The Quantum Fourier Transform and Phase Estimation

George Watkins and Einar Gabbassov.

August 4, 2020

Abstract

In this report we explore the Quantum Fourier Transform (QFT) and Quantum Phase Estimation. To do so we introduce the mathematical framework needed to understand quantum algorithms and programs. We build towards a concrete implementation by formulating the problems in the language of quantum computing, drawing the connection with Fourier Analysis. We provide an implementation of the QFT that is executed on one of IBM's physical quantum computers.

1 Introduction

What is Quantum Computing?

The theory of Quantum Mechanics emerged at the beginning of the 20th century when physicists observed that nanoscopic entities exhibit both particle and wave like behaviour [3]. Driven by the intuition that simulating quantum systems is very hard for classical devices, Richard Feynman and others [9] suggested that quantum mechanical properties could be exploited to perform calculations faster than classical computers. Though the first theorized applications of quantum computing were in the field of simulating quantum systems themselves, other quantum algorithms were discovered that achieve exponential speedup in certain tasks [12]. One of the most renowned quantum algorithms is perhaps Shor's integer factoring algorithm [16], which achieves exponential speedup over the fastest known classical counterpart. One of the key building blocks of Shor's algorithm is the QFT. Without QFT's log-of-input-size running time, such speedup would not be possible.

Fourier Analysis Background

A Fourier Transform takes a function and expresses it in terms of frequencies, like a musical chord can be expressed as its constituent notes. This is often defined as converting the function from the time domain to the frequency domain. The inverse operation, going from a frequency domain to the time domain, is the Inverse Fourier Transform, which, in the musical analogy, is similar to drawing the soundwave of a chord given its notes. In particular, an Inverse Discrete Fourier Transform (DFT^{-1}) accepts a function in frequency representation, as n coefficients of a trigonometric polynomial, and returns n equally spaced time samples.

Motivation

One might wonder why the QFT and Phase Estimation are interesting. Let us begin with the QFT, which is often used as a building block in bigger quantum computing algorithms [8]. When it was first introduced [1] it was described as a tool useful for integer factoring, possibly its most celebrated application. Now, it is an integral component (together with its inverse form) of several quantum algorithms in machine learning [11, 14], number theory [16, 6], linear algebra [5, 7] and possibly more. The reason for the success of the QFT lies in how it operates on the complex numbers used in quantum computing representation; it takes their phases, which we will see are are not directly accessible, and makes them interact causing tangible effects. Thus, other algorithms can encode results in phases and use the QFT methods to extract the needed information [2]. In Section 3 we will delve in the details of the direct QFT's calculations. The inverse operation of the QFT is also of interest. In Section 4 we will explore quantum Phase Estimation, a real problem quantum computers can solve using the inverse of QFT (QFT^{-1}).

2 Quantum States

In this section we develop general intuition about quantum computing and set up necessary mathematical concepts.

The Qubit

Just as in classical computing (CC) quantum computing (QC) uses bits to manipulate data. The difference between the two paradigms is that QC relies on principles of quantum mechanics such as the principles of superposition and entanglement [10] whereas CC is a sequential progression of states. Quantum properties allow for probabilistic simultaneous computation of some problems. The entire QC paradigm is modeled by linear algebra. For example, in the following paragraph we will define a quantum state, state transitions as vectors and matrices in complex vector space.
Let us introduce the necessary notation used in QC, we start with a qubit. Just as in CC, a qubit can be in states 0 or 1 upon measurement. This can be written as follows

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$
$$|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

The definition above gives a linearly independent basis. Hence, we can represent a state vector $|\psi\rangle$ as linear combination of $|0\rangle$ and $|1\rangle$, $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$ for $\alpha, \beta \in \mathbf{C}$. We have another condition, which also fits the probabilistic interpretation of QC,

$$|\alpha|^2 + |\beta|^2 = 1.$$

For example, suppose we want to switch the state of a qubit by negation. This is done with the Pauli X gate

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \text{ and } ||X||_2 = 1 \text{ and } X^*X = I.$$
$$X|0\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |1\rangle.$$

In addition to the above, we introduce an important operator called Hadamard gate H. Hadamard gate turns a $|0\rangle$ or a $|1\rangle$ into a superposition. This gate is one of the most important in QC.

