Quantum Computing for Quantum Chemistry
Module 3

Anthony Slawski

February 2026

This lab focused on simulation in quantum computing, both as classical emulation of quantum circuits and as time evolution implemented with qubits. The first part considered classical state vector simulation. A computational basis bitstring was mapped to an M index complex tensor representing the full 2^M dimensional state. This showed how amplitudes and measurement frequencies are stored and why memory grows exponentially with qubit number.

Single Pauli strings were then applied directly to the state vector using tensor contractions with the Pauli matrices reproducing simple quantum circuits only using linear algebra. Frequencies obtained from $|V_{i_1 \ldots i_M}|^2$ matched the expected computational basis outcomes, which confirmed that the indexing and contraction logic were correct.

The same tasks were repeated using Qiskit. Circuits were initialized to chosen bitstrings and layers of Pauli gates were appended from a PauliString description. Statevector simulation and shot based sampling with the Aer simulator gave the same results as the manual tensor approach.

The next part considered unitary time evolution under a Pauli string $\exp(-it\hat{P})$. Basis changes converted $\hat{X}$ and $\hat{Y}$ operators to $\hat{Z}$, followed by parity accumulation with CNOT ladders, a single $R_z(2t)$ rotation, and uncomputation.

Finally, Trotterisation was used to approximate time evolution generated by a sum of Pauli strings. First and second order Trotter-Suzuki formulas were implemented and applied to the H_2 qubit Hamiltonian. Increasing the Trotter number reduced approximation error but increased circuit depth, which increases hardware noise. A one qubit example with $\hat{Q} = \hat{X} + \hat{Z}$ was used to study the error scaling by comparing Trotterized evolution to the exact solution.