

Undergraduate Research in Quantum Computing: Lessons Learned from Developing Student Researchers

David H. K. Hoe, Loyola University Maryland

David H. K. Hoe did his undergraduate and graduate studies in electrical engineering at the University of Toronto. His professional experience includes positions at the General Electric Research and Development Center as a Staff Engineer and at the University of Texas at Arlington as an Assistant Professor. He is currently an Associate Professor of Engineering at Loyola University Maryland.

Dr. Mary Lowe, Loyola University Maryland

Mary Lowe is a professor of physics at Loyola University Maryland. Dr. Lowe earned her Ph.D. in experimental condensed matter physics from the University of Pennsylvania. At Loyola she has taught all levels of lectures and labs for undergraduate physics majors. Her grant-funded teaching projects have included robotics in the introductory physics lab and the development of physics of medicine modules. Over the years, she has conducted research at Loyola on fluid flows in large, curved ducts and in the microcirculation; multiplexed identification and quantification of DNAs on the surface of microscopic beads using flow cytometry; molecular dynamics simulations of proteins and lipid membranes; and atomic force microscopy measurements on DNA/protein complexes and carbon nanotubes. Her latest teaching interest is quantum computing.

Dr. Dave Binkley, Loyola University Maryland

Dave Binkley is a Professor of Computer Science at Loyola University Maryland where he has worked since earning his doctorate from the University of Wisconsin-Madison in 1991. He has been a visiting faculty researcher at the National Institute of Standards and Technology (NIST), worked with Grammat-ech Inc. on CodeSurfer development, was a member of the Crest Centre at Kings' College London, and a Fulbright scholar working with the researchers at Simula Research, Oslo Norway. Dr. Binkley's current research interests include tools and techniques to help software engineers understand and improve their code.

Undergraduate Research in Quantum Computing: Lessons Learned from Developing Student Researchers

Abstract

Numerous studies have established several positive outcomes associated with meaningful research experiences undertaken by undergraduate students. These benefits include enhanced critical thinking skills, improved learning abilities in subsequent course work, and increased pursuit of graduate studies. This paper reports on our supervision of several undergraduate students over the past couple of years in the area of quantum computing. Case studies of four projects involving six students are described. Lessons learned from working with our student researchers are given and recommended best practices for undergraduate research in this area are detailed. Finally, how this work dovetails with our implementation of a multi-disciplinary introductory quantum computing course running for the first time this Spring is also described.

1. Introduction

Educational scholars widely agree that engaging students in authentic undergraduate research experiences is a high-impact pedagogical practice as it promotes student-centered learning and results in several positive learning outcomes including improved problem-solving, critical-thinking, and communication skills [1], [2], [3]. Undergraduate research programs also help with personal development, giving students improved attitudes, self-confidence, and personal initiative. Finally, it provides opportunities for professional development through networking with peers and mentors, and publication of their work [2]. In addition, such research experiences lead to increased retention in STEM fields and a greater likelihood of pursuing a graduate degree. Noteworthy as well is that undergraduate research experiences are beneficial to students at risk of underachieving [4].

To ensure a student has an impactful undergraduate research experience, it is important to provide appropriate projects. Students are highly motivated when they know that they are contributing to an authentic research question. At the same time the project must be tailored to the knowledge and capabilities of the student so that they are able to make meaningful contributions to the research. This paper argues that *quantum computing provides abundant opportunities for successful undergraduate research projects.*

Peter Shor's discovery that quantum computers could be used, in theory, to factor large integers almost exponentially faster than the best classical algorithms [5] generated a lot of interest in quantum information, science, and technology because of its potential to make current public-key encryption schemes obsolete. In addition, quantum computers show great promise for solving classically intractable optimization problems (e.g., the traveling salesman problem [6], machine learning [7], and problems in computational biology [8]). As suggested by Richard Feynman [9], quantum computers should be able to efficiently simulate quantum systems. Work is proceeding on the quantum chemistry of molecules and materials [10], [11], [12]. Recent experimental devices have confirmed 'quantum supremacy' over certain classical approaches [13], [14], raising the possibility that quantum computers will be a viable computing technology

in the coming decades. The end of Moore's Law appears to be imminent as transistors are scaled deep into the nanoscale dimensions [15]. As such, quantum computing is generating widespread interest and excitement in both industry and academia as an emerging technology of the future.

