Alberto Palazzi

Quantum Computing for Programmers and Investors

GogLiB ***ebooks***

ISBN: 9788897527541

Contents

1. INTRODUCTION

An explanation for programmers and investors
Bibliography and verification
QcNooq project and download of the source code
Practical information
Notation

2. COMPLEX NUMBERS

2.1 Arithmetic of complex numbers

3. OPERATIONS ON VECTORS AND MATRICES OF COMPLEX NUMBERS

3.1 Vectors
3.2 Matrices

4. BITS AND QUBITS

4.1 Basic states and evolution of states
4.2 Actions on a system
4.3 Qubit
4.4 Composition of states
4.5 Measurement and result reading

5. QUANTUM GATES

5.1 Classical gates
5.2 Identity gate and reversible gates
5.3 Quantum gates

6. QUANTUM ALGORITHMS

6.1 Deutsch's algorithm
6.2 Deutsch–Josza algorithm
6.3 Simon's algorithm
6.4 Grover's algorithm
6.5 Shor's algorithm

7. PERSPECTIVES OF QUANTUM COMPUTING

7.1 Useful algorithms and state of the art
7.2 Quantum programming languages
7.3 A brief history of the quantum computer project
7.4 Conclusion for investors

APPENDIX: USE OF THE QCNOOQ PROJECT

BACK COVER

1. Introduction

An explanation for programmers and investors

Why this book is written for programmers and investors together will be fully understood after reading it. Preliminarily, let us say that regarding quantum computing there are two questions:

1) when will it be possible to build an efficient quantum computer?

2) what problems will it solve?

The books on quantum computing that have been written contain notions of a different nature: they speak in greater or less detail of the (quantum) physical principles that govern subatomic phenomena, they expose the mathematics necessary for the study of quantum physics (linear algebra), and finally they deal with quantum computing. In this book, readers will not find any notion regarding physical principles, and with regard to mathematics they will find only the applied part necessary for quantum computing, which consists of algorithms for arithmetic operations on vectors and matrices of complex numbers. Then on this basis the readers will find the description of the most famous quantum gates and quantum algorithms, with implementation in C language. The quantum computer will be described as a hardware black box that is able to transform a given input into a given output, such as it always happens in computer science texts, in which the notions concerning semiconductor electronics underlying the calculations are only hinted at and could even be completely omitted.

Therefore, this book has no answer for question 1. If ever and when shall we succeed in building an efficient quantum computer, it is such a question that requires a thorough knowledge and experience of quantum physics in order to hazard an answer.

Instead, reading this book, readers will find themselves in possession of a precise answer to the second question: if tonight the devil, as in fairy tales, built a perfectly efficient and stable quantum computer, capable of handling a matrix of qubits of considerable size, the next day for what purposes could we use it? It must be said immediately that the peculiarity of quantum hardware will be to perform in a single act, a single change of state of the machine, certain operations on matrices that today's computers based on the principle of the Turing machine must perform through the iteration of numerous cycles nested one inside the other, and therefore with considerable execution times, and for certain problems with such extended times as not to allow technically useful solutions. By

performing transformations of the input corresponding to a certain (ideally very large) number of cycles of a classical computer with a single change of state, we read and hear that the quantum computer will be able to drastically cut the execution time for operations of encryption and cryptography, and for finding solutions to highly complex problems such as those of logistics, optimization, scheduling, operational research, etc.

How this could happen can be understood through the emulation of quantum algorithms using the classical computer available today, although obviously the emulation will have no practical use because the emulation of a quantum algorithm without quantum hardware will always require computing resources greater than those required to run the corresponding non-quantum algorithm. That is, suppose we need to perform a transformation of a matrix that the quantum computer will perform in a single state change of the machine. Suppose that the same transformation by an algorithm executable on a classical computer requires the repetition of, say, a thousand or a million cycles. Well, if we try to get the same result with a classical algorithm that emulates a quantum algorithm, as we shall see and as is imaginable, we shall need a number of cycles much greater than the thousand or one million of the classical algorithm, and we shall also need to allocate much more memory. We shall see below, for example, how the classic computer emulation of Shor's algorithm for finding the factors of a number is enormously less efficient than any elementary algorithm for finding prime factors.

