

QPy – A Quantum Circuit Simulator using Python

Anoushka Chaudhury

Abstract

Quantum computing promises an exponential speedup improvement in the time required to solve certain computational problems by leveraging the principles of quantum mechanics. While the potential power of quantum computers is well-established, their practical implementation poses significant challenges ranging from scalability to decoherence and noise. This paper presents QPy, a Python-based quantum circuit simulator that I programmed and designed to track the quantum state of a model quantum computer through the application of quantum gates to qubits. By performing the requisite matrix calculations, the simulator facilitates the understanding of quantum algorithms through the visualization of the underlying mathematics. This tool enables researchers to explore and implement various quantum protocols efficiently to test and build algorithms.

Key words: Superposition, entanglement, quantum computing, quantum gates

I. Introduction

In the pursuit of solving computationally intensive problems, quantum computing has emerged as a promising platform. Unlike classical computers, which process information in binary bits in the form of 0s and 1s, quantum computers utilize quantum bits or qubits, which can exist in superposition states. Qubits also exploit the quantum property of entanglement. These unique features enable quantum computers to potentially solve certain problems faster than classical computers.

However, harnessing the power of quantum computing requires overcoming formidable challenges. Quantum systems are inherently fragile, prone to errors caused by decoherence and noise. Moreover, designing and implementing quantum algorithms requires a deep understanding of quantum mechanics and complex mathematical operations. Simulating quantum systems on classical computers is exponentially resource-intensive, limiting the scalability of quantum algorithms and parallel computation.

To address these challenges and facilitate the exploration of quantum algorithms, this project introduces a Python-based quantum circuit simulator. It enables researchers to simulate the behaviour of a model quantum computer by accurately tracking the quantum state as quantum gates are applied to qubits. By performing the necessary matrix calculations behind the scenes, the simulator provides a comprehensive view of the mathematical basis of quantum algorithms.

The primary goal of this research is to provide a versatile tool for understanding and implementing quantum protocols. By offering insights into the inner workings of quantum algorithms, QPy empowers researchers to explore novel quantum computing applications efficiently. Additionally, the simulator serves as a platform for testing and validating quantum algorithms in a controlled environment, paving the way for advancements in quantum computing research.

II. Theoretical Foundations

A. Quantum States

A quantum state is a fundamental concept in quantum mechanics that describes the complete configuration of a quantum system. It represents the system's properties, such as position, momentum, energy, or other observable quantities. In quantum mechanics, a system's state is typically represented by a vector in a mathematical space called a Hilbert space [1].

Quantum states possess the remarkable property of superposition, allowing them to exist as a linear combination of multiple basis states simultaneously, until measured. For instance, the spin state of an electron, the polarization state of a photon, or the collective state of multiple qubits within a quantum register are all examples of quantum states.

B. Qubits

In quantum computation, information is stored in the state of a qubit, the quantum generalization of a classical bit. They are fundamental units of quantum information, function analogous to classical bits but with added quantum properties. These properties enable qubits to represent not only $|0\rangle$ or $|1\rangle$ but also superpositions of these states.

Physically, a qubit can be any two-level quantum system, such as electron spin or photon polarization. Their states are described by quantum state vectors, which can undergo evolution over time through quantum operations or interactions with other qubits.

Qubits states are usually written as $|0\rangle$ and $|1\rangle$. These are bases which means that they form a linearly independent spanning set that spans Hilbert Space, implying that they are normalised [3].

A general pure qubit state $|\psi\rangle$ is expressed as:

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

Where α and β are the complex probability amplitudes for each basic state.

The amplitudes $\alpha, \beta \in \mathbb{C}$ obey normalisation such that $|\alpha|^2 + |\beta|^2 = 1$.

C. Quantum Superposition

Vector $|\psi\rangle$ defined in the previous section is said to be in superposition. It can be said that such a superposed state is neither entirely $|0\rangle$ nor entirely $|1\rangle$. The amplitudes alpha and beta, are

directly related to the probabilities of measuring the state $|\psi\rangle$ in any qubit state, given by the Born rule [2]: $|\alpha|^2 + |\beta|^2 = 1$

If a measurement is performed on the qubit state, the state collapses to a definite state of $|0\rangle$ or $|1\rangle$, returning a classical bit 0 or 1, with probability $|\alpha|^2$ or $|\beta|^2$, respectively. However, between measurements, qubits can hold both values at the same time [4].