Based on our own experiences in mentoring undergraduate researchers over the past two years, we believe that quantum computing (QC) is an exciting area of investigation which allows undergraduate students to make meaningful contributions to open-ended research questions. The outline of our paper is as follows. The next sections provide the context and goals for our undergraduate research and explains why QC topics make viable and exciting research projects for undergraduates. Our experience mentoring two student projects in the summer of 2022, simulating Shor's algorithm using a quantum computer simulator and investigating machine learning using quantum computers, is detailed next. Some insights from mentoring undergraduate students during the semester in an independent study in the spring of 2022 are also described. The implementation of our own introductory undergraduate QC course which features innovations such as students developing their own Python-based quantum computer simulator and using it in a course project is outlined. The final section summarizes the lessons learned from our experiences supervising undergraduates in QC.

2. Background, Motivation, and Goals for Our Work

Loyola University Maryland is a private liberal arts institution of approximately 3800 students. It focuses primarily on undergraduate studies and features STEM departments in engineering, computer science, physics, chemistry, biology, and mathematics. At our university undergraduate students have the opportunity to do research during the summer and during the academic year.

The genesis for our foray into QC research began with three faculty members (one each from our institution's physics, engineering, and CS departments) discussing the possibility of developing an introduction to QC course. An interesting project was proposed: to develop a quantum computer simulator from scratch using the Python programming language and NumPy, a Python based numerical programming library. In the summer of 2021, we were fortunate to have two exceptional undergraduate researchers who developed such a simulator. One student had just finished his freshman year as a physics major while the other had completed his sophomore year as a computer science major. While the students had no prior knowledge of QC, they were able to get up to speed through materials provided by their faculty mentors and their own initiative in finding appropriate tutorials on the Internet. By the end of the summer the students were able to complete a rudimentary Python-based simulator.

This project was attractive for two reasons. First, developing the code for a simulator opens the door to a range of interesting research, from adding one's own custom noise models, to investigating ways to accelerate the simulator or add the ability to handle more qubits, which are all viable research problems in their own right [16], [17]. Second, the development of the simulator forms a driving motivator for learning QC and thus provides a basis for our introductory class on QC.

3. Quantum Computing for Undergraduate Research

This section describes our approach for teaching introductory QC to undergraduates, particularly the simulator, which enables them to engage in viable undergraduate research. In addition to learning the theory using Dirac’s bra-ket notation, the students write their own Python code to automate the operations otherwise being done by hand. In the simulator quantum gates are represented as matrices; the tensor product is used to calculate the full matrix at each moment in time; the complete quantum state vector (wave function) is computed at each stage of the quantum circuit; and a measurement method is implemented. We adopted the approach used by Google Cirq (<https://quantumai.google/cirq/build/circuits>) to subdivide the circuit into “moments,” where each moment is a set of gate operations executed in a slice of time. The gates in each vertical column form a moment as illustrated in Fig. 1. In a physical quantum computer, such moments do not necessarily represent the order of operations, but in a simulator, moments help the students organize their thoughts on how the circuit behaves and thus aid in learning QC principles. States, gates, and moments are all represented as Python classes. One can also display quantum circuit diagrams, list the contents of a state in various formats, and display the results of thousands of measurements. Because the size of the matrices increases exponentially with the number of qubits, one important advantage of developing a simulator is that the students can extend the theory to situations with three or more qubits, which are intractable by hand, enabling them to handle more complex circuits earlier in the learning process.

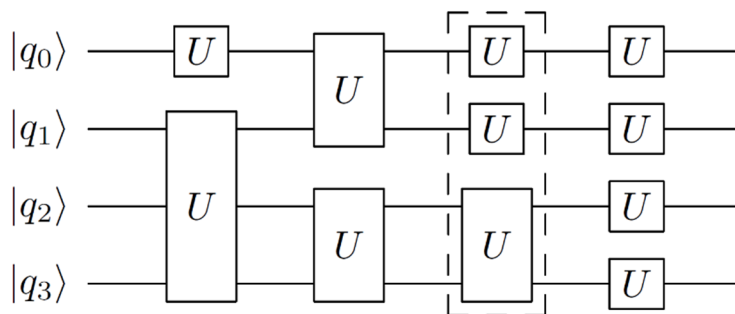


Fig. 1. Conceptual quantum circuit illustrating a moment in the dashed box. Gates are represented by the rectangles labeled “U”.

