The Little Qubitzer:
An Introduction to Quantum Advantage

Book Proposal

by
Adrian German
and
Christina Snyder

Proposal Contents

Description

The UN has named 2025 the International Year of Quantum Science and Technology on the occasion of 100 years of Quantum Mechanics. And it's an exciting time in Quantum Computing as we reach towards the long-sought goal of Quantum Advantage – a critically important concept in Quantum Computing. Quantum computers harness quantum mechanics to compute by different rules than classical computers do. They don't perform operations faster than a classical computer but they perform different operations that a classical computer can't, and sometimes those operations offer a faster route to a solution. It is the purpose of this book to provide access to the details of how this is possible and in doing so to only rely on a string rewriting mechanism (a quantum abacus of misty states) invented by Terry Rudolph with operations no more complicated than basic arithmetic.

At the foundation of our field we have two rewriting systems: Turing machines and lambda calculus. Turing machines are important for many reasons, but especially because of two long-held beliefs regarding computation: the Church-Turing thesis says that everything that is computable can be computed with a Turing machine, although it could take a long time (e.g., exponential time in the size of the input). This correctly suggests that there are problems that cannot be computed—they are called undecidable problems, the most famous of which is the halting problem. Aside from such uncomputable problems, everything else can be computed, and it can be computed using a Turing machine.

The extended Church-Turing thesis is a(nother) foundational principle of computer science that says that the performance of all computers is only polynomially faster than a probabilistic Turing machine. In other words, any model of computation, be it the circuit model or something else, can be simulated by a probabilistic Turing machine with at most polynomial overhead. A probabilistic Turing machine is a Turing machine where the state of the system can be set probabilistically, such as by the flip of a coin. The strong (or extended) Church-Turing thesis says that a probabilistic Turing machine can perform the same computations as any other kind of computer, and it only needs at most polynomially more steps than the other computer.

In 1993, Bernstein and Vazirani showed that quantum computers could violate the extended Church-Turing thesis. Their quantum algorithm offered an exponential speedup over any classical algorithm for a certain computational task called recursive Fourier sampling. Another example of a quantum algorithm demonstrating exponential speedup for a different computational problem was provided in 1994 by Dan Simon. Quantum computation is the only model of computation to date to violate the extended Church-Turing thesis, and therefore only quantum computers are capable of exponential speedups over classical computers. It's equally important to understand that quantum computers would not violate the regular Church-Turing thesis. That is, what is impossible to compute will remain

impossible. The hope, however, is that quantum computers will efficiently solve problems that are inefficient on classical computers. One such problem is the factoring of very large numbers. Another one is simulating nature with computers. Nature appears to be following the laws of quantum mechanics. Quantum mechanics is complex and sometimes classical computers can struggle to crunch the numbers to figure out what nature is doing. But quantum computers play by different rules. Quantum computers don't need to crunch these numbers per se, they can simply mimic nature rather than approximate it numerically like the classical computers need to. And that's because, just like nature, quantum computers are quantum. And the potential here is enormous not just for understanding physics but for designing new materials, and medicines, for instance.

The technology is developing rapidly and today quantum computers exist, access to them via the cloud is affordable, university and industry developed education is increasing, and government funding has been approved to further research and focus on needed workforce development. This proposal outlines a plan for a series of lessons (grouped as chapters in a book and) aimed to teach the basics of quantum algorithms to those who may have little to no background in quantum physics and/or minimal knowledge of coding in Python. Each lesson covers select topics eventually building up to a toolset for tackling more and more challenging quantum algorithms. The sequence of topics and activities is designed with the following three learning outcomes in mind: (a) build a concise but thorough[1] understanding of core terms, techniques and methods from quantum physics used in quantum computing; (b) build an operational understanding of some of the most popular and/or academically important quantum algorithms, and (c) obtain a fluent understanding of how to write code for quantum algorithms, using various emulators and APIs available, such as IBM Qiskit.

This book teaches accurately via simplified cases of quantum advantage using the misty state formalism invented by Terry Rudolph. This [1] is an alternative pedagogical approach that acts as a bridge to the standard quantum computation curriculum but in which the mathematics starts to feel supportive, organic and helpful, instead of oppressive. We should stress that, in our view, there is nothing wrong with mathematics; mathematics by itself is not oppressive. We use mathematics to help describe things going on in the physical world around us. To a physicist, math is an inextricable part of our understanding. Unfortunately, not everyone is good at math, and most have little, if any, training in physics. Terry Rudolph's system can accurately be used to guide our students to the place where we would all like them to be, no less, but going through a stage where they feel that they "really understand" what our mathematics "means" in terms of stuff that goes on in the physical world.

The cases of quantum advantage that we examine are simplified in the sense that they don't require a large number of qubits so, in effect, they can easily be simulated in a popular emulator like IBM Qiskit. We stress from the beginning that every misty state has an exact counterpart in Dirac notation and every calculation can be translated in Dirac algebra. We wrote the book so it addresses at the same time two audiences: both neophytes and experts.

[1] Learn to define, recognize and use (synthesize) for specific purposes.

The experts need reassurance that we don't cut any corners and that students being taught from this book will immediately and actively be able to pick up and read other books such as [2], [3], [4] or even [5] to appear later this year (with No Starch Press). We need to convince the experts that a complete beginner can learn the basics of quantum computing and quantum information science from this book and then make a successful transition to a set of lectures like those that form the playlist at [6].