This book is written for *programmers* because to read it you only need to have the common basics of computer science (logic gates, flowcharts, programming languages). The examples are written in C language in the simplest way and without the use of constructs that are not elementary, so that anyone who can read any programming language will understand it. The purpose of the book is to lead the reader to understand the logic of the quantum algorithms described in chapter 6, and therefore everything explained in chapters 2 to 5 is enormously simplified and reduced to the bare minimum: the deliberate choice was to give to the reader only the premises necessary to understand the logical flow and the calculations of quantum algorithms, which is supposed to be the aim of the reader, and for this reason the chapters preceding the one dedicated to quantum algorithms contain only the information that is a necessary condition for understanding.

It is important to point out that the knowledge of the purely computer science part of quantum algorithms is not subject to any limitations due to the fact that it completely disregards the physical characteristics of the hardware. The proof lies in the fact that by reading this book the readers will be able to implement and execute quantum algorithms on their PC, obtaining the results envisaged by the theory: therefore the knowledge of quantum algorithms will be so complete as to allow their application and verification. This also proves that quantum algorithms in themselves do not require quantum hardware, just as the CPU of a computer could theoretically be built by mechanical means rather than by exploiting electronic properties: but it would be too slow a machine to be useful for anything, and exactly the same happens by emulating quantum algorithms with the classical computers existing today.

Since this book allows people having the knowledge of a *programmer* to understand exactly what a quantum computer could be used for, once built, it solves at least half of the problem that investors face when evaluating whether and how much it is appropriate to risk investing in development of quantum computing. Therefore, investors (private investors, consultants, investment fund managers, managers of funds financing technology initiative, etc.), if they do not personally possess the necessary prerequisites to understand this book, could use it by summoning some IT expert they trust, and leaving to this expert the task to read it, understand it and report on the result.

Bibliography and verification

This book is a simplified presentation of the theory set out in full in two fundamental treatises, which are:

- Nielsen, Michael & Chuang, Isaac L., *Quantum Computation and Quantum Information*, Cambridge University Press, 2000 and 2010
- Yanofsky, Noson S. & Mannucci, Mirco A., *Quantum Computing for Computer Scientists*, Cambridge University Press, 2008.

Everything stated in the following in a descriptive way and as a matter of fact, without demonstrations and without quotations, can be verified and deepened through the study of these two volumes, which the reader who has assimilated this book will probably find easier than it seems at first sight.

The fact that we are now in 2020 is unimportant: these two books expose the theoretical basis of the matter in a way that is

consolidated, and in the ten and more years that have passed nothing has been achieved but some progress in the construction of hardware prototypes. This can be confirmed by reading another very recent, simpler and yet rigorous discussion:

- Bernhardt, Chris, *Quantum Computation for Everyone*, The MIT Press, 2019

which discusses quantum gates and the five fundamental algorithms (referring only by hints to Shor's) in exactly the same way as the previous treatises, and does not contain anything new from the point of view of software development.

Compared to these treatises, our discussion completely lacks both the physical part and the demonstrations of the numerical properties on which quantum algorithms are based, and is aimed at favouring the concrete understanding of the algorithms through implementation; however, the state of things and the potential of quantum computing that are highlighted by our discussion correspond exactly to the conclusions that anyone can draw from a careful reading of those more complex and complete books. Readers are therefore warmly invited to use this book to become familiar with the subject, and then to study the subject in a more abstract and rigorous way in the treatises cited above, which to those who assimilate this book will no longer appear as complex as it happens to those who approach these topics for the first time. The first book to read is Bernhardt's of 2019, which for many readers will have a more than satisfactory theoretical rigor, and also provides a minimal introduction to physics underlying the operation of hardware.

We have no information about the existence of simplified and useful informative works: the simpler books we have consulted are all constructed in a too generic way to allow any understanding of the subject. And in particular we advise readers not to start from quantum programming languages, which as a starting point are incomprehensible, while for the reader who has understood the fundamental quantum algorithms they are an obvious and extremely simple consequence. However, for a simpler and updated discussion oriented to programming languages, you can consult:

- Radovanovic Aleksandar, *Quantum Computing Illustrated*, qpi-book, 2020.

