Improving Quantum Computing Education for Undergraduate Students

Keith Gordon

B.S. Candidate, Department of Computer Science, California State University Stanislaus, 1 University Circle, Turlock, CA 95382

Received 18 April, 2020; accepted 15 May 2020

Abstract

Quantum computing is a promising field of computer science but is difficult to teach to undergraduate students, often requiring extensive backgrounds in computer science, mathematics, and physics. This research project is focused on taking the complex subject of quantum computing and making it more accessible to undergraduate computer science students. I will be examining literature about effective teaching methodology, and then applying these methods to create a hypothetical course syllabus and outline. The syllabus will contain information such as example projects, assignments, and grade breakdowns. The course outline will consist of the most crucial topics necessary to have an acceptable quantum computing introductory course.

Introduction

Quantum Computing is a method of computing that uses quantum mechanics to improve upon classical computing, eliminating barriers encountered within modern computers. Classical computers use sequences of bits, which can be either zero or one, to perform computations. Quantum computers use quantum bits, also known as qubits, which can be a combination of both zero and one, which allows quantum computers to perform calculations on many possible combinations of inputs simultaneously (Giles, 2019). Qubits also have a characteristic known as, "entanglement," which links multiple qubits together and allows quantum computers to drastically reduce the time it takes to perform calculations (Giles, 2019). Both of these characteristics make quantum computing theoretically much more powerful than classical computing, which could introduce new capabilities to computing, such as simulating the behavior of matter, advancing artificial intelligence, and many other possibilities (Giles, 2019). However, Quantum computing is a highly complex field of computer science, involving aspects of classical computing, quantum physics, linear algebra, and several other disciplines, which results in a lack of skilled workers (Giles, 2019). Quantum computing is a very promising field, but due to a lack of education, there is a shortage of people who can actually work in the field. A way to combat this shortage would be to introduce quantum computing in more schools, but this solution is difficult to implement in a practical sense, because most computer science programs don't provide students with an adequate background to understand a typical quantum computing course, at least not with traditional pedagogy. My research is focused around developing a method of teaching quantum computing in a manner that the average computer science student could understand, without having to introduce more requirements to the computer science major or sacrifice essential knowledge.

The research I am doing is primarily in the field of quantum computing, therefore that will be the focus of most of the literature to be reviewed. However, my topic could be described as a comparison between quantum and classical computing, and while the paper is focused on quantum computing, some of the underlying explanations will be centered around an understanding of computers in a traditional sense. There will be several topics my research covers, typically requiring first the classical computing explanation, followed by its comparison to quantum computing, but this may vary depending on the complexity of the topic.

Literature Review

My literature review will be focused on several articles I found regarding quantum computing, and how they could be better explained. Some of these articles lack important details, or they are simply geared more towards readers with adequate backgrounds, so I will be exploring ways they could be better explained to a reader that is less familiar, considering the audience for my proposed course would be an undergraduate student.

One of the first topics about quantum computing to logically start with is why quantum computing is significant, and what it can allow us to do differently. A good piece of introductory literature to quantum computing is an article written by Cardinal (2019). This article covers the basics fairly well, but when it comes to discussing the purpose of quantum computing, it mentions little more than it can change cryptography. The article also does very little to describe how quantum computing can change cryptography. Quantum computing has the capability to change many fields we're familiar with today, and I'd like to pique students' interest by showing the incredible capabilities of quantum computing, which includes cryptography but also includes much more.

As mentioned previously, cyber security is a big concern when it comes to quantum computing. This topic is discussed in an article by Wallden and Kashefi (2019). This article discusses quantum computing's possible changes to cyber security, but seems to shy away from technical details. The topic is very complex, of course, and is difficult to explain well without including some technical information. I, for example, would include more information about how cyber security is currently maintained, and go in depth as to why quantum computers would change these already existing methods. This would ideally contain statistical data and numbers to express efficiency of cyber security methods for example.

I will also, of course, discuss the more technical aspects of quantum computing. To begin, the bit is arguably the most essential part of a computer. Bits convey all information to the computer in the form of zeros and ones, representing two states: off or on. This is not the case when it comes to quantum computing. Quantum computers use bits known as, "qubits" which can exist in a state of both zero and one simultaneously through the concept of superposition. This concept is explained in a book written by Silva (2019). Silva explains that qubits have to extremes representing the base vectors |0> and |1>, which can be represented on a sphere by the north and south pole. However, the space in between the two poles can be described as a mixture of the two vectors, allowing them to both exist simultaneously (Silva, 2019). This source is so far one of the most understandable explanations of quantum computing I've come across, but when it comes to quantum computing, that still isn't understandable enough. For example, there is no background information given about the vectors that are a crucial part of qubits, a fundamental characteristic of quantum computing. This is because Silva presumably made the assumption that the reader will have prior knowledge about vectors before reading. This may be true for some readers, but vectors are not always discussed much throughout the computer science curriculum, therefore many students may have trouble understanding this concept.