The beginners also need reassurance that what we are offering is a complete and correct set of techniques and tools and that in the process of mastering those no mathematical rigor is lost (or dropped) for the sake of accessibility. The misty state formalism is simple and very powerful, perhaps surprisingly so. As we showed in several workshops and presentations there comes a time when for the sake of dealing with a wider range of states and algorithms the system needs to be extended slightly and in the process the syntax of conventional math is slowly settling in.

We have approached the presentation from the mindset of maker-centered learning: "What I cannot create I cannot understand" is a good description of that persuasion and a quote from Richard Feynman. Whether we discuss single or multiple qubit systems; entanglement; teleportation; quantum states, quantum gates and measurement; evolving quantum states with quantum gates; quantum circuits; primitives for a quantum processing unit; reversible computation or quantum algorithms, we offer an environment of concrete[2] representations via Python, Qiskit and the misty states formalism. According to Piaget "children in the early years of primary school need concrete objects, pictures, actions, and symbols to develop mathematical meanings." The same is true of students who lack a certain background or affinity for the pure structures of mathematics. This is where the simplicity of the misty state formalism shines through. Piaget also said "[l]ogic and mathematics are nothing but specialized linguistic structures." The misty state formalism can facilitate access to both.

Another quote, from Seymour Papert, is relevant here: "If people believe firmly enough that they cannot do math, they will usually succeed in preventing themselves from doing whatever they recognize as math[3]. The consequences of such self-sabotage is personal failure, and each failure reinforces the original belief. Papert also said: "My basic idea is that programming is the most powerful medium of developing the sophisticated and rigorous thinking needed for mathematics." So in this book we're trying to scaffold the knowledge needed to understand quantum computing and quantum information science starting from computing in Python in a notebook (Google Colab). We build an understanding of the misty state formalism and use it to define, recognize and synthesize (operationally, in Python) the following concepts: superposition, phase, interference, entanglement, quantum gates and quantum circuits, the Deutsch-Josza algorithm, the Grover search algorithm, the Bernstein-Vazirani algorithm (and the phase kickback phenomenon that makes it possible) along with

[2] Our brains need to interact with something in order to create a model of it. As Papert puts it: "You can't think about thinking without thinking about thinking about something."
[3] This resembles Henry Ford's quote "whether you think you can, or you think you can't, you're probably right."

superdense coding and the GHZ game (quantum telepathy via quantum entanglement). We then need to extend the system and present quantum teleportation and the phenomenon known as entanglement swapping (which allows qubits that have never met to become entangled). Our extension of the original misty state formalism includes the discussion of irreducible quantum states, simplest of which are the eigenvectors of the Hadamard gate.

Everything is worked out in either the original or the extended misty state formalism and encoded in IBM Qiskit. As we proceed we ask Qiskit to confirm our calculations and we thus put, at every stage, side by side the actual math with the result of the calculations done using the misty state formalism. Thus by the end of the book the reader can be articulate in both misty states and Dirac algebra. Complex numbers and matrices are not used in the book until the end where it is revealed that the axiomatic definition of quantum gates using misty states is in fact extracting, via those axioms, the columns of the unitaries that define those gates. But the reader doesn't know and does not need to know that as they build proficiency.

So, are our examples in this book actual examples of quantum advantage? Strictly speaking, yes, because they are examples of situations where a quantum circuit can perform better than a classical circuit (and prohibitively so, for any classical circuit). They are very well known circumstances and algorithms having all been introduced in just the last 30 years. The search for industrial quantum advantage is still going on so don't expect these problems to solve problems of the kind and size that an industrial organization can exploit to acquire market share. Quantum computing is described as a disruptive technology. The examples in this book will first disrupt your old way of thinking and propose new paradigms. Learn these well and you will be able to continue to push the envelope of quantum computing after you are hired by one of the many quantum computing companies existing today in the world.

A discussion of more advanced techniques and a glossary of terms ends this book.

About the Authors

Adrian German is a Teaching Professor of Computer Science and Director of Outreach for the IU Quantum Science and Engineering center (IU QSEc). At IU Bloomington he is the recipient of several teaching awards at the departmental and university level. His interests include learner-sighted teaching practices and maker-centered learning. He is the organizer of several successful symposia and conference workshops and has had presentations at various national and international conferences (SIGCSE, SIGITE, ITiCSE, ISSOTL, FIE, ECEL, 2024 APS March Meeting, IEEE Quantum Week (QSEEC) and many others). In 2021 he served as Chair for the Technical Advisory Committee (TAC) on Workforce Development in the Quantum Economic Development Consortium (QED-C) a broad international group of stakeholders from industry, academia, national labs and professional organizations that aims to enable and grow the US quantum industry and its associated supply chain. From 2021 until 2024 he was on the CS2023 Task Force and was responsible for the design and development of the Quantum Architectures Knowledge Unit in the CS2023 Curricular Guidelines. At IU Bloomington he helped create Quantum Technologies for Everyone (QuTE) a self-governed student organization now in its fourth year. He played a central role in both the creation of the Center of Quantum Technologies (CQT) an NSF I/U CRC that is being run jointly by Purdue, Indiana and Notre Dame and the creation of the new accelerated MS program in QIS (2021) at Indiana University in Bloomington.