D. Orthonormal basis

A basis is a set of vectors that span a vector space, meaning any vector in that space can be expressed as a linear combination of the basis vectors. In quantum computing, the vector space of interest is the Hilbert space, which represents the possible states of a quantum system. A basis in the Hilbert space allows us to describe quantum states in terms of a set of fundamental states.

An orthonormal basis is a special type of basis in which the basis vectors are both orthogonal and normalized. Orthogonal vectors are perpendicular to each other, meaning their inner product (or dot product) is zero. Normalized vectors have a magnitude of one, ensuring that each vector has unit length [3]. In the context of quantum computing, an orthonormal basis provides a convenient way to represent quantum states and perform calculations [14], as simplified below.

$$\begin{aligned}\langle 0|0\rangle &= (1^* \ 0^*) \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \langle 1|1\rangle = (0^* \ 1^*) \begin{pmatrix} 0 \\ 1 \end{pmatrix} = 1 \\ \langle 0|1\rangle &= (1^* \ 0^*) \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \langle 1|0\rangle = (0^* \ 1^*) \begin{pmatrix} 1 \\ 0 \end{pmatrix} = 0\end{aligned}$$

In this way, we can represent any quantum state in 2D space using the $|0\rangle$ and $|1\rangle$ bases, through vector addition and scalar multiplication.

Orthonormal bases are fundamental for representing qubits. Since qubits can exist in superpositions of states, we need a basis to describe these states. The two most commonly used bases in quantum computing are the computational basis and the Hadamard basis.

1. **Computational Basis:** In the computational basis, the basis vectors correspond to the orthogonal states of a qubit: $|0\rangle$ and $|1\rangle$, forming the standard basis for representing qubits.
2. **Hadamard Basis:** The Hadamard basis is derived from the Hadamard gate, a fundamental quantum gate used to create superpositions. The basis vectors in the Hadamard basis are $|+\rangle$ and $|-\rangle$, which are superpositions of the computational basis states $|0\rangle$ and $|1\rangle$. These vectors are also orthogonal to each other and form another orthonormal basis for qubits.

E. Quantum Entanglement

Quantum entanglement is a key concept in quantum computing that occurs when particles become interconnected in such a way that the state of one particle is dependent on the state of another, regardless of the distance between them. This allows entangled qubits to share information instantaneously over vast distances, enhancing computational power of quantum computers by enabling complex calculations to be performed [4].

F. Destructive and Projection measurement operations

Destructive measurement refers to a process where information is gained about a quantum system by interacting with it in a way that destroys the original state of the system [5]. By measuring a qubit in a superposition state, since the value collapses to a definite state, information is altered irreversibly, due to the interaction required for measurement.

Alternatively, projective measurement is a method in quantum mechanics used to determine the properties of a quantum system without destroying it [6]. In projective measurement, the system is observed in a way that allows specific properties to be determined accurately without altering the system's state. This type of measurement is crucial to understanding quantum systems while still preserving their original values.

III. Design and Implementation

QPy is a Python-based framework for simulating quantum states and mathematically representing quantum algorithms and vectors. The simulator makes use of a variety of pre-defined quantum algorithms and mechanisms which will be explored in this section.

A. Quantum Gates

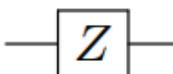
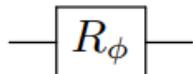
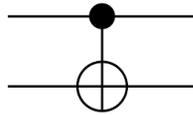
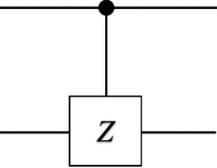
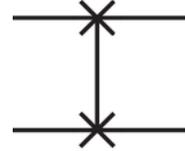
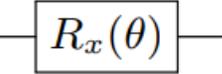
The simulator has been developed by encoding quantum gates in the form of mathematical matrices using python's NumPy library in the *gates* module, which provides a set of common single- and multi-qubit quantum logic gates.

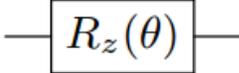
Just as classical algorithms can be represented with a series of Boolean logic gates, quantum algorithms can be represented by quantum gates [7] that are used to manipulate states of quantum systems.

Table 1 is a list of all the currently defined gates in the *gates* module of QPy.