The next topic I will discuss is logic gates. Logic gates are the basic way that computers use bits to perform logical tasks. Logic gates implement boolean logic, which is essential to computer operation, considering it takes in binary as an input, and returns binary as an output. Gibilisco (2018), gives multiple examples of basic logic gates. The issue with this source, is that it only discusses logic gates in respect to classical computing, which is not very useful when trying to understand quantum computers and how they operate. This is why I will attempt to use some of the examples from this book and compare them to quantum logic gates found in another one of my sources from Spector (2007). For example, one of the gates listed by Gibilisco is the NOT gate. The NOT gate takes in a bit, and returns the opposite. Therefore, if a NOT gate is applied to a zero, the output is a one (Gibilisco, 2018). Spector also explains a NOT gate, but in relation to quantum computing instead. Spector explains that a quantum NOT gate can be represented by the matrix:

[0][1]
[1][0]
(Spector, 2007).

The issue with this explanation is that it assumes the reader has knowledge of matrices, which many computer scientists do not.

Another topic I would like to discuss is the quantified improvements of efficiency using quantum computers as opposed to classical computers. Throughout my paper I will discuss how quantum computing works and how it differs from classical computing, but it is also useful to understand how these differences result in improved performance. Null and Lobur (2019), explains that quantum computing is more efficient because the ability for the qubits to be in superposition allows the computer to process computations for all possible combinations of the bits, resulting in what is known as "quantum parallelism." This source does well explaining this concept, but I believe it would be more useful if it were to also demonstrate a practical application that displayed the superiority of quantum computers, such as searching through an unordered database.

Lastly, when it comes to computers, programming is very important. Quantum computer programming is much different from classical programming, however, a language called Quantum++ has been developed that hopes to merge classical programming with quantum. This language is discussed by Gheorghiu (2018). Gheorghiu gives a lot of information, but it reads a lot like documentation for a programming language. My focus would be centered around a more practical approach. Many programmers use Java, for example, but very few know how it operates. Knowing how a programming language operates is very useful knowledge, but it is unnecessary when it comes to introducing a language. Programming should be emphasized first, and the technical aspects should be taught later. Another useful source regarding quantum programming I found was an article by Sofge (2008). This article discusses the evolution of quantum programming languages, including how quantum programming is different from classical programming. This information can be of use to include in my course content to help better understand what differentiates quantum from classical programming, and how they should be subsequently approached.

There are more sources I will be using as my research develops, but these are some of the sources I have come across in my research thus far. My main approach to bettering the literature I use will be connecting concepts through their relation to both quantum and classical computing, considering most computer science students are more comfortable with classical computing. The next step I make will be taking these concepts and making them understandable to computer scientists that lack a background in some of the more advanced fields involved in quantum computing. My goal is to make quantum computing more understandable to the average computer science student without sacrificing accuracy, and while the existing literature on quantum computing is great, it is most often far from being easily understood.

Methods

The goal of my research is to develop a simple, yet effective course outline to teach quantum computing to undergraduate computer science students. To accomplish this task, I will be studying quantum computing and examining which topics a quantum computing course generally includes. I will also be studying literature about teaching methods and what methods are most effective to determine how I will structure the course. I will then produce a hypothetical course outline and syllabus that could be used as a guideline for an undergraduate computer science course. The purpose of the course outline and syllabus is to act as a substitute for a full course, as creating a full course is outside the scope of this project. As a senior undergraduate computer science student, I know which courses other students can be expected to have taken, so I will be able to determine which topics need to be simplified due to a lack of background understanding. The course syllabus will also draw from my research on pedagogy, as it will include how assignments are to be performed, how the grades will be determined, and more information that a student would generally expect from a course syllabus to have a full understanding of the expectations of the course.

Design

I hope to develop a theoretical course on quantum computing, but I can’t realistically create material that would span the entire course, assuming it is the length of a normal course. Therefore, I will create a syllabus that could be used to model such a course, including information like necessary prerequisites, projects, and grading policies. I will also include a course outline to grant readers an idea of how the course would be organized and which topics would be covered.