Christina Snyder is a 2015 CSCI graduate of Georgia Tech. She is currently a high school Computer Science teacher at The Westminster Schools in Atlanta, Georgia. Her introduction to QIS was participating in the initial cohort of “Quantum for All” Teacher Workshops organized by Mark Hannum from AAPT. Christina attended a SIGCSE 2024 Workshop on Quantum Computing with Misty States and is part of a Faculty Learning Community that was established last year within CSTA Indiana. She has numerous other accomplishments, including a faculty award. From 2022-2023, she worked as a Subject Matter Expert for Kenzie Academy partnering with Amazon to upskill their employees, and from 2014 until 2022 she was the Upper School Registrar and CS Teacher at Pace Academy in Atlanta, GA.

Table of Contents

Annotated Table of Contents

See also [7], [10], [11], [12].

1. Python and Google Colab

The only prerequisite we need to get started is the ability to write simple but useful programs in some programming language. The programming language of choice here will be Python. For the integrated development environment of our programs we choose Google Colab notebooks. Access to those is free as long as we have a gmail account (free as well). In this chapter we review the main features of Python and Google Colab including the ability to typeset beautiful mathematics in Google Colab with LaTeX. We work with Python's random number generators and also demonstrate some visualizations to show what's possible.

2. Calculating Frequencies via Computer Simulations

Simple experiments with coins or dice result in probabilistic outcomes. We practice our Python knowledge by writing programs that determine for us such outcomes. Examples of questions we answer in this chapter: "You throw three dice and sum the outcomes. What is the likelihood you don't get a prime number?" or "Craps is a game played with two dice. You decide to play. Each bet is one chip. The goal is to roll a seven (with two dice). The house pays 4 chips plus your original chip if you win. Is this fair? (If not, define fair)." These exercises help us define probability as the ratio of favorable cases over the total number of cases (and our programs do the counting) or as the experimental frequency obtained through a simulation that includes a large number of experiments (via the law of large numbers).

3. Classical Bits with Phase

Classical bits are usually represented as 0 and 1. Quantum bits are more complicated than that but when we measure them we always obtain a classical bit value. Classical bits exist within classical circuits. Classical circuits are made of classical gates. We introduce the basic elements of the misty state formalism by showing how we represent classical bits as black or white blobs. In this notation some quantum gates seem to act exactly like their classical counterparts. We introduce the notion of phase (which does not have a counterpart with the classical bits) and get ready to introduce quantum gates and circuits. A distinctive property of the misty states formalism introduced by Terry Rudolph used here is that it's made entirely out of combinations of white and black blobs, without any numbers anywhere.

4. Quantum Gates: NOT and Z

In quantum theory quantum gates are represented by unitary matrices. In the misty states formalism quantum gates are defined axiomatically, that is, their behavior is given on a case by case basis for any possible input. Note that at this stage inputs are classical as we have not yet defined what a superposition of quantum states is. The axiomatic definition is

essentially equivalent with the definition via unitaries only the reader does not need to know, or worry about that aspect. In this chapter we define and examine the simplest of the one-qubit quantum gates and show how we can define and prove properties using the misty state formalism. We will revisit these gates after we define superposition of quantum states.

5. Superposition: the Hadamard Gate

The first genuinely quantum gate that we will introduce is the Hadamard gate[4]. In the misty state formalism the gate is known as the PETE gate[5]. This quantum gate will help us introduce one the mysteries of the quantum theory: the superposition of quantum states. Quantum theory has many mysteries but this one is the first and the most special. The superposition operator looks like a cloud in which possible outcomes are listed. As we did in the previous chapter multiple quantum gates could be composed (applied in succession) in a quantum circuit. New properties are now available to us as we discover and prove them. It becomes clear that operators (gates, phase and the superpositions operator) are linear although this property emerges naturally and does not bring about any preconceived cognitive overload caused by traditional mathematical syntax (other than the use of simple arithmetic).

6. Misty States and Dirac Notation

This book relies on (and extends) the misty state formalism introduced in [1]. We aim to teach the reader a simple but reliable and accurate method of tracing the evolution of quantum states through quantum circuits for the purpose of implementing and thus understanding and internalizing the operation of various, well-known quantum algorithms. The standard formalism in which these concepts are presented in literature is through Dirac algebra. Our goal is to prove that the method in this book is accessible to those readers with little or no mathematical prowess (or background) and it leads, by the end of the book, to an operational understanding of Dirac algebra. This approach also aims to alleviate (and correct) possible preconceived notions that the misty state formalism might be cutting corners by trading accuracy for accessibility, which simply isn't the case.

7. Introduction to IBM Qiskit

Having established that we aim to offer side by side alternative formulations in misty states and Dirac notation for the sake of convincing the readers that our development is in fact standard (just in an alternative notation) we introduce another such representation tool. We build quantum circuits in Python, using Qiskit. We run the circuits and produce outputs expressed in either analytical form (Dirac notation) or experimental results (measurements) not because we want but because these are the standard output from Qiskit. We continue to explain how these coincide with calculations in the misty state formalism which is our

[4] See Terry's video at https://youtu.be/DZBS5_s7Dsw?si=5JNnQRRKCgxpYjlQ
[5] See https://www.qisforquantum.org/faqs for how the name originated.

main way of presentation and analysis. Being able to calculate something in more than one way and getting the same result increases our confidence that our thinking is in fact correct.