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \text{ and } HH = I,\ H^*H = I. \tag{1}$$

The operation of H gate yields the following

$$|x\rangle \xrightarrow{H} \frac{|0\rangle + (-1)^x |1\rangle}{\sqrt{2}} \quad \text{for } x = \{0, 1\}, \tag{2}$$

$$\alpha |0\rangle + \beta |1\rangle \xrightarrow{H} \alpha\Big(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\Big) + \beta\Big(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\Big). \tag{3}$$

Multiple Qubits

The algorithms we cover in this paper operate on multiple qubits. To discuss how quantum computations operate on more than one qubit, we need to introduce a representation of quantum information: multiple qubit states.

Before we give a general definition, we start with a simpler case. Our general definition in the next section will build upon it. Suppose we have 2 qubits. As seen in the previous subsection, each individual qubit has two measurement outcomes, 0 or 1, associated with states $|0\rangle$ and $|1\rangle$. If one qubit is in state $|0\rangle$ and the next in state $|1\rangle$ we describe the state as whole using the tensor product:

$$|0\rangle \otimes |1\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}.$$

For brevity we will use the following notation,

$$|01\rangle := |0\rangle \otimes |1\rangle$$

As another example suppose we have 3 qubits, the first in state $|0\rangle$ the second in state $|1\rangle$ and the third in state $|0\rangle$. Such a system in described in the notation of the field as follows,

$$|010\rangle := |0\rangle \otimes |1\rangle \otimes |0\rangle = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

We can generalize this definition to n qubits. Let n qubits be associated with $|x_0\rangle, \ldots, |x_{n-1}\rangle$ respectively, where $x_0, \ldots, x_{n-1}$ are either 0 or 1. The state of such qubits as a whole is given by $|x_0, \ldots, x_{n-1}\rangle$. We call these states *basis states*. A remark that will be more interesting later is that tensor products of basis states always have all zeros in all entries except for one. The states described above resemble classical ones, because their individual qubits are in one of two choices. In the previous subsection we explained for single qubits how such "binary" basis states from more interesting ones, through superposition, the next section will do the same for multiple qubits. Now, using all of the above we are ready to proceed to the first quantum algorithm.

3 QFT

The QFT is a transformation on multiple qubits. It takes qubit state information, encoded in the form of a complex vector, and it applies phase manipulations to it. In its most basic form, the QFT is acting on qubits that are in basis states, from which the QFT creates a superposition, similar to the Hadamard operator. When applied to a state that already is in a superposition, the phase rotations induced by the QFT produce more complex behaviour. When the phases of two basis states are close, their respective coefficients add up, when the phases are further apart they subtract from each other.

In the rest of the section we look at more general qubit states which are inputs and outputs of the QFT and Phase Estimation. We give a formal definition of QFT, while attempting to guide the reader in understanding its operation and scope, with emphasis on how it relates to Fourier Analysis. The exposition is complemented with worked examples of the calculations performed the QFT and their quantum circuit implementation.

Formal Definition

To present the formal definition of QFT, we begin by introducing a computational basis, that is the sequence of individual qubit outcomes. Then we give a precise definition of a general multiple qubit state. We will then use these definitions in that of the QFT.

Let us proceed by introducing the basis vectors of given by 3 qubits; the definition naturally generalizes to other numbers of qubits. Using the notation defined in section 2, the basis is:

$$\beta_{2^3} = \{|000\rangle, |001\rangle, \ldots, |111\rangle\}.$$

Expanding out the tensor products yields a representation of β_{2^3} from which we directly see that it's an orthonormal basis (the standard one):

$$\beta_{2^3} = \{ \begin{bmatrix}1\\0\\0\\0\\0\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\\0\\0\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\\0\\0\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\0\\1\\0\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\0\\0\\1\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\0\\0\\0\\1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\0\\0\\0\\0\\1\\0\end{bmatrix}, \begin{bmatrix}0\\0\\0\\0\\0\\0\\0\\1\end{bmatrix} \}.$$

For convenience we also enumerate the elements of β_{2^3} using the integers from 0 to $2^3 - 1$. With a slight abuse of notation, at times used in Quantum Computing:

$$|0\rangle \mapsto |000\rangle, |1\rangle \mapsto |001\rangle, \ldots, \left|2^3 - 1\right\rangle \mapsto |111\rangle. \tag{4}$$

