

Quantum Weirdness: Exploring Quantum Error Correction

Andry Paez

Why this project?

I was a computer science student in community college and was always interested in the computational space. I knew about Quantum computing, but I wasn't familiar with it.

Upon transferring, I made a big decision to switch to Physics and I don't regret it. I realized I thought of computational science as more of a tool and that what I was really interested in was understanding the world.

I was exposed to many topics in a short amount of time and have been looking to do research during undergrad. After considering some options, I decided on finding a union between computational science and physics, where this union I believe is quantum computing. In short, I chose this project because I know this will prove to be a valuable experience moving forward in my career.

What will the project cover?

This project will simulate simple quantum error-correcting codes, like the Shor or Steane code, to protect quantum information from noise and decoherence.

I will try to demonstrate these concepts using a simple toy model that attempts to visualize the quantum effects. I plan to use something like *pygame* or some animation module to simulate this either literally or using a creative analogy.

I will be using Qiskit to simulate noise and quantum gates to explore the effectiveness of error correction techniques, as well as explore different ways we can possibly implement error correction in the future.

Progress Log/What I learned so far

- Used Quantum Error Correction to learn about quantum error correction and how to implement it. Haven't finished the paper yet but have learned some pretty cool stuff such as
 - Classical error codes
 - * redundant encoding
 - * 3-bit repetition code
 - * *distance* and relationship between distance and the number of errors it can correct, $d = 2t + 1$ where t is the number of errors that can be corrected.
 - * notation $[n, k, d]$ with n = number of bits per codeword, k = length of original bit string, and d = distance, or minimum number of errors needed to change codeword.
 - Quantum error codes:
 - * Peter Shor and how he shifted quantum computing from theoretical to possibly practical
 - * representation of general qubit state using Dirac notation and 0 and 1 basis vectors,

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$$

with α and β being complex numbers such that $|\alpha|^2 + |\beta|^2 = 1$. Note that this basically says that the state is a linear combination of the basis vectors (in a state of superposition).

- * Computational space scales by 2^n where n is the number of qubits.
- * A useful way to represent the qubit state is geometrically via the Bloch sphere and spherical coordinates.

$$|\psi\rangle = \cos \frac{\theta}{2} |0\rangle + e^{i\phi} \sin \frac{\theta}{2} |1\rangle$$

where we can see that coherently rotating the qubit will result in an infinite number of possible errors.

Mathematical representation using unitary operation $U(\delta\theta, \delta\phi)$

$$U(\delta\theta, \delta\phi) |\psi\rangle = \cos \frac{\theta + \delta\theta}{2} |0\rangle + e^{i(\phi + \delta\phi)} \sin \frac{\theta + \delta\theta}{2} |1\rangle$$

At first I thought this looked very intimidating, but I think it is essentially just rotating around the Bloch sphere by a certain amount, and arbitrary values of $\delta\theta$ and $\delta\phi$ lead to the infinite number of possible coherent errors.

- * Introduced to the Pauli Basis, but didn't have the brain capacity to continue reading the paper at that moment.
- Explored a bit of IBM videos on YouTube that went over the basics of quantum computing. Also learned some useful things such as
 - Quantum computing makes use of superposition, gates, measurements, interference, and entanglement
 - Superposition is the ability to be in multiple states at the same time
 - Quantum gates are constructs composed of qubits that can alter the state of qubits in a quantum circuit.

One particular example of a gate is the Hadamard gate, which is a single qubit gate that flips the state of the qubit from a well-defined state to a state of superposition.
 - Measurement follows the uncertainty principle where observation means it loses its superposition and collapses into 1 or 0 (collapse of wavefunction)
 - Interference is the process of quantum gates being arranged in a certain way that amplifies correct answers and cancels all incorrect ones.
 - Entanglement is a quantum phenomenon where, in the case of quantum computers, two qubits states become strongly correlated; they become entangled.
 - Qiskit originally worked like an API responding to requests by using CaaS (cloud as a service) to send results via cloud. However, the data transfer was slower than the actual computation, which was a severe bottleneck. They introduced the Qiskit runtime which still uses a pseudo-classical-quantum computer interaction but instead imports

the whole Qiskit program into an environment in the IBM quantum computer.

- Briefly checked the different methods in the Qiskit as well as the documentation.

To-Do List

- Finish the article and **only take the key parts**. I don't think it will be valuable to make this project unnecessarily complicated with the intricate aspects of the paper, but I believe it will be useful for the future and in the creation of the toy model if I sufficiently review the paper.
- Go through the Qiskit module and implement some test programs to simulate gates
- Decide between literal implementation (a circuit diagram/animation) or a fun analogy that follows the same concepts. This could be using pygame or some other module and would be interactive.
- Consider hosting on GitHub Pages, where the site will have a web application for an interactive presentation of the project.
- Figure out complete dev stack for the project as well as find new resources.
- Explore the possible applications of HPC in quantum computing to address the last goal of the project. Find ways that it can help with error mitigation and boost the efficiency of our pseduo-classical-quantum computer system by providing an additional layer of parallel computing.