Results

SYLLABUS:

Introduction to Quantum Computing

Prerequisites: CS 3740 (Computer Organization) and CS 3100 (Data Structures and Algorithms)

Course Description:
This course will serve as an introduction to quantum computing. The course will not be as in-depth as a quantum computing course one could typically find within a graduate program, nor will it be as in-depth as a quantum computing course with a long list of prerequisites. This course is designed for undergraduate computer science students and will approach the subject in a way that makes it understandable for students who haven’t had extensive backgrounds in physics or math. This course will introduce you to basic quantum computing concepts, the physics behind them (including math), how quantum computers differ from classical computers, and possible ways it can change the world of computing. This course will also introduce you to basic programming with quantum computers.

Grading Breakdown:
Regular Assignments: 50%
Tests/Quizzes: 30%
Programming assignment(s): 10%
Final project: 10%

Grading Policy:
Late work will be accepted for 80% of the grade received. Test and quiz portion can be adjusted if the student shows improvement. Programming assignments will be graded based on effort (to be determined by the professor).

Grading Scale:
This course will be graded on a standard scale. 97–100% A+, 93–96% A, 90–92% A-, 87–89% B+, 83–86% B, 80–82% B-, 77–79% C+, 73–76% C, 70–72% C-, 67–69% D+, 63–66% D, 60–62% D-, <60% F.

Final Project:
The final project will be centered around performing a lecture for your peers about a topic of your choice, relating to quantum computing of course.

Weekly Homework:
Each week you will be turning in a short summary in your own words of the information covered throughout the week.

Required Materials:
Any necessary subject materials will be provided by the professor. These will mostly include learning materials included with your enrollment to the university, provided by the university library.

A large influence on how the syllabus was constructed was the book, Handbook of College Science Teaching by Joel J. Mintzes and William H. Leonard. In the book, Mintzes and Leonard explain that motivating students to learn is incredibly important for a science course. Mintzes and Leonard also give advice on how to motivate students to learn (Mintzes & Leonard 2006). One of the tips the authors provided was to allow students some degree of freedom regarding what to learn and how to learn it. The authors also state that peer instruction has been proven by research to increase conceptual understanding as well as improve the ability to solve quantitative problems. It is for these reasons that the final is a project centered around peer instruction, and the students are able to choose their topic.

Some choices made in the syllabus are more arbitrary, but are guided by principles learned from my research. The grade breakdown is a great example, since research won't tell you exactly how to grade a class, but can give advice to increase student learning. Another source of influence for this syllabus was an article titled, "Teaching About Teaching Science: Aims, Strategies, and Backgrounds of Science Teacher Educators" by Amanda Berry and Jan H. Van Driel. This article explained that teaching educators, when teaching preservice teachers, strongly emphasized focusing on student learning and avoiding traditional teaching methods (Berry & Driel 2012). This motivated me to avoid structuring the class in traditional ways for the sake of familiarity, and instead making decisions I felt best emphasized learning. For example, this class accepts late work for up to 80% of the original grade. This is because students need to be somewhat motivated to turn it in on time so the educator is allowed time to grade, but also not to deter students from completing work simply because the due date is passed. If the late work is not accepted, students are less likely to complete the assignment, but the point of the work should be to help students learn. If the students complete the work later, they still learned the material. This article also motivated me to include details such as the test and quiz portion of the grade being adjusted if the student shows improvement. This is another example of focusing on student learning and shying away from traditional methods.

COURSE OUTLINE:

Week 1: Syllabus/What is quantum computing?

Week 2: Classical vs Quantum Computing

Week 3: Linear Algebra Crash Course

Week 4: Superposition

Week 5: Entanglement

Week 6: Classical problems in quantum solutions
Week 7: Quantum algorithms

Week 8: Shor's algorithm

Week 9: Basic quantum programming

Week 10: Review

Week 11: Final presentations

Week 12: Final presentations

The topics to be included in this proposed course are also arbitrary and can be changed in accordance to what the professor may consider to be more important, but this outline can act as a general guideline. My focus was to include important concepts, as well as provide time for practical application in the form of programming. In, Handbook of College Science Teaching, Mintzes and Leonard discuss a study performed on introductory level science courses. The study found that in introductory courses, it was important to focus on fewer concepts and to explore the topics extensively using examples and practical applications (Mintzes & Leonard 2006). As mentioned earlier, the final for the class is a presentation of a quantum computing topic by students, so there is also time allotted for that at the end of the course.