Our approach requires a foundational knowledge of linear algebra (i.e., vector spaces, linearity and superposition, and matrix and tensor products) and a minimal knowledge of quantum physics. While the instructors allude to quantum physics ideas and experimental results, most of the presentation is more abstract and hardware independent [18], [19]. For the summer research projects, computer programming skills are important, but for our introductory QC course, many STEM students having completed their sophomore year will have seen sufficient computer science.

4. Investigating Advanced Quantum Computing Algorithms

In 2022 we had the opportunity to engage four undergraduate students in various QC projects. The first two students did an independent study during the spring semester and the latter two did two projects during the summer months. A description of these projects and an assessment of what we learned in the process are detailed in this section.

4.1 Understanding the Quantum Fourier Transform and Its Applications

During the spring semester two computer engineering undergraduate students, a junior and a senior, undertook an independent research study with one of the authors. The goal was to determine how well these engineering students could assimilate the necessary background material on QC to begin reading the scholarly literature and to apply it to a selected problem involving the use of the *Quantum Fourier Transform* (QFT). The actual problem was left open-ended, to be determined by student interest and progress in the second half of the semester. The students were assigned readings from a textbook [20], tutorials developed the prior summer, and met with the faculty member once a week for an hour to discuss their progress. The students were engaged in understanding QC through writing Python code and using NumPy to study the matrix operations and in the use of IBM's Qiskit simulator. Several papers were studied in detail, the focus being on the ones that made use of the QFT. The two-qubit implementation of the QFT was analyzed in detail and the three-qubit version was studied. A recent paper that compared the QFT with the classical Fourier transform [21] was assigned for reading. (The fact that this paper was written by a high school student was somewhat motivating to our college students!) The students' prior study of the Discrete Fourier Transform in their signal processing classes was helpful to their study of the QFT. The inverse QFT used for quantum phase estimation was dubbed a binary-to-phase converter—such an abstraction enabled the quantum circuit proposal for solving the NP-hard traveling salesman problem to be understood at a conceptual level [6].

The students were primarily evaluated on their performance in this independent study by a midterm and final report. For the final report the students had to describe the working of a QC algorithm and write some relevant Python code. The senior student was assessed to have an above-average level understanding of the material while the junior student was deemed to have a more average understanding. In terms of actual research progress, we can say the results were rather modest, as the semester's work resulted in grasping an understanding of some current research under the guidance of their faculty mentor. Nevertheless, given the limited time students (and faculty) have to interact during the semester, we conclude the efforts were worthwhile for the insight gained into the extent to which engineering students are able to grasp the basic concepts of QC and to begin reading the scholarly literature. The students also seemed highly engaged by the topic itself and appreciated the opportunity to take part in this independent study.

4.2 Understanding Machine Learning using Quantum Computing

Introduction: Importance of the Research Topic

Advances in machine learning algorithms combined with increased computing power of classical computing platforms (e.g., the move to multi-core and graphical processing units), have enabled machine learning to become a key technique for analyzing the vast amounts of data being generated in the era of 'Big Data'. For example, in computational biology, the development of statistical models of molecular structure is important in drug discovery [8], [10] and for investigating the structure of proteins [22], [23]. Due to computationally intensive nature of these machine learning algorithms (due to the so-called *curse of dimensionality*) combined with the exponential growth of datasets, researchers have begun investigating the potential of QC to increase the performance of machine learning algorithms [24].

Quantum machine learning is a challenging research area, as it requires expertise in both QC and machine learning. For an initial undergraduate research project, a quantum machine learning approach that implements a very simple distance-based classifier [25] was studied. From the machine learning perspective, the classification metric based upon the distance between data points mapped to a Euclidean plane can be understood mathematically with a knowledge of basic high school trigonometry. From the QC perspective, a single quantum gate (the Hadamard gate) was at the core of the algorithm, making the basic concept straightforward to understand. Furthermore, the algorithm highlights the key QC concepts of superposition, interference, and entanglement. While a simple dataset was adopted, the advantage of quantum parallelism is demonstrated as the test data can be compared to both sets of training data at the same time.

