Complex Analysis in Quantum Computing And Mechanics

Rowan Hennessy

November/December 2023

1 Introduction

All the way back in the spring of my freshman year, I was taking CS 2800: Discrete Mathematics, when Dr. Van Zuylen first introduced me to the idea of encryption, and quantum computing. We were learning about RSA Encryption, and how while it may be *theoretically* possible to reverse-engineer, it is not practically possible due to, essentially, computers not being very good at factoring. She followed this up with a mild comment about quantum computing, and thus began a seemingly never ending series of people mentioning this bizarre concept. I heard more about it from Dr. Bracy in CS 3410: Computer System Organization and Programming, then from some friends majoring in Physics, then from Dr. Mészáros in Math 3360: Applicable Algebra, and finally, a brief mention from classmates in Math 4220: Applied Complex Analysis of something called "qubits". So, I decided, for this paper, to explore quantum computing, and by extension quantum mechanics, and how it is built off of applying complex analysis.

2 What is Quantum Computing?

For sake of being thorough, we'll quickly run through the necessary information about quantum computing and quantum mechanics. Note that it's not really possible to communicate an understanding of either of these concepts in a few paragraphs (or arguably an undergrad degree), so there *will* be simplification and jumps in logic, as the purpose of this project is not to teach about quantum computing or mechanics, but to explore two of complex analysis' many applications.

2.0: The Basics:

Quantum computing is, first of all, a completely different approach to computation than normal computing. While normal computing carries out calculations in a linear fashion, one after the next, quantum computing aims to solve problems in *parallel* by utilizing quantum mechanical principles, namely superposition (existing in multiple states simultaneously), entanglement (two units becoming entangled and linked), quantum uncertainty (measuring a quantum system breaks it), and quantum interference (quantum units have unique means of interfering with each other). There are more terms below that will be important throughout the paper.

2.1: Quantum State Representation:

Normal computers utilize a unit called *bits*. Think of a lightbulb- it's either on or off. Quantum computers aim to use a unit called "qubits". Qubits can exist (for simplicity's sake) in the *range* of $[0, 1]$, i.e. holding probabilities of being either state, and do not have a specific value until they are measured (see Quantum Measurement). This new state, of allowing bits to not be binary, lets computers do things that would be otherwise impossible. Being able to calculate things in parallel is just the beginning- this one change to computers, from bits to qubits, will revolutionize the way that our safety systems work.

2.2: Quantum Gates:

Quantum Gates are essentially, functions that take in a qubit and manipulate this superpositional state. In normal computing, we use "logic gates" such as AND, OR, and XOR to process bits. In Quantum computing, we cannot simply AND two qubits, and so we must use quantum gates to preserve information encoded into qubits.

2.3: Quantum Measurement:

Quantum Measurement is the idea that just by *measuring* a qubit's state, you affect it. This is a concept called **superposition**, which allow computers to perform many calculations in parallel. When multiple qubits are in this superposition state, an entire system of them may represent all possible combinations of 0s and 1s for those qubits at once, which a quantum computer can use to examine multiple solutions to a problem simultaneously. Think about how much faster at exams if you could do every problem at once! However, many difficulties arise from measuring qubits. By measuring, you essentially "force" a qubit to choose either 0 or 1, and one of the large problems in quantum computing is finding ways to measure a qubit without reducing the entire system to regular old bits.

2.4: Quantum Entanglement:

Entangled qubits have their states "linked". Essentially, imagine you have two coins. You flip one, and it's heads. You examine the other, and it's heads too! Now, imagine this is true every single time- you flip one coin, and look back at the other, and they're showing the same face. (This is a massive oversimplification, see here for more!)

3 Where Does Complex Analysis Come in?

3.00: Complex Numbers in Quantum Mechanics:

Finding it very difficult myself to understand an application of quantum mechanics without understanding (a tiny fraction of) quantum mechanics first, it's important to begin with complex numbers in quantum mechanics, before moving on to quantum computing itself.

A key aspect of complex numbers is that they are able to *generalize* what real numbers make *binary*. In real numbers, we define them to have a magnitude and a direction- either positive or negative. This direction is binary in real numbers. -1 has a magnitude of 1 and a negative direction. 1 has a magnitude of 1 but a positive direction, so on and so forth. In complex spaces, we are able to generalize this direction. For instance $2i$ has no direction- it's straight up, when thinking of direction "how positive or negative is this number"!