8. The Unit Circle State Machine.

At this stage we can already establish clear parallels with the standard modality [13] used to present quantum computing concepts to Computer Science audiences. We reformulate Andrew Helwer's talk in terms of misty states. This gives us a unit circle state machine which allows us (as it did for Andrew, in his presentation) to reason about one-qubit circuits in a manner that is independent of the underlying representation. We prove various properties of one-qubit gates and one-qubit circuits and confirm our results with Qiskit. Deutsch-Josza can also be introduced and explained using the unit circle state machine [14] independently of the underlying formalism in which we describe and reason about quantum gates.

9. Phase and Interference in One-Qubit Quantum Circuits

It's time to put the misty state formalism to work and we discuss the reversibility of quantum computation. We start by proving that the Hadamard gate is its own inverse. In the process we show how phase contributes to interference. Other properties are proved in this chapter for example: we define what a basis is, name the basis used thus far as the computational basis, introduce the sign basis (to which the Hadamard gates converts) and show that the effect of the NOT gate in the computational basis is the same as the effect of the Z gate in the sign basis. All calculations are done in the misty state formalism.

10. Introducing Quantum Flytrap: Interacting with the Quantum World

Quantum computing is based on quantum mechanics which is describing the real world of the very small. It is difficult to interact with that world directly to build an intuition. The only alternative would be to have access to a physics lab and be able to conduct and interpret experiments. For computer science undergraduates or HS and even middle school students access to such lab equipment is prohibitive. Fortunately we have the Quantum Flytrap [15] which allows us to set up and run a full range of quantum experiments. This virtual lab is presenting itself as a no-code IDE for Quantum Computing and we can run and set up experiments to confirm some of the properties proved earlier in Qiskit or with misty states.

11. Two-Qubit Quantum Gates: SWAP and Controlled NOT

It is time to introduce two-qubit gates. We first introduce controlled-NOT. We show that three such gates carefully arranged in sequence can implement a SWAP gate. The controlled-NOT gate is also the basis for creating entangled states. We define entanglement and design circuits to create the EPR pairs (the set of four maximally entangled quantum states of two qubits). Entanglement will be the basis for some examples of quantum advantage to be presented later in the book. For the time being we decide to focus on just phase kickback.

12. The Phase Kickback Phenomenon

We have established that the classical bits represent the computational basis. We have also shown that any quantum state is a linear combination of the basis states with probability amplitudes as the coefficients of that linear combination. The misty states formalism avoids using any numbers and relies on the symmetry of the superpositions produced by Hadamard gates. The mists (clouds) act as linear combination operators. We have also shown that the effect of the Hadamard gate is to convert from the computational basis to the sign basis. In quantum computing, phase kickback refers to the fact that controlled operations have effects on their controls, in addition to on their targets, and that these effects correspond to phasing operations[6]. In this chapter we prove phase kickback with misty states. Phase kickback is one of the key effects that distinguishes quantum computation from classical computation. It also provides a justification for why qubits would be disrupted by measurements: a measurement is an operation that flips a classical bit (the result) with the flip being controlled by a quantum bit (the qubit being measured). This creates kickback from the bit to the qubit, randomizing the qubit's phase. In the next chapter we use phase kickback to describe, trace and prove a simplified version of the Bernstein-Vazirani algorithm.

13. The Bernstein-Vazirani Algorithm

With this chapter we start the tour of the six cases of quantum advantage. The first stop is the celebrated Bernstein-Vazirani algorithm. In this simple representation we have a black box with an unknown pattern of controlled-NOT gates inside. The goal is to determine that pattern in as few steps as possible. We show that in the classical case the solution is linear in the size of the black box. In the quantum case though we can determine the pattern in just one step. This part, again, does not use misty states. The smallest necessary logical building block here is the phase kickback phenomenon. At this point we want to stress why the misty state formalism, which will be used unchanged, that is, without any coefficients whatsoever throughout most of this book is so effective. We will have to extend this formalism towards the end of the book but that does not diminish the surprising effectiveness of the formalism as initially proposed in “Q is for Quantum.” To understand that part and how and when we need to provide the extension we quote from Terry Rudolph’s FAQ on the book’s website:

> “In the book I used only a single ‘actually quantum’ box, the PETE box. By this I mean it is the only box that has ‘mist-creating’ properties. All the remaining boxes introduced are things that just shuffle colors around—they would be at home in a classical computer for example. Only having to introduce a single new mysterious thing is very nice pedagogically. [...] Now for the [...] genesis of

[6] In quantum computing, operations have the ability to introduce phase changes to quantum states. This is the basis for complex interference patterns and quantum entanglement. When a controlled operation, such as a controlled-NOT gate, is applied to two qubits, the phase of the second (target) qubit is conditioned on the state of the first (control) qubit. Because the phase of the second qubit is being “kicked back” to the first qubit, this phenomenon was coined “phase kickback” in 1997 by Richard Cleve, Artur Ekert, Chiara Macchiavello and Michele Mosca through a paper that solved the Deutsch-Josza problem.