What was done

One of the authors of this paper worked on this research over the summer with a data-science student who had just finished his freshman year. The student had completed two programming courses but did not have any prior background in machine learning or QC. Hence, close supervision of the student was required, and coding examples and tutorials were given to the student. With this guidance the student was able to understand and explain the operation of the circuit and write Python code to analyze the circuit. To gain insight on the circuit, the focus was on hand analysis of the circuit and developing Python code that generated the matrices at each stage of the circuit.

A simplified version of the Iris dataset [26] was used to train and test the quantum classifier. Following the work of Schuld *et al.* [25], only two of the four features available for each sample were used, permitting the data points to be represented using complex numbers. As is typical in classification applications, the data were normalized so the data points were on a unit circle and just an angle was needed to specify each point. The data could then be encoded with a quantum circuit consisting of a series of rotation gates combined with CNOT and CCNOT gates to entangle the data. With some effort, it was found that the data preparation circuit could be completely analyzed and understood via hand analysis. This enabled insight to be gained on the critical role played by the data preparation circuit used for quantum machine learning applications.

In addition, the actual circuit, shown in Fig. 2, was implemented on a 5-qubit quantum computer through IBM's Quantum Experience [27]. We were able to reproduce the results of the targeted paper by Schuld *et al.* [25] and to further the work by investigating additional dataset examples that were executed on the IBM quantum computer. In doing so, the student's work highlighted an issue with such a machine learning approach: the required data preparation on both a classical computer and the quantum computer becomes the bottleneck if large amounts of data are to be processed by a quantum machine learning algorithm. The execution of the algorithm on an actual quantum computer emphasized the issue with noise in the data preparation scheme. Furthermore, the research highlighted the shortcomings of this simple approach as some of the additional data points that were used to test the quantum circuit were misclassified.

Lessons learned

In summary this project exemplified the key concepts involved in quantum machine learning. State-vector analysis of the quantum circuit from Schuld *et al.* [25] provided insight on how

quantum parallelism is affected through the superposition and entanglement of qubit states. In addition, the issues associated with the classical-quantum interface, namely how classical data can be prepared and uploaded onto the quantum computer were vividly illustrated in this project. Implementing noise models to analytically understand the effect of noise and how it might be mitigated in the data preparation circuits are some ways that this research can be meaningfully extended by an undergraduate researcher. The work also underscores the need to balance finding a challenging piece of research that is of interest to a student while being within the cognitive scope of an undergraduate. We conclude that this project succeeded on both levels: the topic was quite interesting and motivating to the undergraduate researcher, who was able to successfully present his work at the end of the summer to his fellow students and other faculty mentors associated with the science-wide summer research program.

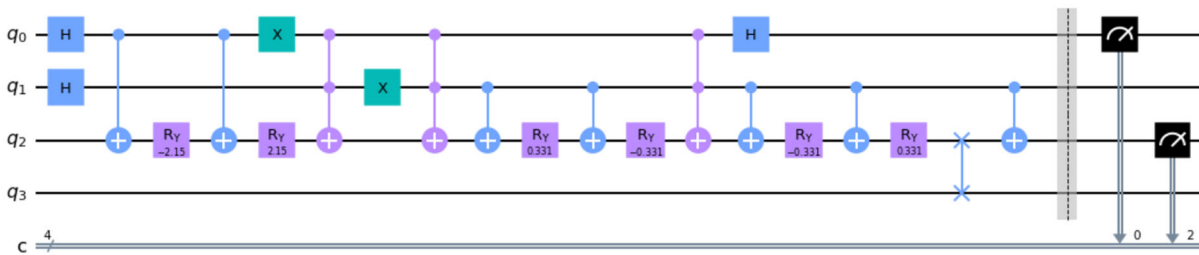


Fig. 2. The quantum machine learning circuit implemented on IBM’s 5 qubit quantum computer.