Waves are important in quantum mechanics, as particles have superposition and movement. But how do you use numbers to capture what a moving wave is? Using complex numbers, it's simple to represent a perfect wave as $e^{i\theta}$, where θ is a function of x and time, since we want waves that change over time. This representation of points as complex numbers allows addition of waves- two complex points with magnitude 2 have a sum with magnitude anywhere in $[0, 4]$, which allows us to represent the construction or destruction of waves. Meanwhile, the multiplication of two complex numbers with magnitude 2 will always have magnitude 4, but the "phase angle" of the product will depend on the sum of the phase angles of the two complex numbers.

This generalization of positivity and negativity is really useful- when considering particles that have magnitude and phase angle, as is essential in quantum mechanics, but where we cannot actually *measure* anything about these values, since we have generalized the positivity and negativity, along with interference and magnitude of the interacting complex numbers, we can generalize the addition of two quantum bits, as long as we have a way to represent them in a complex manner.

3.01: Quantum Computing at a High Level:

When first reading about these quantum principles, they sound almost like magic. Or maybe you're remembering your high school physics teacher struggle to explain how light can be *both* a wave and a particle to a room full of confused 16-year-olds. Regardless, they are all possible, primarily due to complex analysis. At the broadest level, qubits are typically described using complex numbers, using vectors in a complex vector space. These complex vectors then can go through an orthogonal complex matrix U (orthogonal meaning "unitary", meaning $U^*U = UU^* = UU^{-1} = I$, where U^* is the conjugate transpose of U), with U representing a quantum gate as defined above. These quantum gates are essential for algorithms relating to superpositional state, and manipulation of a superpositional state relies on complex exponential functions. That, at a *very* high level, is how complex analysis provides the backbone for quantum computing.

3.1: Qubits as Complex Vectors:

The concept behind qubits is that they can be 0, 1, or a combination or superposition of 0 and 1. The representation for superposition of a particle in quantum mechanics is $|\psi(t)\rangle$, a vector. In specifically quantum computing, where we are interested in representing qubits as a linear combination of basis states, a qubit can be said to be in state $\alpha|0\rangle + \beta|1\rangle$, with $\alpha, \beta \in \mathbf{C}$, $|\alpha|^2 + |\beta|^2 = 1$. α and β represent probability amplitudes, relating to the probability of the state being 1 or 0. Why must they be complex numbers? Simply put, complex numbers, with their $ax + bi$ representation, allow just α and just β to capture both the *magnitude* and *phase information* of the represented quantum state. For example, with α, $||\alpha||$ represents the magnitude of a given qubit, and $|\alpha|^2$ the probability. α cannot just be a real number, as the probabilities, when added and subtracted, would not be able to demonstrate both of these properties, which are necessary for representing quantum bits.

The state of any qubit are vectors in a two-dimensional complex Hilbert space (a vector space with an induced distance function, that lets us have length and angle). Projecting qubits into these complex vector spaces allows us to describe them geometrically- any state of a qubit is a point in this complex vector space, with position based on the complex amplitudes of α and β.

3.2: Quantum Gates (contd.)

Now that we've seen how qubits are represented as complex vectors, how do we do anything with these vectors? The answer is quantum gates. A quantum gate which must perform an operation on a *single* qubit can be represented as a 2x2 unitary matrix, with each element of the matrix being a complex number itself. This way, we hold the properties of not changing the inner product or norm of the qubits, so the system's probabilities and magnitudes are not changed by "going through" (matrix multiplying) a gate.

One of the most important methods of analyzing quantum gates comes from Taylor expansions. Since these quantum gates are unitary, they preserve the norm and inner product of our quantum states. Despite how fragile these quantum states can be, however, it can be necessary to approximate or use decomposed quantum gates for more efficient implementation (we do this in normal computation, as well). Taylor expansions give us this method of approximation, allowing us to examine the behavior of a quantum gate in a more local region, or in more specific edge cases. Due to quantum computing's poor performance under noisy conditions, Taylor expansions are very important to understand how a gate will behave under noisy conditions, for instance when the qubit is slightly different than the "perfect" or "expected" value.

Unfortunately, we are still plagued by poor performance in quantum computers due to noise (decoherence that changes the quantum states). However this paper shows (in much deeper detail) how to use channel spectrum benchmarking to examine, using the eigenvalues of the quantum gate matrix U, to determine the noise of a given gate.