> the whole misty-method: You may wonder whether my reliance on only the single PETE box is limiting, in the sense of [...] does it limit the calculations you could do, and the phenomena you can demonstrate? [...] The answer is that it is *not* limiting, that every calculation can be done (to good-enough accuracy, and again, perhaps with a small overhead) using only PETE boxes and the classical boxes. This is a remarkable mathematical result due to [Yaoyun] Shih, leveraging another powerful result (I think due to Kitaev). There is a citation at the end of the book. A few years ago I was in the middle of pondering this result when I realized I was running late to give a talk at a math camp for 12-14 year olds which was being run in part by my friend PETE Shadbolt. I raced for the tube, and while on it thought about what could I explain to these kids that wasn't the usual jargon-filled quantum fluff. And so here we are."

We said that later in this book we will need to (and we actually do) extend the original, "pure" misty state formalism. For us it happens when we try to implement W-entangled states which rely on the use of controlled-Hadamard gates and arbitrary rotations. It happens again when we discuss teleportation, since the input to the quantum teleportation algorithm is an arbitrary[7] quantum state. Here's how this extension of the formalism is anticipated by Terry Rudolph in the FAQ of "Q is for Quantum" (same place where previous quote came from):

> "[T]he misty formalism is 'universal', in as much as you can use it to do any quantum calculation with only a small overhead. I should reiterate I am not advocating that we should recast all of quantum theory into this formalism. The misty state picture is a good way of getting people to the heart of some nontrivial quantum theory without them having to absorb a boatload of irrelevant math. But that math is *not* largely irrelevant if you actually want to work in the field, it makes many things much easier."

Math, which is essential if you actually want to work in the field, because it makes many things much easier is our ultimate goal here as well. For example we'd like our readers to be ready to read [6], [3], [1] as soon as they master the contents of our book.

14. The Deutsch-Josza Quantum Algorithm

The next case of quantum advantage presented and demonstrated with misty states is the Deutsch-Josza algorithm. We use a variant developed at Virginia Tech by Eddie Barnes and Sophia Economou for their summer school for HS students. This modified problem is known as "Money or Tiger". The user is presented with a situation in which behind each one of two doors there can be: a large amount of money or a hungry tiger. We eliminate the case of two tigers and consider just the other three cases. The two doors have to be opened at the same time or not at all. The user needs to avoid opening a door that leads to a tiger. There is an oracle on the wall that can be asked whether the tiger is behind a specific door. It is obvious

[7] We emphasize then that we cannot clone but we can teleport an unknown, arbitrary state.

that two questions must be asked for the user to be sure they don't risk to be eaten alive by the hungry tiger when the only oracle available is classical. However if we are allowed to use quantum gates to design the oracle one single question is enough to determine if opening the doors is safe or not. That's quantum advantage.

15. The Grover Search Quantum Algorithm

A similar setup that can be referred to as "Money or Tigers" is used to present a simplified version of the quantum search algorithm devised by Lov K Grover in 1996. We pull four cards from a deck of playing cards: a queen and three cards that have a numeric denomination. The cards are face down on the table. It can be shown that without any information we need on average of 2.5 attempts (turning a card face up is defined as an attempt) to find the queen. But we show that if we are allowed to use a (quantum) oracle that is able to recognize (not generate) the solution (that is, to answer the question: is the queen here? as we point to one of the cards that are face down on the table) we can find the queen in one try. The analogy with the previous game from Virginia Tech is in the fact that we can associate the queen with a large sum of money and all the other cards with hungry tigers and we prove that we can open the door with the money (avoiding the other three doors with tigers) in just one try.

16. Entanglement

Quantum entanglement is responsible for super-luminal correlations between particles that are arbitrarily far apart. While entanglement allows for instantaneous correlations, it does not enable faster-than-light communication. In this chapter we describe systems of two qubits and use the misty state formalism to characterize entangled states. We also show how the phenomenon can be set up, observed and understood using Quantum Flytrap. Our simple misty states calculations are in fact equivalent to tensor products.

17. Superdense Coding

In quantum information theory, superdense coding is a quantum communication protocol used to communicate a number of classical bits of information by only transmitting a smaller number of qubits, under the assumption that the sender and receiver are pre-sharing an entangled resource. In our case we send two classical qubits via a one qubit channel and assuming that the two parties share an(other) entangled qubit. Before we start we can prove that the amount of information extractable from one qubit is one bit and that an EPR pair cannot carry any information. Superdense coding is another case of quantum advantage.

18. The GHZ Game (Quantum Pseudo-Telepathy)

In the GHZ game there are three cooperating players (Alice, Bob and Carol) and a referee. The referee is posing a binary question to each player. The players need to respond in kind. The players can formulate a strategy together prior to the start of the game. However, no communication is allowed during the game itself. Classically the players can only win 75%

of the time. When the players adopt a quantum strategy and share a GHZ state they win every time. The GHZ game is a nonlocal game. A nonlocal game is said to display quantum pseudo-telepathy if players who can use entanglement can win it with certainty while players without it can not. The prefix *pseudo* refers to the fact that quantum pseudo-telepathy does not involve the exchange of information between any parties. However, quantum pseudo-telepathy is a real-world phenomenon which can be verified experimentally. We do that with misty states and using Qiskit. Several variations are considered. The GHZ game and other similar games (the CHSH game) are experimental confirmations of Bell inequality violations.