Table 1: List of all the currently defined gates in the simulator.

<i>Operation</i>	<i>Name</i>	<i>Definition</i>	<i>Symbol</i>
------------------	-------------	-------------------	---------------

H	Hadamard Gate	$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$	
X	Pauli - X Gate	$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	
Y	Pauli - Y Gate	$\sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$	
Z	Pauli - Z Gate	$\sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	
S	Phase shift gate	$R_\phi = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{pmatrix}$	
$cnot$	Controlled-NOT Gate	$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$	
CZ	Controlled Z	$CZ = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$	
$swap$	SWAP Gate	$SWAP = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	
$R_x(\theta)$	Rotation - X Gate	$R_x(\theta) = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -i \sin\left(\frac{\theta}{2}\right) \\ -i \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix}$	

$R_z(\theta)$	Rotation - Z Gate	$R_z(\theta) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix}$	
---------------	-------------------	--	---

Additional gates can be added through compositions of existing gates, making use of Kronecker and dot products to approximate an operator with arbitrary accuracy.

Wires in a circuit denote tensor product by padding with identity matrices. To add and make use of gates, a function *add_gate* has been defined as well as a function *apply_kron* to implement the Kronecker product on multiple qubits at the same time.

In order to apply a controlled gate across non-adjacent qubits, another function *control_gate* has been defined with the parameters: gate, control qubit, target qubit and the number of qubits. It pads the circuit with identity operators unless they are being operated on and is, therefore, effective while operating on circuits with a larger number of qubits.

B. Limitations

The quantum density matrix, in the context of quantum computing, is a mathematical tool used to describe the statistical properties of a quantum system composed of qubits. It represents the quantum state of a system, accounting for both pure states and mixed states. Pure states correspond to density matrices with rank one, representing a system in a definite quantum state. Mixed states, on the other hand, arise from a statistical mixture of multiple pure states and are represented by density matrices with rank greater than one. Density matrices are essential for characterizing the behaviour of quantum systems, particularly in the presence of noise and errors, and play a crucial role in quantum error correction and fault-tolerant quantum computation [8]. This simulator does not and cannot account for density matrices and operates purely on quantum states, limiting some of the functions and algorithms that can be executed.

IV. Demonstrations and Applications

Demonstration 1: Quantum Teleportation

Quantum teleportation allows two states that share an entangled pair to transfer an arbitrary quantum state using only classical communication. In order to transfer a quantum state, two distant qubits must be connected via a classical information channel and share a maximally entangled state [9]. Due to the no-cloning theorem described in this section, quantum states cannot be copied and are hard to transport physically due to decoherence and noise.

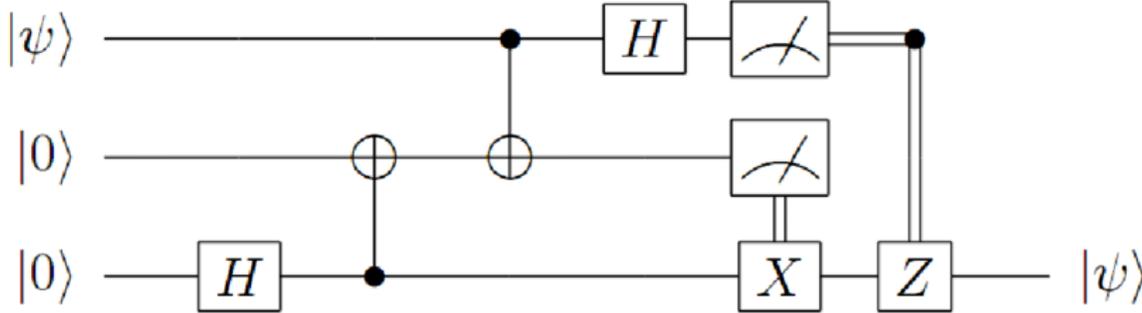


Figure 1: Quantum Circuit for Quantum Teleportation

The circuit for quantum teleportation is shown in Figure 1 [12]. $|\psi\rangle$ denotes the qubit in an arbitrary quantum state. In the first step, the $|0\rangle$ qubits in the second and third line became entangled into a new state by applying the Hadamard and CNOT gate on them. Subsequently, by performing another CNOT and Hadamard gate to their maximally entangled state, to complete the algorithm, there is a local measurement done on the final state of the algorithm.