This gives us all the tools to give the definition of QFT. We build it starting from QFT_{2^n}, as a special case, mapping from a basis state $|x\rangle \in \beta_{2^n}$ to a superposition of basis states.

$$\mathrm{QFT}_{2^n} : |x\rangle \mapsto \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{\frac{2\pi i x y}{2^n}} |y\rangle. \tag{5}$$

Note how x and y are integers, but $|x\rangle$ and $|y\rangle$ are basis states contained in β_{2^n}. This is by using the enumeration given in Eq (4), generalized to n qubits. All coefficients of $\mathrm{QFT}_{2^n} |x\rangle$ have the same square magnitude, hence they form a *uniform superposition*. For example let us apply the QFT on the state $|10\rangle$,

$$\mathrm{QFT}_{2^2} |10\rangle = \frac{1}{\sqrt{2^2}} \big(e^{0i} |00\rangle + e^{\pi i} |01\rangle + e^{0i} |10\rangle + e^{\pi i} |11\rangle \big) \tag{6}$$

Now we extend QFT_{2^n} to act on a general n qubit state, that is a normalized vector $|\psi\rangle \in \mathbb{C}^{2^n}$,

$$|\psi\rangle = \begin{bmatrix} \psi_0 \\ \psi_1 \\ \vdots \\ \psi_{2^n-1} \end{bmatrix} = \psi_0 |0\rangle + \psi_1 |1\rangle + \cdots + \psi_{2^n-1} |2^n - 1\rangle . \tag{7}$$

$$\tag{8}$$

We define the general QFT as distributing to the components of a quantum state:

$$\mathrm{QFT}_{2^n} : |\psi\rangle \mapsto \psi_0 \mathrm{QFT}_{2^n} |0\rangle + \psi_1 \mathrm{QFT}_{2^n} |1\rangle + \cdots + \psi_{2^n-1} \mathrm{QFT}_{2^n} |2^n - 1\rangle . \tag{9}$$

As an example, applying the QFT to a uniform superposition, we see that all but $|00\rangle$ of the terms cancel out,

$$\mathrm{QFT}_{2^2} \frac{1}{\sqrt{2^2}} \big(|00\rangle + |01\rangle + |10\rangle + |11\rangle \big) = |00\rangle \tag{10}$$

Connection with Fourier Analysis

Next we will show how the QFT can be seen as a transformation from the frequency domain to the time domain. Particularly in vector form the input and output states are related by the DFT^{-1}. Such connection is elucidated by explicitly writing the relationship between the entries of the vectors of the input and output quantum states of a QFT, which is what we do next.

Once again consider an n qubit quantum state $|\psi\rangle$ and let $|\phi\rangle$ be the result of the application of the QFT:

$$\begin{aligned} |\phi\rangle &= \mathrm{QFT}_{2^n} |\psi\rangle \\ &= \sum_{x=0}^{2^n-1} \psi_x \mathrm{QFT}_{2^n} |x\rangle \\ &= \sum_{x=0}^{2^n-1} \psi_x \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{\frac{2\pi i x y}{2^n}} |y\rangle \\ &= \sum_{x=0}^{2^n-1} \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} \psi_x e^{\frac{2\pi i x y}{2^n}} |y\rangle \\ &= \sum_{y=0}^{2^n-1} \underbrace{\left(\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} \psi_x e^{\frac{2\pi i x y}{2^n}} \right)}_{\phi_y} |y\rangle . \end{aligned}$$

Observing the above equation, note that the portion in parentheses is just a scalar, while $|y\rangle$ is a basis state. Thus each y value of the summation defines a unique component $\phi_0, \ldots, \phi_{2^n-1}$ of $|\phi\rangle$. Explicitly:

$$\phi_y = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} \psi_x e^{\frac{2\pi i x y}{2^n}} \quad \text{for } y = 1, \ldots, 2^n - 1. \tag{11}$$

Forgetting for a moment that $|\psi\rangle$ and $|\phi\rangle$ are quantum states, and treating them just as complex vectors, the essence of the QFT is revealed: $|\phi\rangle = \mathrm{DFT}^{-1} |\psi\rangle$. It follows from the fact that Eq (11) matches exactly the definition of DFT^{-1} in equation 6.1.2 (c) of [13]. This means that that $|\psi\rangle$ represent the coefficients of a trigonometric polynomial and $|\phi\rangle$ its time sampled representation. It must be noted that for $\mathrm{QFT}_{2^n} |\psi\rangle$ to be a valid quantum state, it has to be a normalized vector. The result follows from the fact that the DFT^{-1} is unitary [13] and that, as pure vectors, $\mathrm{QFT}_{2^n} |\psi\rangle = |\phi\rangle = \mathrm{DFT}^{-1} |\psi\rangle$. A unitary transformation preserves magnitude, and the $|\psi\rangle$ is normalized, therefore $|\phi\rangle$ is as well.

