Contents

List of Figures

Chapter 1

Introduction

1.1 Interesting Quantum Programming Languages

Qiskit, which is the most prominent and based around Python. q1tsim, which is based on Rust and the docmentation is mostly in Rust but apparently can be used in Python as well.

1.2 Qubit-Basics

The two orthogonal z-basis states of a qubit are defined as $|0\rangle$ and $|1\rangle$. When we talk about the qubit basis states we implicitly refer to the z-basis as the computational basis states.
The two orthogonal x-basis states are

$$|+\rangle = \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) \qquad |-\rangle = \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle) . \tag{1.1}$$

The two orthogonal y-basis states are

$$|R\rangle = \frac{1}{\sqrt{2}} (|0\rangle + i\,|1\rangle) \qquad |L\rangle = \frac{1}{\sqrt{2}} (|0\rangle - i\,|1\rangle) . \tag{1.2}$$

1.3 Eigen, Eigen, Eigen

Quantum mechanics uses the concept of eiegenvalue and eigenvector and extends it to linear operators. An operator is a thing that transforms a function into antoher function - analogously to how a matrix transforms a vector into another vector. An "eigenfunction" of some linear Operator A, is then a function that satisfies

$$A\Psi = \lambda\Psi, \tag{1.3}$$

where Ψ is some function, A an operator, and λ some real value. Differentiation is a linear operator. Define the differential Operator

$$\hat{D} = \frac{\partial}{\partial x}. \tag{1.4}$$

What is its eigenfunction? That is,

$$D\Psi = \lambda\Psi. \tag{1.5}$$

If you write the operator out, you have

$$\frac{\partial \Psi}{\partial x} = \lambda\Psi. \tag{1.6}$$

The solution to this differential equation is

$$\Psi = e^{\lambda x}. \tag{1.7}$$

So any such exponential function is an eigenfunction of $\hat{D}$. An example related to quantum mechanics specifically, you have the time-independant Schrödinger equation

$$H\Psi = E\Psi. \tag{1.8}$$

The energies here are the eigenvalues, and the energy states are described by the corresponding eigenfunctions, hence eigenstates.

1.4 Quantum Fourier Transform

The quantum Fourier transform (QFT) transforms between two bases, the computational (Z) basis, and the Fourier basis. It can be written as a mapping

$$|x\rangle \mapsto \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} \omega_N^{xy} |y\rangle \tag{1.9}$$

or as an unitary matrix

$$U_{\text{QFT}} = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \omega_N^{xy} |y\rangle \langle x| \,. \tag{1.10}$$

The H-gate is the signle-qubit QFT, and it transfroms between the Z-basis states $|0\rangle$ and $|1\rangle$ to the X-basis states $|+\rangle$ and $|-\rangle$. In the same way, all multi-qubit states in the computational basis have corresponding states in the Fourier basis. The QFT is simply the function that transforms between these bases.

$$|\text{State in Computational Basis}\rangle \xrightarrow{QFT} |\text{State in Fourier Basis}\rangle \tag{1.11}$$

$$QFT\,|x\rangle = |\tilde{x}\rangle\,. \tag{1.12}$$

$$\tag{1.13}$$

In the computational basis, we store numbers in binary, just like a Johnson Counter. The first qubit has the highest frequency and changes every iteration, while the last one has the lowest frequency. In the Fourier basis, we store numbers using different rotations around the Z-axis. To encode the state $|\tilde{5}\rangle$ on 4 qubits, we rotate the leftmost qubit by $\frac{5}{2^n} = \frac{5}{2^4} = \frac{5}{16} \times 2\pi$radians. For the next qubit the numerator is doubled, thus $\frac{10}{16} \times 2\pi$. This leads to the result, that now the first qubit has the lowest frequency, while the fourth qubit has the quickest. Note that in the Fourier basis, we are no longer counting binarily.