The resulting two classical bits are sent to a distant qubit. If the projective measurement in the first line outputs a 1, the receiver performs the Z gate on the received state, while if the measurement in the second line outputs a 1, the receiver performs the pauli-X gate on the received state. Finally, the information was changed into the original $|\psi\rangle$ state. This technique is also known as entanglement swapping.

For this demonstration, I prepared an ensemble of qubits in the state $|\psi\rangle = R_x(\theta)|0\rangle$ for various values of $\theta \in [0, 2\pi]$, defined in the function `prep_state_angle(theta)`. I replicated the algorithm described in Figure 1 on QPy to compare the expected and observed outcomes. The expected ratio is modelled by the equation:

$$\frac{1 - \cos(\theta)}{2}$$

The results of the simulated experiment are shown in Figure 2.

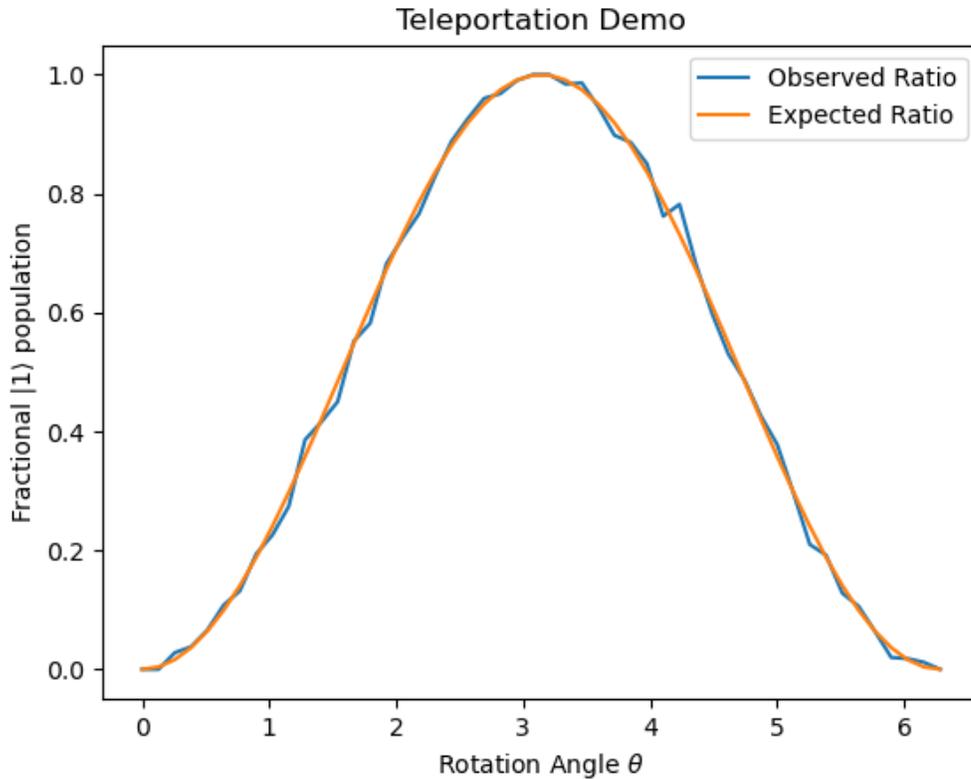


Figure 2: Observed (blue) and expected (orange) fractional $|1\rangle$ populations of measurements which result in $|\psi\rangle \rightarrow |1\rangle$ for the quantum teleportation simulation.

No-Cloning Theorem

An important difference to classical information processing can be found in this theorem as it is a fundamental principle in quantum mechanics that forbids the creation of identical copies of an arbitrary unknown quantum state. However, it is possible to make an imperfect copy of an unknown quantum state or a perfect copy of a known quantum state [10].

The consequences of the no-cloning theorem are that one cannot make copies for error correction, so a special quantum error correction had to be found, which is explored in the following demonstration. Eavesdroppers cannot create copies of a transmitted quantum message, making quantum communication secure.

Demonstration 2: Quantum Error Correction

When qubits are transmitted over quantum channels, they are susceptible to a complex set of errors which can cause decoherence, depolarisation, or information to be lost to the environment. For quantum information transfer to be feasible, the information must be encoded in an error-resistant format using any of a variety of quantum error correction models. This

demonstration implements Shor’s 9-qubit error correction algorithm demonstrated in Figure 3 [13].