Aside from the organization and policies of the class, it is also important to convey the information in an understandable way, considering the students in this class are simply undergraduate students with limited backgrounds in math and physics. The professor can't simply start mentioning matrices and other advanced concepts without first addressing them and giving some background. This isn't a math course, so there are limitations to how much detail you can dive into, but the professor should make an effort to convey as much information as necessary to understand the topics. For example, in the literature review section, I mentioned that Spector in an article from 2007 explained that the quantum NOT gate can be represented by the matrix:
[0][1]

[1][0]
(Spector, 2007). In this course, the professor can't assume the students are all familiar with matrices, so the professor would have to first remind students of a classical NOT gate, and then explain that the quantum NOT gate behaves in the same way, but is presented in a 2 x 2 matrix because it is acting on a single qubit, and quantum gates are represented by a 2^n x 2^n matrix where n is the number of qubits (Roell 2018). The professor would also give some background regarding vectors and matrix multiplication because quantum gates are inseparable from these concepts and can be best explained by them. However, the professor doesn't need to be certain that students are completely educated in matrix mathematics, but the students need to be introduced to it just enough to be able to understand what is taking place within a quantum gate and have enough of an understanding to read literature about quantum computing and not be discouraged at the mere sight of a matrix. While we are currently discussing matrices and logic gates, the professor would have to take these approaches for all the topics in the course that typically require a more extensive background to discuss.

Discussion

I feel that the proposed course syllabus and outline can act as an effective guideline for an introductory undergraduate quantum computing course. This project is mostly centered around my own experience and my own conclusions drawn from the literature I studied, so there is no certain way to say this is undoubtedly the best approach to take, but I hope some educators who are also struggling with how to approach introducing quantum computing to undergraduate education can examine this project and draw inspiration from it and make adjustments where they see fit. As mentioned earlier, many details included in the syllabus and outline were arbitrary but were meant to provide a general idea of how one could approach a course like this, so there are plenty of changes one could make if they saw fit. I am hoping that my study can further quantum computing education, even in the smallest ways.

References

Berry, Amanda, and Jan H. Van Driel. "Teaching About Teaching Science." Journal of Teacher Education, vol. 64, no. 2, 2012, pp. 117–128., doi:10.1177/0022487112466266.

Cardinal, D. (2019). How Does Quantum Computing Work? PC Magazine,105-110.

Gheorghiu, V. (2018). Quantum : A modern C quantum computing library. PLoS ONE,13(12), 1-27. doi:10.1371/journal.pone.0208073

Gibilisco, S. (2018). Beginners guide to reading schematics (4th ed.). New York: McGraw Hill Education.

Giles, M. (2019, January 29). Explainer: What is a quantum computer? Retrieved from https://www.technologyreview.com/s/612844/what-is-quantum-computing/

Jun, Li, and Zhang Huiqing. "A Study on the Reform of VB Programming Teaching Methods." 2012 7th International Conference on Computer Science & Education (ICCSE), 2012, doi:10.1109/iccse.2012.6295409.

Mintzes, Joel J., and William H. Leonard. Handbook of College Science Teaching. NSTA Press, 2006.

Null, L., & Lobur, J. (2019). The essentials of computer organization and architecture (5th ed.). Burlington, MA: Jones & Bartlett Learning.

Roell, Jason. "Demystifying Quantum Gates - One Qubit At A Time." Medium, Towards Data Science, 28 Feb. 2018. Retrieved from towardsdatascience.com/demystifying-quantum-gates-one-qubit-at-a-time-54404ed8064.

Silva, V. (2019). Practical Quantum Computing for Developers Programming Quantum Rigs in the Cloud using Python, Quantum Assembly Language and IBM QExperience. New York: APRESS.

Sofge, Donald A. "A Survey of Quantum Programming Languages: History, Methods, and Tools." Second International Conference on Quantum, Nano and Micro Technologies (ICQNM 2008), 2008, doi:10.1109/icqnm.2008.15.

Spector, L. (2007). Quantum computing. Proceedings of the 2007 GECCO Conference Companion on Genetic and Evolutionary Computation - GECCO 07. doi:10.1145/1274000.1274128

Wallden, P., & Kashefi, E. (2019). Cyber security in the quantum era. Communications of the ACM,62(4), 120-129. doi:10.1145/3241037

.