19. $\sqrt{\text{NOT}}$ and S. Irreducible Misty States

We can introduce additional quantum gates and prove properties such as the square root of the Z gate is equal to the S gate (which is now introduced). We can demonstrate that behind any gate and in fact behind any circuit there is a matrix, which Qiskit readily calculates and reports for us for the asking. Since we have access to packages and various other tools that can manipulate and multiply matrices for us (e.g., Wolfram Alpha, `numpy`) we can easily prove that two circuits are equivalent by comparing the matrices reported by Qiskit. We can then prove (or, rather, check) that the square root of the NOT gate is the same as a sequence of three gates, in order: H, S and H. This chapter also introduces $i^2 = -1$.

We also revisit here the one-to-one correspondence between the misty state formalism and Dirac notation. We identify a set of misty states that are irreducible. One of these states is the simplest such state. It also acts as a fixed point for the Hadamard gate. We reveal that physicists refer to such fixed points (very common in computer science) as eigenvectors.

20. Eigenvectors of the Hadamard Gate

We use the first eigenvector of the Hadamard gate to extend the misty state formalism with (real) probability amplitude coefficients. We just ask the question: “Is there any mist which passes through the Hadamard gate such that the probabilities of observing a white ball or a black ball are unchanged?”. And if we write the equations and solve them we find that the ratio of white to black balls in that state has to be an irrational number. This is an opportunity to tell the story of poor Hippasus as well (math education). We then show that this eigenvector of the Hadamard gate is responsible for a minimal non-classical situation which we model and measure also through an experiment in Quantum Flytrap.

21. Quantum Teleportation

Quantum physics allows the teleportation of information without any direct interaction or transfer of mass. In the teleportation protocol, Alice and Bob are respectively given photons B and C belonging to an entangled Bell state. Alice teleports the state of an arbitrary photon A to Bob’s photon C by first performing a Bell measurement of photons A and B. She then sends the measurement result to Bob, who performs a unitary transformation on photon C to complete the teleportation. Because of this last step, teleportation cannot be used to

send information faster than light. We demonstrate the protocol with extended misty states, and then implement it in Qiskit with a procedure that relies on a chapter from [15].

22. Entanglement Swapping

Quantum teleportation is a protocol that enables the transfer of the quantum state of an arbitrary quantum particle from one location (source) to the other (destination) without physically moving the particle itself (however the state is lost at the source). Quantum teleportation has been achieved over very large distances (13,000 kilometers) for example in collaboration between scientists in South Africa and China using a quantum satellite link between China and Austria. The question that we want to address in this chapter is: what happens if the photon to be teleported also belongs to an entangled pair? The answer is very simple although perhaps at first a little surprising: photon A starts out being entangled with a photon D. Thus at the beginning D-A are entangled and B-C are entangled. We find that the Bell measurement creates A-B entanglement, and the result is C-D entanglement. Because of this, the process is called entanglement swapping. This process generates entanglement between two photons that have originated from different sources and have never interacted.

23. Factoring: Shor's Algorithm

This is the first of four chapters that present advanced topics without detailed calculations. The goal here is to generate literacy and the presentation follows the presentation in [17]. The goal for this chapter is for the reader to be aware of the realistic parameters (e.g., number of logical qubits needed) of the procedure. It is also important to have an idea of how many steps the procedure requires and what those steps mean.

24. Quantum Key Distribution

The purpose of this chapter is to introduce the topic (QKD) in such a way that access to a resource such as [18] becomes meaningful. Quantum Key Distribution (QKD) is a method that uses the principles of quantum physics to securely exchange cryptographic keys between two parties, ensuring that any eavesdropping attempt will be detected. The goal is, as before, literacy. BB84 and E91 protocols are described.

25. Quantum Error Correction

Unlike classical bits, qubits are highly susceptible to environmental noise and decoherence. As quantum computers grow larger, the probability of errors accumulating also increases. To build large scale practical quantum computers we need to develop techniques to correct these errors and ensure the integrity of quantum computations, a process known as fault-tolerant quantum computing. Quantum error correction relies on "logical" qubits. Errors are detected through "syndrome measurements" which result in specific correction operations to the encoded qubits, restoring the original quantum state. Some examples of codes used in quantum error correction: Shor codes, Steane codes, surface codes and repetition codes.

As stated before presentation in this part (chapters 23, 24 and 25) is geared more towards literacy of essential terms, concepts, procedures and protocols.

26. Quantum Hardware: Qubit Modalities

We acknowledge and describe briefly nine current modalities for qubits: superconducting, silicon spin, optical (photonics), quantum dots, trapped ions, color centers in diamond, neutral atoms in an optical tweezer array, topological and electron on helium. The goal is to have a minimal appreciation for the physics involved. As noted elsewhere [21] for many years computer science students have been led to believe that they can get by with some knowledge of discrete mathematics and little understanding of physics at all. In the field of quantum computing, and throughout this book, computers that process information according to classical laws of physics are referred to as "classical computers," in order to distinguish them from "quantum computers," which rely upon quantum effects in the processing of information. Computer and communication systems using quantum effects have remarkable properties but a successful transition to them presents challenges. Students and professionals interested in quantum information sciences will need to adopt a different kind of thinking than the one used to construct today's algorithms. It's important then to understand that as a discipline, with respect to quantum computing, computer science is of necessity going back to an age when a strong relationship between physics and computer science existed.