3.3: Quantum Entanglement (contd.)

Quantum entanglement is essentially impossible to authentically implement in quantum computing. It simply isn't possible for two completely "unconnected" pieces of information on a computer to be entangled without there being a wire of some sort connecting them. (This is a big reason why we can't have nearly as many qubits on a CPU currently as we can bits). Although, this paper describes a method of doing it locally, but I found it a little hard to decipher, so will not be referencing the points made beyond that. However, delving more into the quantum mechanical aspect, complex analysis provides a framework for how quantum entanglement should operate.

In entangled states, such as Bell States, the state representation involves complex coefficients, and can be represented as $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \frac{1}{\sqrt{2}}(\begin{bmatrix}1\\0\end{bmatrix} \bigotimes \begin{bmatrix}1\\0\end{bmatrix} + \begin{bmatrix}0\\1\end{bmatrix} \bigotimes \begin{bmatrix}0\\1\end{bmatrix})$, with state vectors representing basic states. Essentially, complex analysis allows us to represent entangled states in an effective manner, providing us a framework to build our qubits entanglement function around.

3.4: Quantum Algorithms

Quantum algorithms are a necessity for quantum computers. Already we have hundreds of thousands of algorithms to compute things in our current linear method, and are even able to optimize using threads on different cores. A key benefit of quantum computers is their parallelization, which would go unused if we failed to design new algorithms. Complex Analysis allows us to do so.

Beyond all of the aforementioned applications, using inner products and complex vector spaces to describe states, and transformations through quantum gates relying on complex coefficients, etc., complex analysis is necessary to understand the two most important (besides just being correct) pieces of any algorithmic design: optimization and complexity analysis.

When analyzing optimization for algorithms, it is important to recognize how the algorithm *scales*. We do this using Big-Oh notation currently, in a method called asymptotic analysis. To be brief, the methods we apply to do so in our linear hardware is not effective for quantum hardware. So, using complex analysis techniques like asymptotic bounds and growth rates from complex analysis, along with analyzing analyticity of quantum gates, poles, etc. is the only way to effectively discuss how a quantum gate scales with more inputs.

Using contour integration, residues, and singularities, allows us to analyze the computational behavior of functions arising in quantum algorithms. Since one of the most important behaviors of any algorithm is ensuring correctness, and no memory leakage, using these different techniques to analyze different edge cases, similar to how we examined poles in our class this year, is essential for the design of quantum algorithms.

Complex limit behavior, and examining how different geometries and path integrals evaluate, is another key part of analyzing the behavior of algorithms at critical points. On the same note, limit behavior has become very important in computing optimization of algorithms (i.e. what happens to my limit if $n \to \infty$ would help me analyze complexities if I have infinite data?), in our new age of seemingly limitless inputs and potential datapoints. A quantum computer serves no purpose if the algorithm it runs is unable to calculate the billions, if not trillions, of statistics necessary in Big Data.

4 Closing

4.1: Complex Analysis, Ahead of its Time

While researching for this project, I found a description of complex analysis that I found apt. It was described as a system that "was invented by people who didn't know how it would be utilized". This was in regard to how we call i "imaginary", despite that not really being true (or any more true than every number). Moreover, it's interesting how a mathematical system built so, so long ago is being used now to do unimaginable things. Part of the beauty of complex analysis is how it's a very niche subject, on the surface level. Further down, however, it's an interesting way to redesign the way we interpret numbers, and has very interesting consequences.

4.2: Moral/Ethical Problems

One of the key pieces discussed in every lecture I've been at that has brought up quantum computing has mentioned the potential moral and ethical, downsides. I would be remiss if I didn't devote at least a small segment to potential downsides. It's an amazing technology built on a mathematical system that allows us to represent so many things in only one symbol. Beyond this, however, quantum computing has the potential for extremely high levels of abuse. Quantum computing is *expensive*, not only in electricity, but manpower. Along with that, quantum computing will shatter every form of encryption we currently have, from Reddit passwords to nuclear launch codes. I think it's important to ask while discussing quantum computing- do we need it?

5 Sources/Interesting Reads

Ball States:

Basics of Quantum Computing

IBM Discussing Quantum Computing

Short Blog Post

Quantum Entanglement

Principles of Quantum Mechanics

Benchmarking Quantum Algorithms

Local Changes to make Entanglement Possible

Moralities of Quantum Computing