QCIS Summer School 2023: Introduction to Numerical Techniques for Quantum Circuit Simulation

Matthew J Reagor

CTO SQMS Fermilab, Rigetti
August 2023

1 Evolving A Single-Qubit Quantum State via Unitary Gates

Our goal in this Exercise is to write a simulator that computes the resulting single qubit state of a quantum program consisting of multiple unitary gates are applied to an initial state. The simulator will take two inputs: (1) an initial single-qubit state $|\Psi\rangle_0$, and (2) a list of length-M single-qubit gates $\{U_0, U_1, ...U_M\}$ with $U_k \in \{X, Y, Z, S, T\}$ that we call a program P. The simulator should return the final single-qubit state vector $|\Psi\rangle_M$. Basing your work on the numerical computing tools in the Python package `NumPy` is strongly encouraged.

Use your simulator to compute the result of $|\Psi\rangle_0 = |0\rangle$ and $P = \{X, S, Y, T, X, Z, Y\}$. How does the compute time of your simulator generally scale with the length of the program M?

2 Extending to Simulator to Multi-Qubit Programs

Our next goal is to evolve an N qubit register under the action of a program P comprising a list of single-qubit gates $\left\{U_k^i\right\}$, with the superscript-i denoting the target qubit for the operation. Without two-qubit gates to generate entanglement, efficient representations for the total state vector exist. But, we will soon be including two-qubit gates, so we encourage you to represent the qubit register using a Kronecker product (see `numpy.kron`).

Use your simulator to compute the result of $|\Psi\rangle_0 = |0\rangle \otimes |0\rangle \otimes |0\rangle$ and $P = \left\{X^0, Y^1, Z^2, Z^0, Y^1, X^2\right\}$. How does the compute time of your simulator generally scale with the size of the qubit register N?

3 Adding Instructions to Generate Entanglement

Let's now add entanglement to our simulator. A controlled-NOT operation can be represented as...

4 Making Quantum Measurements On Resulting State

On real hardware, we have access to single-shot measurements...