4.3 Simulating Shor’s Algorithm on the QC Simulator

Introduction: Importance of the Research Topic

In 1994 Peter Shor developed a QC algorithm to find the prime factors of a number in fewer steps than can be achieved by classical computing methods [5]. This was one of the first practical demonstrations of QC and showed that public-key cryptosystems, such as RSA, could be efficiently broken. In these systems information is encrypted using a number $N = pq$, where p and q are two large primes. The information is decrypted with knowledge of p and q . The security of the encryption depends upon the extreme difficulty in factoring N . Shor’s algorithm stimulated the growth of the QC industry. Because this algorithm has a number theory part and a QC part, it takes a significant effort to understand and implement it without predefined functions on a simulator. One of the authors of this paper therefore worked on simulating Shor’s algorithm with an undergraduate physics major during the summer. The student had no prior knowledge of QC. We were able to factor $91=13\times 7$ on the student PC laptop and $143=13\times 11$ with 23 qubits on a Core i7, 2.8 GHz, 16 GB RAM computer.

What was done

During the first month of summer research the student worked on fundamentals of QC, the simulator, and the number theory associated with RSA encryption and Shor’s algorithm. A stripped-down version of the simulator was given to the student. Through a series of exercises mostly adapted from the ‘Try It’ problems in Flarend and Hilborn [20], the student learned to work with object-oriented programming, build up the simulator, execute various quantum circuits, write a measurement method, and interpret the results. Simultaneously the student and advisor worked step-by-step through RSA encryption and decryption following descriptions

from [20] and [28]. Tests of finding the period of modular exponentiation were accomplished with Mathematica and Excel.

Work on simulating Shor’s algorithm occupied the second half of the summer. We followed suggestions by Candela [29]. An example of a quantum circuit for seven qubits is shown in Fig. 3, which is capable of factoring 15. Each rectangle in the diagram represents a matrix whose size increases exponentially with the number of qubits passing into the rectangle. These matrices can become very large in practical problems. To factor larger numbers, effort was needed to understand the structure of the controlled-U matrices that are shown in the diagram as “ $\times a \pmod C$.” The student wrote Python methods to generate a U matrix for arbitrary n and to expand IQFT (Inverse Quantum Fourier Transform) to automatically handle any number of qubits. After making thousands of measurements with the quantum simulator, the student wrote code to mimic what a classical computer would have to do to obtain the factors from the measurements. Histograms of the data were constructed; peaks were found; and a continued fractions method converted the location of a peak into a fraction. This resulted in the period of the modular exponentiation from which the prime factors could be calculated.

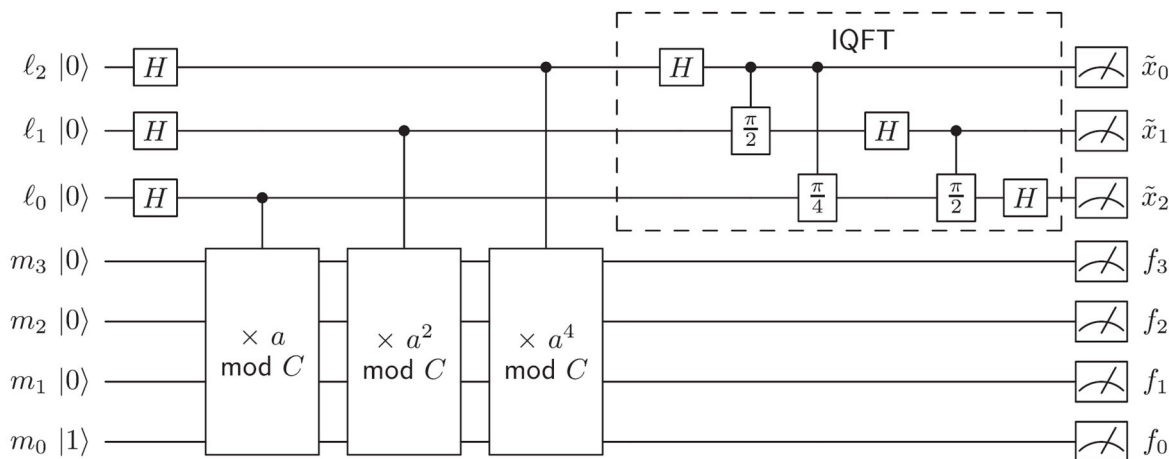


Fig. 8. Quantum circuit to implement Shor’s algorithm with $N = 7$ qubits.

Fig. 3. Reproduced from [29] with the permission of the American Association of Physics Teachers.