Quantum Operator Formulation

Unlike their classical counterparts, quantum computers are not programmable in the usual sense, what they do is apply operators to qubits. This means that in a quantum program, we start from a qubit state and we apply combinations of operators to "evolve" the quantum state over time. In this section we show how to implement a 2 qubit QFT on a real quantum computer. This is done by splitting the operations of the QFT into a series of unitary matrices, which are then represented by nodes called *gates* on a quantum circuit. Since we mainly intend to give the reader an intuitive understanding of the QFT, we restrict ourselves to a 2 qubits case. Also, due to a phenomenon called *quantum decoherence* [15], we are limited to a handful of working qubits, as using more would result in a high error in the output. A two qubit QFT is implemented with exactly two kinds of operators: the Hadamard operator H, from section 2, and the phase rotation operator R_θ. Both operators are unitary and have an equivalent matrix form. We will show operations that perform the QFT to state $|10\rangle$, but analogous steps apply to $|00\rangle$,$|01\rangle$ and $|11\rangle$. We will proceed sequentially in composing the QFT, this is to mimic the applications of operators as it would happen in a quantum computer. Finally we will see how such operators yeild a QFT. We begin explicitly the state of two qubits with the tensor product,

$$|10\rangle = |1\rangle \otimes |0\rangle \,.$$

We put the first qubit into a superposition by applying the Hadamard operator, and do nothing on the second,

$$\mathrm{H}\,|1\rangle \otimes |0\rangle \,.$$

Now, we have a tensor product of a qubit in an superposition state with another qubit $|0\rangle$. Next, we know that QFT performs phase rotations. To this end, we apply $\mathrm{R}_{\frac{\pi}{2}}$ operator to the second qubit,

$$\mathrm{R}_{\frac{\pi}{2}} = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{2}x_1} \end{bmatrix} . \tag{12}$$

This operator rotates the phase of the $|0\rangle$ state of a qubit by $\frac{\pi}{2}$,

$$\mathrm{H}\,|1\rangle \otimes \mathrm{R}_{\frac{\pi}{2}}\,|0\rangle \,. \tag{13}$$

And finally, we apply a Hadmard operator on the second qubit. This results in a QFT operator expressed in tensor algebra,

$$\mathrm{QFT}\,|10\rangle = (\mathrm{H} \otimes \mathrm{H}R_{\frac{\pi}{2}})\,|10\rangle = \mathrm{H}\,|1\rangle \otimes \mathrm{HR}_{\frac{\pi}{2}}\,|0\rangle \,. \tag{14}$$

Now we carry out the calculations to show that the above steps really perform a QFT on the desired state,

$$\begin{aligned} \mathrm{H}\,|1\rangle \otimes \mathrm{HR}_{\frac{\pi}{2}}\,|0\rangle &= \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \otimes \mathrm{H}\,|0\rangle \\ &= \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ &= \frac{1}{\sqrt{2}}\frac{1}{\sqrt{2}}(|0\rangle \otimes |0\rangle - |0\rangle \otimes |1\rangle + |1\rangle \otimes |0\rangle - |1\rangle \otimes |1\rangle) \\ &= \frac{1}{\sqrt{2^2}}\big(e^{0i}\,|00\rangle + e^{\pi i}\,|01\rangle + e^{0i}\,|10\rangle + e^{\pi i}\,|11\rangle\big) \\ &= \mathrm{QFT}_{2^2}\,|10\rangle \,. \end{aligned} \tag{15}$$

Therefore, we obtain a result identical to that we presented in Eq (6). Repeating the calculations for the other basis states, we can express QFT in compact operator form,

$$\mathrm{QFT}\,|x_1x_2\rangle = (\mathrm{H} \otimes \mathrm{H}R_{\frac{\pi}{2}})\,|x_1x_2\rangle = \mathrm{H}\,|x_1\rangle \otimes \mathrm{HR}_{\frac{\pi}{2}}\,|x_2\rangle \tag{16}$$

A consideration on the qubit and operator complexity of the QFT. Let n denote the number of components of a vector. To store this vector as a quantum state we only need $O(\log n)$ qubits, as shown by Eq (8). To implement the QFT $O((\log n)^2)$ gates are necessary [1]. Therefore, QFT is exponentially faster than a DFT^{-1} performed with the Fast Fourier Transform technique [13], which would have taken $O(n \log n)$ classical operations.