27. Matrices

A brief overview of matrices. For this and the next chapter see [20], [22]. This and the next two chapters are written as summaries for software developers and tech enthusiasts that have not learned the math required for quantum computing in many years or possibly not at all. HS and younger students easily fall in the second category. For them this is an excellent opportunity to start building a solid foundation for the future. Quantum computing is based on a combination of quantum mechanics and computer science and these are built on a foundation of math. We recast some of the things we learned earlier in the book in the language of these three chapters (to show that we've already learned some and that we could easily learn the rest in the traditional syntax from now on).

28. Vectors

A basic review of two-dimensional vectors and vector spaces, elementary linear algebra, and of using matrices to transform space.

29. Trigonometry

A summary of the knowledge that makes [17] accessible.

30. Probability

A basic review of the topic we started with. We also mention Quantum Bayesianism [19].

31. Adiabatic Quantum Computing

Adiabatic quantum computation is an alternative to the better-known gate model (that we have used thus far exclusively in this book) of quantum computation. The two models are polynomially equivalent, but otherwise quite dissimilar: one property that distinguishes AQC from the gate model is its analog nature. Quantum annealing (QA) describes a type of heuristic search algorithm that can be implemented to run in the “native instruction set'' of an AQC platform. D-Wave Systems Inc. manufactures quantum annealing processor chips that exploit quantum properties to realize QA computations in hardware. The chips form the centerpiece of a computing platform designed to solve NP-hard optimization problems.

32. VQE, VQAs **and** QAOA

In quantum computing, the variational quantum eigensolver (VQE) is a quantum algorithm for quantum chemistry, quantum simulations and optimization problems. It’s a hybrid algorithm that uses both classical computers and quantum computers to find the ground state of a given physical system. Variational Quantum Algorithms (VQAs) are a class of hybrid quantum-classical algorithms to which VQE belongs. They leverage both quantum and classical computational resources to tackle problems that are difficult for classical computers. They use a classical optimizer to train a parameterized quantum circuit, aiming to find an optimal solution by iteratively adjusting the circuit's parameters. QAOA (the Quantum Approximate Optimization Algorithm) is a quantum algorithm that aims to solve NP-hard combinatorial optimization problems, which become increasingly difficult to solve precisely as the problem size grows. It is a hybrid quantum-classical algorithm with potential applications in various fields, including logistics, finance, machine learning, and materials science. It is designed to find approximate solutions to combinatorial optimization problems by leveraging a parameterized quantum circuit and classical optimization techniques. VQE and QAOA are both variational quantum algorithms, but QAOA is a specialized version of VQE, primarily used for combinatorial optimization problems, while VQE is more general, applicable to various problems, including quantum chemistry. Their performance varies depending on the problem and available resources. In general, VQE is better suited for problems that require a high degree of precision, while QAOA is better suited for problems with a large array of initial possibilities.

Sample Chapter

Both [7] and [12] are good examples. The first is an actual booklet written for and used in several tutorials and workshops, most notable SIGCSE 2024. The second is a coherent set of milestones and plan for a CS2023 paper [21] that was eventually submitted for publication to the ACM InRoads magazine. Rather than duplicating the contents here (150+ pages) we thought it would be easier to just provide links to those documents.

The documents mentioned above were mainly developed for and used in the summer Boot Camp class [22], [23], [25] offered every year to CSCI undergraduates and incoming graduate students in the accelerated MS in QIS program at Indiana University Bloomington. The program is aimed at STEM students that have graduated with a degree other than Physics.

The Competition

In terms of audience there is only one group (at Quantinuum and Oxford University in UK) that are trying to reach out to teenagers and have proved they can do so effectively. But they work with a different tool (the ZX calculus) and a slightly different pictorial formalism based on category theory. Their purpose is also different. Our approach and theirs are orthogonal so they're more likely to be used together rather than instead.

There is no other book that deals in depth with Terry Rudolph's Misty State formalism other than the original [1]. There are some papers most notably the ones authored by Eddie Barnes with Sophia Economou and Terry Rudolph for their HS summer school at Virginia Tech.

Our approach is currently the most advanced anywhere with respect to Terry's system and it's been tested in classroom, workshops, tutorials and last summer has spawned a Faculty Learning Community (FLC, [25]) for HS teachers in the state of IN (via CSTA Indiana).

This book is meant to be a prequel to conventional Quantum Computing (QC) books. As such it could be used together with No Starch's other QC title, by Andrew Glassner, scheduled to appear later this year [6].