Lessons learned

There were two major lessons learned. The first pertained to multiqubit gates. To learn QC starting from a stripped-down simulator, the student successfully built up the simulator to handle any number of qubits and sequential single qubit gates. However, the approach could not handle multiqubit gates (ex. CNOT). The advisor concluded that from the beginning, simulator development needs to be steered in a way that can handle arbitrary multiqubit gates.

The second lesson pertained to the limitations of using the NumPy’s kron function to compute the tensor product of the gates at a given moment of time in the quantum circuit. To factor a number such as 143, $N=23$ qubits were required, forming state vectors each containing 2^{23} ($\approx 10^7$) elements. Because the matrices are $2^N \times 2^N$ sparse block matrices, using kron to form the matrices was an inefficient use of memory and became impossible as N increased in size. Since

the U matrix operated on a subset of qubits, a new function was written, which we nicknamed “superKron,” that applied U to different portions of the state vector, requiring a much smaller matrix to be held in memory. To handle controlled- U operations, a lookup table was generated that could map one bit arrangement to another in order to facilitate the calculation. The author concluded that for simple circuits, where N is relatively small, kron is sufficient and convenient. But when N becomes large, either a superKron function would have to be given to the students or time would have to be spent instructing the students on the mathematics underlying it.

5. Introduction to Quantum Computing Course

This section provides an overview of our initial teaching of an introductory undergraduate course in QC during the spring of 2023. In addition to learning about the standard theoretical topics, the students developed a Python-based quantum computer simulator. They also worked on research projects during the last 2.5 weeks of the semester.

This introductory course was primarily geared towards students at the sophomore, junior, and senior levels majoring in physics, computer science, and electrical engineering. The students collaborated on in-class learning activities using the Jigsaw active learning method [30] and on programming assignments which built towards a simulator. Our hypothesis was that students with a mixture of STEM backgrounds would help each other, making the course more accessible to a broad range of students. The course pre-requisites were only an introductory programming course and Calculus I – the latter was to ensure a basic level of mathematical maturity. Background in quantum physics, linear algebra, and complex numbers was introduced to the students in the course. We assumed students had a working knowledge of high school algebra and trigonometry.

The students coded their own quantum computer simulator over five assignments lasting about 10 weeks, where one or two key software modules were created per assignment. From a software engineering perspective, this bottom-up approach worked well and enabled students to internalize the mapping of the QC concepts to the code. The more proficient coders were then able to extend their simulator as part of the course research project. A key classroom challenge was dealing with the wide range of coding experience, which spanned as little as one semester through those graduating in a few months with a computer science degree. Our plan was to ask more advanced coders to teach others in their group. For some, this was a valuable and rewarding learning experience, but for others, teaching was not their forte. Helping the students understand the value of being a mentor was one of the challenges faced by the course instructors. But in the end, there were half a dozen simulators in use in the room, which could not have been coded without having the interdisciplinary groups.

An important objective of our course was to give students opportunities to investigate open areas of QC research. The learning benefits include improvements in student understanding and cognition, practical experience in developing into a researcher, and retention in STEM fields [31], [32]. Groups of 1 to 3 students worked on research projects after finishing the development of their quantum computer simulator. Students presented their findings to the class during the last week of the semester. At the time of this writing, the final project reports remain to be

submitted, but from the presentations, we can provide some preliminary assessments of several types of projects.

First, students worked on methods to improve the quantum computer simulator. One project explored the `scipy.sparse` library for handling sparse matrices with the goal of handling circuits with a larger number of qubits. Another project explored parallel computation to improve the speed and size of matrices that can be processed. Using the Dask library, the students demonstrated that their simulator could run faster on a conventional multi-core laptop and that the simulator could be scaled relatively easily to computing clusters [33]. They used a quantum adder circuit as a test case.

A second type of project investigated the impact of noise on Grover's search algorithm for two qubits. The students compared real-world noise on multiple IBM quantum computers and looked at a noise model on the IBM Qiskit simulator. They made modifications to the course simulator to inject noise at various points in the circuit. The instructors believe that the speed with which these students could learn to use Qiskit and modify the simulator was the result of the many coding assignments in this course as exercises and for development of the simulator. This type of project could lead to a future project on noisy intermediate-scale quantum machines.