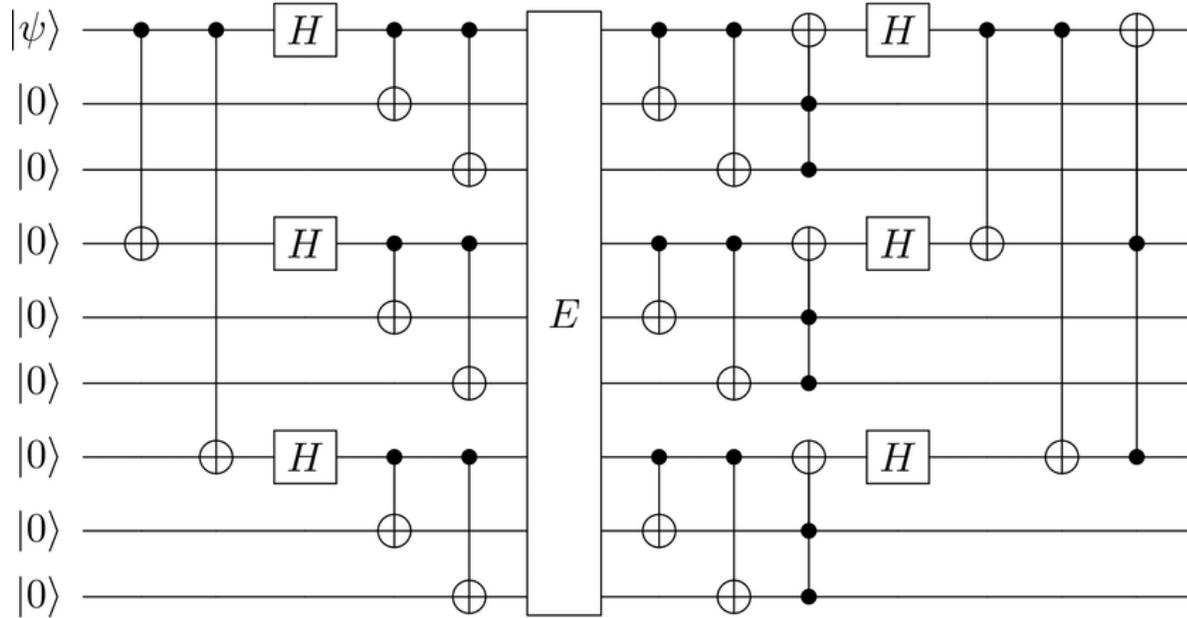


Figure 3: Quantum circuit for encoding and decoding qubits using Shor’s 9-Qubit Error Correction algorithm.

This error correction model encodes a single logical qubit into the product of 9 physical qubits and is capable of correcting for arbitrary single-qubit errors, however, cannot correct multiple errors at once [11] which is demonstrated by the decoherence and erratic results in Figure 6. The algorithm, moreover, cannot directly reveal with qubit values were altered. E represents a channel that induces a single error by applying a random unitary operator.

I replicated the algorithm described in Figure 3 dividing the encrypting stage of the algorithm as *shor_encode* while the decoding stage is defined under *shor_decode*. By deliberately inducing randomized unitary errors into the message after encoding, the simulator tracks the output of the algorithm.

For this demonstration, I again prepared an ensemble of qubits in the state $|\psi\rangle = R_x(\theta)|0\rangle$ for distinct values of $\theta \in [0, 2\pi]$ using the function *prep_state_angle(theta)*. Figures 4 to 6 are plots that compare the expected and observed outcomes with different numbers of induced errors.