Circuit Implementation

Having this defined the QFT as a series of operators we can convert it into a quantum circuit. We show that to process a vector of size 4 we only need 3 gates which gives us rough idea of the speedup from $O(n \log n)$ to $O((\log n)^2)$. We construct a circuit and execute it remotely on the IBM's quantum computer using the cloud service IBM Quantum Experience [4]. The figure below shows a left to right sequence of gates which correspond to our above computation, applied to the two qubits (top two horizontal lines in the figure). The circuit execution is as follows. The system starts with top and bottom qubits in a classical state $\{10\}$. Next, the top qubit is put in the superposition by the Hadamard gate H. The gate H is followed by the phase rotation gate $\mathrm{R}_{\frac{\pi}{2}}$. Then the bottom qubit is put in the superposition by the second H gate. Finally, two qubits are measured. All these gates comprise a QFT on a state $|10\rangle$.

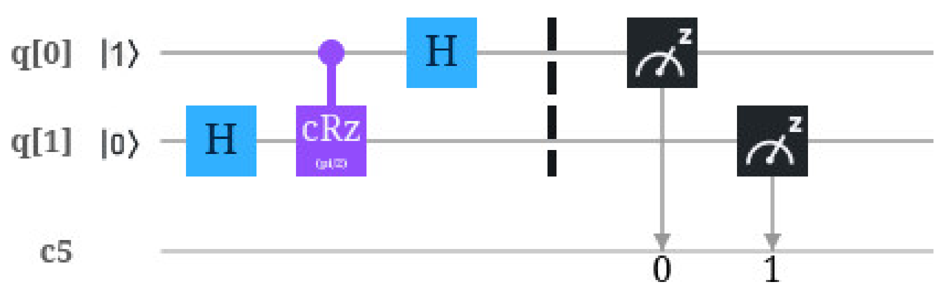

Figure 1: Our quantum circuit implementation of the QFT on the state $|10\rangle$. The circuit reads from left to right, as applications of operators to the qubits. "H" is the Hadamard operator. "cRz", with the "(pi/2)" option, represents the $R_{\frac{\pi}{2}}$ operator, with the protrusion indicating the dependency on first qubit. The black squares stand for measurement.

Now that we have the circuit ready, we execute it on IBM's quantum computer and obtain a new state which is in superposition with a phases determined by the QFT. When we take repeated measurements, the manifestation of the superposition of equal magnitudes of all states becomes evident in the histogram below, i.e. every basis state has an equal probability of being observed after execution.
We can not observe or access the phase of the state. In general, actual coefficients of states are not observable. So if one wants to know the complete information about some given state, the Phase Estimation Algorithm, which is closely related to the QFT, is needed. Our next section will fully describe and derive the Phase estimation Algorithm.