There were topics that we did not have time to cover in the course but were explored in the student research projects. One student looked at the quantum Fourier transform and quantum phase estimation. The Deutsch-Jozsa algorithm was extended. Quantum machine learning was a popular topic as three groups studied the simple distance-based classifier described earlier [25]. The students attempted to reproduce the published work in [25] and to evaluate it further with additional data points and a different dataset.

6. Summary and Conclusions

Quantum computing is regarded by many as the next major computing paradigm that will enable continued exponential improvements in computing efficiency that have been 'enjoyed' over the last five-plus decades of Moore's Law. As such, it is an exciting field of study for both our undergraduate students and their faculty mentors. The thesis of this paper is that QC provides ample opportunities for exciting and open-ended undergraduate research projects. Our experience to date demonstrates that undergraduate researchers, under the close supervision of their faculty mentors, are able to undertake meaningful research. Even if this research is considered somewhat modest compared to the efforts of graduate students, the students are studying legitimate and open-ended research problems in QC, not standard textbook problems.

For summer research, ten weeks are dedicated to intensive study and research, which is considered the full-time job for the student. We find it beneficial to have the prospective research students do some reading on QC prior to the summer. During the actual research period close supervision with meetings at least twice per week during the summer is highly recommended for keeping the research on track. The initial development of the quantum computer simulator in the summer of 2021 gave us insight on how to integrate the development of a simulator in our introductory course as well as use it as a foundation for future research. For one of the summer projects, a freshman data science student required a fair amount of guidance to understand and

write the necessary code to analyze the quantum machine learning circuit. However, he was able to comprehend and successfully present his research by the end of the summer. For the other summer project, a sophomore physics major seemed to be able to grasp the basics of QC relatively quickly and was able to make significant progress in understanding Shor's algorithm and simulating it. Doing research as an independent study during the school year is more challenging. We have found that one meeting a week is usually all the time that can be devoted to supervising student research. Nevertheless, we believe it is a worthwhile activity for keeping students engaged in research, although more modest achievements can be expected.

Our innovation of integrating the development of a Python-based simulator into the standard theoretical approaches taught in a QC course was accomplished during the spring of 2023. We suspect the practical nature of the course has attracted a significant number of students who might not have taken this course if the material were purely theoretical. While we still need to complete a final assessment of the course, we expect to see some benefit in having multidisciplinary groups of students work together. The mini-research projects are expected to improve student learning and motivation. The course should also improve the preparation of the students for doing future QC research.

Acknowledgments

We wish to thank Peter Kutt, students Maggie Sullivan, Kevin Wang, Mike Murphy, and Kyle Geary, and support from the Loyola Hauber Fellowship fund.

References

1. S. Laursen, A.-B. Hunter, E. Seymour, H. Thiry, and G. Melton, *Undergraduate Research in the Sciences: Engaging Students in Real Science*, San Francisco: Jossey-Bass, 2010.
2. D. Lopatto, "Undergraduate Research as a High-Impact Student Experience," Association of American Colleges and Universities, Spring 2010, vol. 12, no. 2.
3. J. Parker, "Undergraduate research, learning gain and equity: the impact of final year research projects," *Higher Education Pedagogies*, vol. 3, no. 1, pp.145-157, 2018.
4. K. Eagan, *et al.* Making a difference in science education: The impact of undergraduate research programs. *American Educational Research Journal*, vol. 50, no. 4, pp. 683–713, 2013.
5. P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," *Proceedings of the IEEE 35th Annual Symposium on Foundations of Computer Science*, Nov. 1994, pp. 124–134.
6. K. Srinivasan, S. Satyajit, B. K. Behera, and P. K. Panigrahi, "Efficient quantum algorithm for solving travelling salesman problem: An IBM quantum experience," *arXiv preprint arXiv:1805.10928*. [Online]. Available: <https://arxiv.org/abs/1805.10928> [Accessed Feb. 5, 2022]
7. M. Schuld and N. Killoran, "Quantum machine learning in feature hilbert spaces," *Physical review letters*, vol. 122, no. 4, p.040504, 2019.
8. C. Outeiral, *et al.*, "The prospects of quantum computing in computational molecular biology," *Wiley Interdisciplinary Reviews: Computational Molecular Science*, vol. 11, no. 1, p.e1481, 2021.
9. R. P. Feynman, "Simulating Physics with Computers," *Int. J. Theor. Physics*, vol. 21, no. 6/7, p 467-488, 1982.
10. Y. Cao, *et al.*, Quantum chemistry in the age of quantum computing. *Chemical reviews*, vol. 119, no. 19, pp.10856-10915, 2019.
11. B. Bauer, *et al.*, "Quantum algorithms for quantum chemistry and quantum materials science," *Chemical Reviews*, vol. 120, no. 22, pp.12685-12717, 2020.

