Reflection on coding Density matrix using Bloch sphere

PHYS161 Project1.b

Sarah Dweik

Earlier this year, I worked on a project that helped me strengthen my understanding of quantum computing and programming. In February 2025, I created a couple of Jupyter notebooks that used Qiskit to explore quantum circuits with built-in gates and Bloch sphere properties [1]. At first, I was mainly using the functions provided by Qiskit, such as calling `q.H` to apply a Hadamard gate. However, as I continued, I wanted to see what was happening behind the scenes. Instead of only relying on Qiskit, this project helped to start encoding qubits as vectors and applying gates through matrix multiplication and inner loops. This approach gave me a clearer view of the mechanics of quantum circuits rather than just using pre-made functions.

Through this process, I also came across other tools such as QuTiP. Although it seemed useful, I decided not to use it because I wanted to build the properties from the ground up, rather than depending on another layer of built-in functions. This made the work slower, but it also made me more confident in understanding what was happening at each step. Since most of my programming background has been in Java, working with Python was another area of growth. In fact, this was my first real experience using Python to take user input, which is something I had never done before. It was a small but important step in becoming more comfortable with the language.

Visualization was another part of the project where I learned a lot. There are many sophisticated libraries for representing the Bloch sphere, but I decided to take a simpler approach. By defining the spherical coordinate system, I was able to produce a basic but functional visualization. It was not perfect, and I know there are better methods available, but it allowed me to check that my understanding of the qubit states was correct. For a first attempt, it was enough to demonstrate that the concepts could be represented in a straightforward way.

I also experimented with using artificial intelligence as part of this project. While AI did not play a major role, I used it occasionally to compare methods and see if there were more efficient ways of doing certain tasks. For example, when working with transposes or determinants, AI suggested shorter code using existing Python libraries. Even though I mostly stayed with my manual implementations, I kept some of the AI's suggestions as comments in the code. This gave me a way to remember alternative approaches and to think about how optimization could be applied in future projects.

Looking back, this project helped me in several ways. I gained a deeper understanding of the mathematics behind quantum circuits, practiced Python in ways I had not done before, and built simple but useful visualizations. I also saw how AI can act as a supportive tool, offering suggestions without taking over the work. Overall, it was a good starting point for continuing to explore quantum computing and for improving both my coding and problem-solving skills.

References

[1] Project Notebooks (Feb 2025). Available at: https://github.com/Sarah-Dweik/Qiskit-qBraid-Intro-to-Quantum-software