (dot product, norm, etc.), complex numbers, inner product, polar form of complex number, matrices (multiplication, addition, transpose, conjugate), and tensor products.

Set theory, Cartesian product, functions, relations, composition, linear transformations, vector space, subspace, span, linear independence, bases, orthonormal bases, Abelian group, and fields.

Chapter 12. Mathematical Tools for Quantum Computing II.
Matrices associated with linear transformations, determinants, matrix inversion, eigenvectors, eigenvalues, Kronecker delta function, Hermitian operators, unitary operators, direct sum, tensor product, Hilbert space, and a summary of relationship between quantum computing and linear algebra.

Chapter 13. Mathematical Tools for Quantum Computing III.
Boolean functions and Euler's identity.

Chapter 14. Dirac Notation.
Using Dirac notation to represent vectors, vector operations, and tensor products.

Chapter 15. Table of Quantum Operators and Core Circuits.
A table of quantum operators and core circuits.

3 Opinion

I used *Quantum Computing: An Applied Approach* as a textbook for my *Introduction to Quantum Computing* course during the Fall 2023 semester. My students were junior and senior computer science majors who had taken a course in linear algebra and were proficient in Python. None of them had any previous experience with quantum physics or quantum computing.

In the introduction Hidary gives three options for using the book in university courses. The options are keyed to the type of student in the course: STEM majors, physics graduate students, or computer science graduate students. My demographic seemed to fit the author's description for a course in quantum computing for STEM majors as they were not graduate students in either physics or computer science. We covered the material in Chapters 1, 2, 3, 4, 6, 7, 8 and parts of 9 and 10. Because my students had already studied linear algebra, we were able to use Part III as a resource.

I was disappointed with the text. First, the quantum physics discussion in Chapter 1 was too sketchy and confusing for my students. My students responded better to explanations of superposition and entanglement that were rooted in the matrix algebra of qubits. Hidary does use the polarizing filter example of superposition in Chapter 1 and that's fine. However, it was easier for my undergraduates to see what was happening in the context of linear algebra. Hidary does do some of that in 11.3, but it would improve the text, I think, to have that upfront. Hidary's explanation of entanglement on p. 8 is way too brief. Again, I would have preferred to see the concept arising from the linear algebra calculations. The first chapter could be much longer with some of the material in Part III incorporated. I used another text (*Quantum Computing for Everyone* by Chris Bernhardt) and provided my own supplements to clarify the notions of superposition and entanglement and the mathematics involved therein. The problems for Chapter 1 from the GitHub resources required concepts beyond those presented in the chapter. This was another disappointment.

The problems in the GitHub resources were, except for Chapter 8, very difficult and often tangential to the chapter material. For example, one problem for Chapter 2 is about the CZ-gate, which had not yet been defined. Most of the chapters in Part II had no intertextual exercises, so I needed to write my own.

I was also disappointed with the code provided. I am not surprised when printed textbook code is out-of-date or faulty, but I was hopeful that the GitHub resources would provide up-to-date code. That was not always the case. The faulty code was usually due to mismatches in class names in Cirq. For example, in the text the Adder class (p. 124) inherits from `cirq.ArithmeticOperation`, but that superclass is actually `cirq.ArithmeticGate`. My students and I needed to perform numerous searches on the Cirq website (`https://quantumai.google/cirq/start/basics`) to obtain proper syntax, as well as method and class names, in order to make the code execute properly. Fortunately, the Cirq code for the fundamental algorithms is also available on the Cirq website, along with tutorials.

On the positive side I liked how Hidary broke down some of the standard algorithms, like Shor's, into smaller code sections. That built up the algorithm using easy to digest pieces. The result was that my students understood the structure of the algorithm. Also, in general it was helpful to have the algorithms implemented rather than in pseudocode. I was able to guide my students to explore quantum computing ideas and experiment with minimal coding. To improve the text to make it more undergraduate friendly I would suggest expanding the first chapter to include the basics of the matrix theory of qubits, design more and easier exercises for each chapter intertextually or in the GitHub resources, and make sure the GitHub code is up to date with the current rendition of Cirq.

Perhaps graduate students in either physics or computer science would have a much easier go with *Quantum Computing: An Applied Approach* than my class did. The text contains an abundance of information on quantum computing. I think physics students would find Chapter 5 of particular interest. Graduate computer science students would be less bothered by code inconsistencies and be more able to appreciate the program design. Nevertheless, I cannot recommend the book for an undergraduate course.