12. S. McArdle, *et al.*, “Quantum computational chemistry,” *Reviews of Modern Physics*, vol. 92, no. 1, p.015003, 2020.
13. F. Arute, *et al.*, “Quantum supremacy using a programmable superconducting processor,” *Nature*, vol. 574, no. 7779, pp. 505–510, 2019.
14. Q. Zhu, *et al.*, “Quantum computational advantage via 60-qubit 24-cycle random circuit sampling,” *Science bulletin*, vol. 67, no. 3, pp. 240-245, 2022.
15. T. N. Theis and H.-S. P. Wong, “The End of Moore’s Law: A New Beginning for Information Technology,” *Computing in Science & Engineering*, vol. 19, no. 2, pp. 41-50, 2017.
16. Z.Y. Chen, *et al.*, “64-qubit quantum circuit simulation.” *Science Bulletin*, vol. 63, 15, pp. 964-971, 2018.
17. R. Schutski, *et al.*, “Adaptive algorithm for quantum circuit simulation,” *Physical Review A*, vol. 101, no. 4, p.042335, 2020.
18. N. Yanofsky and M. Mannucci, *Quantum Computing for Computer Scientists*. Cambridge: Cambridge University Press, 2008.
19. T. L. Wong, *Introduction to Classical and Quantum Computing*, Rooted Grove, 2022.
20. A. Flarend and R. Hilborn, *Quantum Computing: From Alice to Bob*. Oxford University Press, Oxford, 2022.
21. D. R. Musk, A comparison of quantum and traditional Fourier transform computations. *Computing in Science & Engineering*, vol. 22, no. 6, pp.103-110, 2020.
22. A. Perdomo, *et al.*, “Construction of model Hamiltonians for adiabatic quantum computation and its application to finding low-energy conformations of lattice protein models,” *Physical Review A*, vol. 78, no. 1, p.012320, 2008.
23. B. A. Cordier, *et al.*, “Biology and medicine in the landscape of quantum advantages,” *Journal of the Royal Society Interface*, vol. 19, no. 196, p.20220541, 2022.
24. V. Dunjko and H. J. Briegel, “Machine learning & artificial intelligence in the quantum domain: a review of recent progress,” *Reports on Progress in Physics*, vol. 81, no. 7, p.074001, 2018.
25. M. Schuld, M. Fingerhuth, and F. Petruccione, “Implementing a distance-based classifier with a quantum interference circuit,” *Europhysics Letters*, vol. 119, no. 6, p.60002, 2017.
26. R. A. Fisher, “The use of multiple measurements in taxonomic problems,” *Annals of eugenics*, vol. 7, no. 2, pp.179-188, 1936.
27. “IBM Quantum.” <https://quantum-computing.ibm.com/> [Accessed Feb. 5, 2022].
28. J. Lake, “What is RSA encryption and how does it work?” 2021, [Online]. Available: <https://www.comparitech.com/blog/information-security/rsa-encryption/#:~:text=Under%20RSA%20encryption%2C%20messages%20are,known%20as%20the%20private%20key>
29. D. Candela, “Undergraduate computational physics projects on quantum computing,” *American Journal of Physics*, vol. 83, pp. 688-702, 2015. <https://doi.org/10.1119/1.4922296>
30. The Jigsaw Classroom. <https://www.jigsaw.org/>
31. G. D. Kuh, “High-Impact Educational Practices,” in *What They Are, Who Has Access to Them, and Why They Matter*, Washington, DC: Association of American Colleges and Universities, 2008.
32. K. M. Kortz and K. J. van der Hoeven Kraft, “Geoscience education research project: Student benefits and effective design of a course-based undergraduate research experience,” *Journal of Geoscience Education*, vol. 64, no. 1, pp. 24-36, 2016.
33. Dask. <https://www.dask.org/>