The expected ratio is modelled by the same equation:

$$\frac{1 - \cos(\theta)}{2}$$

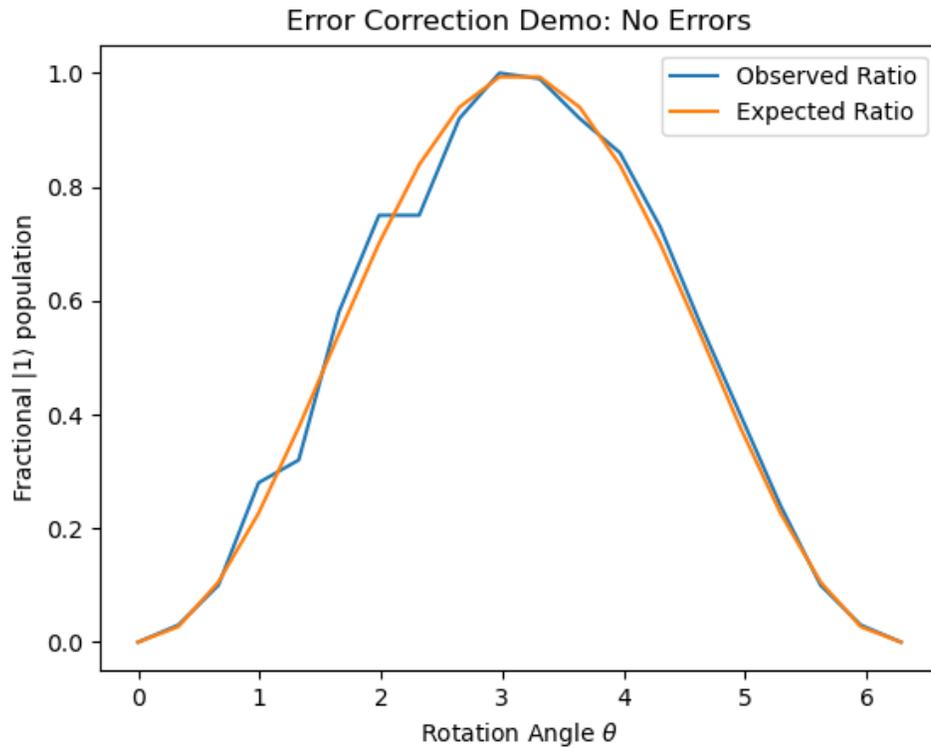


Figure 4: Observed (blue) and expected (orange) fractional $|1\rangle$ populations of measurements which result in $|\psi\rangle \rightarrow |1\rangle$ for the quantum error correction simulation with no occurring errors.



Figure 5: Observed (blue) and expected (orange) fractional $|1\rangle$ populations of measurements which result in $|\psi\rangle \rightarrow |1\rangle$ for the quantum error correction simulation with only one induced error.

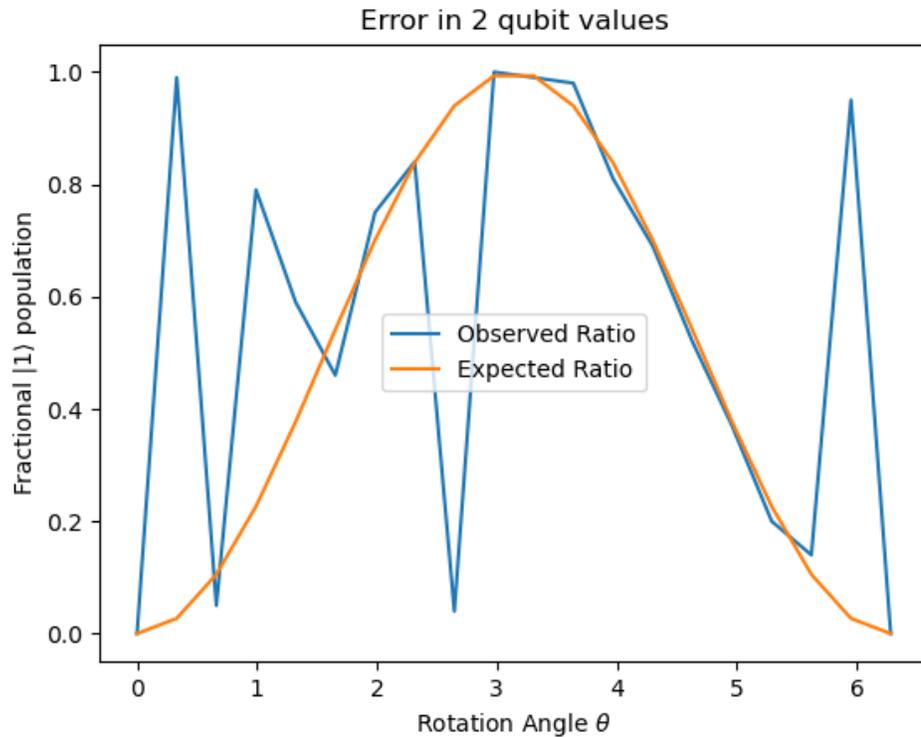


Figure 6: Observed (blue) and expected (orange) fractional $|1\rangle$ populations of measurements which result in $|\psi\rangle \rightarrow |1\rangle$ for the quantum error correction simulation with two induced errors. The information is distorted.