Contact Information

Adrian German
Indiana University Bloomington
700 N Woodlawn Avenue
Luddy Hall 2010
Bloomington, IN
812 855-7860
812 320-8333
dgerman@iu.edu
https://luddy.indiana.edu/contact/profile/?profile_id=210

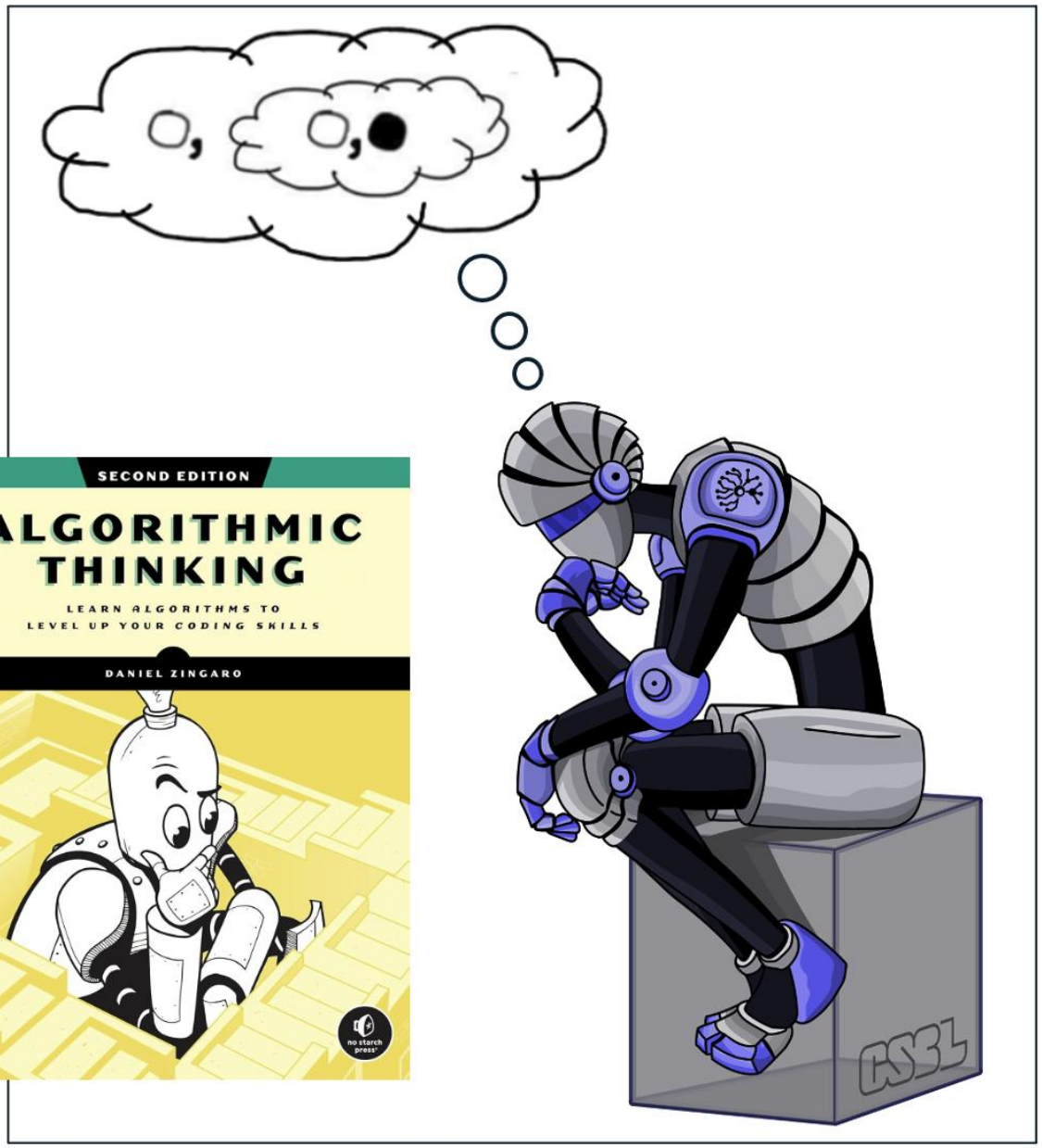
Christina Snyder
The Westminster Schools
1424 W Paces Ferry Rd
Atlanta, GA
678-464-9321
https://www.linkedin.com/in/christinasnyder11/
christina.snyder11@gmail.com
christinasnyder@westminster.net

References

1. Q is for Quantum
2. Rieffel and Polak
3. Nielsen and Chuang
4. Artur Ekert at https://qubit.guide/how-to-cite-this-book
5. https://tinyurl.com/watrous
6. Glassner (No Starch)
7. https://tinyurl.com/qubitzer
8. https://tinyurl.com/misty-ghz
9. https://tinyurl.com/ghz-in-bw
10. https://dl.acm.org/doi/10.1145/3545945.3569845
11. https://tinyurl.com/quantum-curricular-maps
12. https://tinyurl.com/quick-outline
13. Andrew Helwer https://ahelwer.ca/files/qc-for-cs.pdf
14. Advay Mansingka https://www.amazon.com/dp/B09WQ5PDX2
15. Quantum Flytrap https://quantumflytrap.com/
16. https://tinyurl.com/barry-burd
17. https://tinyurl.com/yuly-billig
18. https://ramonawolf.com/qkdtextbook/
19. https://tinyurl.com/q-bism
20. https://tinyurl.com/martin-laforest
21. https://tinyurl.com/acm-inroads
22. https://tinyurl.com/essential-maths
23. https://legacy.cs.indiana.edu/classes/c290-quantum-dgerman/
24. https://tinyurl.com/c290-whatsnew
25. https://tinyurl.com/cqt-slides

We could annotate or further develop this list of references.

Please let us know.


SECOND EDITION
ALGORITHMIC THINKING
LEARN ALGORITHMS TO LEVEL UP YOUR CODING SKILLS
DANIEL ZINGARO
no starch press
CS31