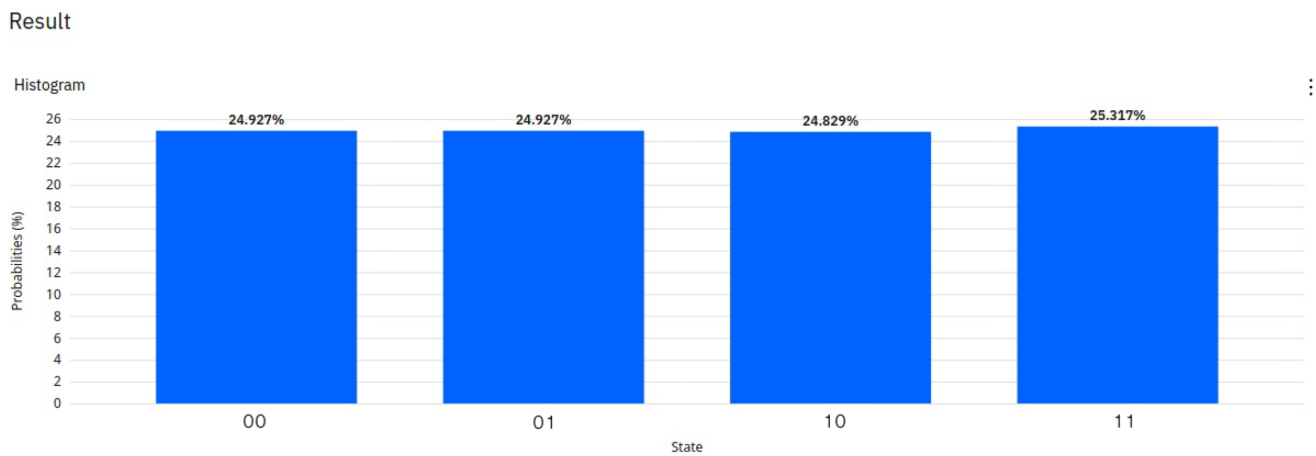

Figure 2: After applying the QFT on the state 10 we get the following bar chart. The bar plot above, shows the frequency each basis state is observed with, i.e. we see that the state 00 is observed in 24.9% of cases, states 01, 10 and 11 are observed in 24.9%, 24.8% and 24.3% respectively.

4 Quantum Phase Estimation

Having defined and demonstrated the application of QFT, we proceed to apply its inverse to an important problem in quantum computing, Quantum Phase Estimation. As we mentioned before, the state of the quantum system is not directly accessible. For example, in classical computing a programmer can access any variable stored in a system's memory. However, this is not the case in quantum computing. Having any quantum state at hand it is not possible to access it. In the preceding section we showed that we could obtain a rough estimate of the probability distribution of a state by repetitive measurements. Such an approach does not fully describe the system as it doesn't reveal the phase of the coefficients. Therefore, Phase Estimation is an important algorithm in QC because it can approximate unobservable quantities.

Problem Statement

The problem statement is as follows, given a quantum state

$$|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{2\pi i \omega y} |y\rangle \,, \tag{17}$$

we want to estimate the phase ω which is a real number on the interval $(0, 1)$. We estimate ω using Phase Estimation Algorithm

$$\mathcal{A} : |\psi\rangle \to |x_1 x_2 ... x_n\rangle \tag{18}$$

where $x_j = \{0, 1\}$ for $j = 1, ..., n$. The input of the algorithm $\mathcal{A}$ is a state vector $|\psi\rangle$ and the output is another state which encodes binary representation of estimated phase, i.e. $|x_1 x_2 ... x_n\rangle$ is equivalent to $\omega = 0.x_1 x_2 ... x_n$ in binary and $\omega = x_1/2 + x_2/4 + ... + x^n/2^n$ in decimal representation. From here we can immediately observe that more qubits allow for better estimation.In the following section we present analytical and numerical results with 2 and 3 qubits respectively.

Algorithm Derivation

In this section we derive the Phase Estimation algorithm. Most literature on the Phase Estimation presents general results for n qubits. Such presentation relies on additional theorems which may not be obvious in general setting. Since in this work we aim for an intuition, we formally derive the algorithm for $n = 2$ such that it becomes clear how to inductively generalize it to more qubits. As before we start with a state $|\psi\rangle$ defined in Eq (17) and rewrite it in a tensor product form which is much easier to work with.

$$\frac{1}{\sqrt{4}}\sum_{y=0}^{3} e^{2\pi i y\omega}|y\rangle = \frac{1}{2}\left[e^{0}|00\rangle + e^{2\pi i\omega}|01\rangle + e^{4\pi i\omega}|10\rangle + e^{6\pi i\omega}|11\rangle\right] \tag{19}$$

$$= \frac{1}{2}\left[|0\rangle\otimes\left(|0\rangle + e^{2\pi i\omega}|1\rangle\right) + e^{4\pi i\omega}|1\rangle\otimes\left(|0\rangle + e^{2\pi i\omega}|1\rangle\right)\right] \tag{20}$$

$$= \frac{|0\rangle + e^{4\pi i\omega}|1\rangle}{\sqrt{2}}\otimes\frac{|0\rangle + e^{2\pi i\omega}|1\rangle}{\sqrt{2}}. \tag{21}$$