V. Conclusion

The framework includes modules for representing quantum information at the single-particle, multi-particle, and ensemble levels, and a large number of quantum gates for manipulating quantum information. Virtually all components of the simulator are configurable, allowing users to program simulations ranging from simple demonstrations to complex and detailed computational models. The simulator's versatility lays the groundwork for investigating quantum algorithms and protocols, contributing to the advancement of quantum computing research. I hope that the development of this project will stimulate exploration in the field of quantum algorithms.

VI. Acknowledgements

I would like to acknowledge Polygence for giving me this opportunity to work on a project that helped me hone and expand my coding and research abilities. It also exposed me to a new field of study which is exciting for me as it certainly has the potential to be developed on a large scale in the future. I am especially thankful to my mentor, Mr. Ben Bartlett, for guiding me through this project.

References

- [1] "Hilbert Space.". In Encyclopædia Britannica online. Retrieved from <https://www.britannica.com/science/Hilbert-space>
- [2] Marinescu, D. C. (2011). *Classical and quantum information*. Academic Press.
- [3] Bronson, Richard, and Costa, Gabriel B. (2009) *Real Inner Products and Least-Square*, Matrix Methods, 3rd Edition. <https://doi.org/10.1016/B978-0-08-092225-6.50016-4>
- [4] Quantum Inspire. Superposition and entanglement. Retrieved from <https://www.quantum-inspire.com/kbase/superposition-and-entanglement/>
- [5] Braunstein, S. L., and Pati, A. K. (1999). Quantum Information Theory with Continuous Variables [PDF]. Retrieved from <https://quantum.phys.cmu.edu/QCQI/qitd122.pdf>
- [6] Solomon, G. S., Santori, C., and Kuhn, A. (2013). *Single-Photon Generation and Detection: Chapter 13. Single Emitters in Isolated Quantum Systems* (Vol. 45). Elsevier Inc. Chapters.
- [7] QuantumPedia. An Introduction to Quantum Logic Gates. Retrieved from <https://quantumpedia.uk/an-introduction-to-quantum-logic-gates-cee92ba9c1cc>
- [8] "Density Matrix.". In Wikipedia. Retrieved from https://en.wikipedia.org/wiki/Density_matrix#:~:text=The%20density%20matrix%20is%20a,basis%20in%20the%20underlying%20space.
- [9] D. G. Bussandri, G. M. Bosyk, F. Toscano, Challenges in certifying quantum teleportation: moving beyond conventional fidelity benchmark. (2024). arXiv. Retrieved from <https://arxiv.org/html/2403.07994v1>
- [10] Dave Bacon, The No-Cloning Theorem, Classical Teleportation and Quantum Teleportation, Superdense Coding. (2006). Retrieved from <https://courses.cs.washington.edu/courses/cse599d/06wi/lecturenotes4.pdf>
- [11] Aakash Warke et al. The first three-qubit and six-qubit full quantum multiple error-correcting codes with low quantum costs. ResearchGate. Retrieved from https://www.researchgate.net/publication/334634646_The_first_three-qubit_and_six-qubit_full_quantum_multiple_error-correcting_codes_with_low_quantum_costs
- [12] Maximilian Plenert. (2009). Quantum Circuit for Quantum Teleportation [Image]. Retrieved from https://www.researchgate.net/figure/Quantum-Circuit-for-Quantum-Teleportation_fig6_281376174
- [13] Aakash Warke et al. Shor's nine-qubit error correction code [Image]. Retrieved from https://www.researchgate.net/figure/Shors-nine-qubit-error-correction-code_fig5_334634646
- [14] Nielsen, M. A., & Chuang, I. L. (2000). Quantum Computation and Quantum Information [PDF]. Retrieved from <https://profmcruz.files.wordpress.com/2017/08/quantum-computation-and-quantum-information-nielsen-chuang.pdf>