QcNooq project and download of the source code

The source code of the programming examples that are reproduced in the book contains only the instructions necessary for the

understanding of the algorithms. We recommend that you read the book carefully a first time without worrying about running the code: if you don't understand what you read in the book, the source code won't help.

Those who wish it could implement their own emulation of the five algorithms using only the code mentioned in the book, just adding the necessary instructions for the output and verification of the results. Anyway, all source code is available in Visual C for Windows. The project is called *QcNooq* and is described in Appendix, with instructions for use in Windows and other environments.

Practical information

This book is available in e-book on many e-book stores and also in print format on Amazon. The format was designed to make the book readable even on the small screen of an e-book reader, but with inevitable limitations, so some readers will prefer the paper format.

Notation

The ► symbol draws attention to the fact that the following lines contain a definition to remember.

Sometimes in the text there are additional notes and explanations, or parts that can be read quickly by those who already know the details discussed: these parts of the text are in slightly smaller type.

The notation of the source code fragments is that of the C language, of which the simplest constructs similar to those of any other language are used, which are assumed to be known by the reader. However, some explanations are added to clarify what is not entirely obvious. Since the C language is currently used in the discursive parts of the text, when for readability this is appropriate the long names of variables are underlined as in this example: "the result is read in the variable mresult".

The indexes of vectors and matrices throughout the text are indicated in the simplest way given the context, therefore with subscripts when looking at them from a mathematical point of view, and with the notation of C when referring to the implementation. So a vector of real numbers could have the notation v_n or: double v[N], and for an element of it v_j in the first case and v[j] in the second.

For quantum circuits we shall use the notation generally used by all texts, and we shall introduce it at the appropriate moment.

2. Complex numbers

2.1 Arithmetic of complex numbers

Probably all readers of this book remember the basics of complex numbers. Here we recall only those notions that will be used in the implementation of the code to emulate quantum algorithms; the study of quantum physics and quantum computer hardware, on the other hand, would require a complete review, starting with the geometric representation of complex numbers in polar coordinates.

..... end of preview

Back cover

This book is aimed at people who simply know the basics of computer programming. It does not require any notion of physics and allows its readers to understand with total accuracy and in the simplest way the use that could be made of a quantum computer, by explaining step by step how to write a software to emulate its operation. The usual expression that a qubit "is an object that can be simultaneously in both binary states 0 and 1" will lose all the aura of mystery that surrounds it, and readers will understand exactly its meaning and implications for computing without the need for any knowledge of physics. The book describes quantum computing by considering it strictly from a computer science point of view, simply as a machine that is capable of transforming a given input into a given output using any suitable physical principle to work, and thus allows to become completely familiar with quantum gates and with the most famous quantum algorithms. The only condition is that the readers are familiar with some programming language and with the basic concepts of classical computer science: those who have this knowledge will easily follow the description of quantum algorithms and understand the software emulation that is implemented in the book, and will also have fun running and testing it with their own PC.

The knowledge acquired through this book is of vital importance to *investors* because it allows them to judge independently on the risk of investing in this technology. It has been written for *programmers* because the knowledge of basics of computer science is useful to understand exactly what a quantum computer, once built, could be used for. But this understanding is also essential for *investors* who must evaluate whether and how much it is appropriate to risk investing in the development of quantum computing. Therefore, even *investors* (private investors, consultants, managers of financing funds for technological enterprises, etc.) who want to decide on the allocation of resources in quantum computing with full knowledge of the stakes, must know this book, and if they do not personally own the necessary prerequisites, they can use it by hiring some trusted computer expert to read it, understand it and report on the result.

Alberto Palazzi

A scholar of history and philosophy of science, and designer of computer algorithms for the solution of higher complexity problems, the author of this book is a member of the *QcNooq* consulting team

(www.zonabit.it/qcnooq), whose mission is to provide investors the clarity of view necessary to decide on investments in quantum computing.