Next, we want to further simplify the exponential coefficients of the left most factor of Eq (21). Recall that the phase is encoded in binary i.e. $\omega = 0.x_1x_2$. This is equivalent to $\omega = x_1/2 + x_2/4$ in the decimal system. Therefore, we get $2\omega = 2(x_1/2 + x_2/4) = x_1 + x_2/2$ which in binary form becomes $2\omega = x_1.x_2$. Hence, we can rewrite the exponential coefficients as follows

$$e^{4\pi i\omega} = e^{2\pi i 2\omega} = e^{2\pi i x_1.x_2} = e^{2\pi i(x_1 + 0.x_2)} = e^{2\pi x_1}e^{2\pi i 0.x_2} = e^{2\pi i 0.x_2}.$$

The tensor product Eq (21) becomes

$$S_1\otimes S_2 := \frac{|0\rangle + e^{2\pi i 0.x_2}|1\rangle}{\sqrt{2}}\otimes\frac{|0\rangle + e^{2\pi i 0.x_1x_2}|1\rangle}{\sqrt{2}}. \tag{22}$$

Because we split the input state ψ into two factors we can work on each factor separately. We start with S_1 and further simplify it

$$S_1 = \frac{|0\rangle + e^{2\pi i 0.x_2}|1\rangle}{\sqrt{2}} = \frac{|0\rangle + e^{2\pi i x_2/2}|1\rangle}{\sqrt{2}} = \frac{|0\rangle + e^{\pi i x_2}|1\rangle}{\sqrt{2}} = \frac{|0\rangle + (-1)^{x_2}|1\rangle}{\sqrt{2}} = H|x\rangle_2. \tag{23}$$

Referring back to Section 2 Eq (2) we can see that S_1 is simply $H|x_2\rangle$. Therefore, applying the Hadamard transformation one more time we get

$$H|x_2\rangle \xrightarrow{H} |x_2\rangle, \tag{24}$$

$$\frac{|0\rangle + (-1)^{x_2}|1\rangle}{\sqrt{2}} \xrightarrow{H} |x_2\rangle. \tag{25}$$

Therefore, we recover x_2 of ω. If $x_2 = 0$ we recover x_1 in similar fashion. However, if $x_2 = 1$ we need to do some extra work. Suppose $x_2 = 1$, then the second factor S_2 is

$$S_2 = \frac{|0\rangle + e^{2\pi i 0.x_1x_2}|1\rangle}{\sqrt{2}} = \frac{|0\rangle + e^{2\pi i x_1/2 + 1/4}|1\rangle}{\sqrt{2}}.$$

For clarity let us express the above in terms of vectors in $\mathbb{C}^2$,

$$S_2 = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\0\end{bmatrix} + \frac{e^{2\pi i(x_1/2+1/4)}}{2}\begin{bmatrix}0\\1\end{bmatrix}$$
$$= \frac{1}{\sqrt{2}}\begin{bmatrix}1\\0\end{bmatrix} + \frac{e^{\pi i x_1 + \pi i/2}}{\sqrt{2}}\begin{bmatrix}0\\1\end{bmatrix}.$$

If we annihilate the $\pi i/2$ term in the exponent of the second coefficient we get similar result as in Eq (23) but with x_1 instead of x_2. To this end consider the phase shift matrix

$$R_{-\frac{\pi}{2}} = \begin{bmatrix}1 & 0\\ 0 & e^{-x_2\pi i/2}\end{bmatrix} \quad \text{where } x_2 = 1. \tag{26}$$

Multiplying S_2 by $R_{-\frac{\pi}{2}}$ and rewriting the result back in Dirac notation we get

$$\frac{|0\rangle + (-1)^{x_1}|1\rangle}{\sqrt{2}}.$$

And again, we arrive to the same form as in Eq (23). Applying Hadamard transformation to the above recovers the value for x_1. We summarize the Phase Estimation procedure in the following algorithm

Algorithm 1: Phase Estimation

Result: $\omega = |x_1 x_2\rangle$;
$S_1, S_2 \leftarrow \text{factor}(|\psi\rangle)$;
$|x_2\rangle \leftarrow HS_1$;
if $|x_2\rangle == 1$ **then**
  $|x_1\rangle \leftarrow HRS_2$;
else
  $|x_1\rangle \leftarrow HS_2$;
end

Finally we express the algorithm as the inverse of QFT operator presented in Eq (16).

$$\text{QFT}^{-1}|\psi\rangle = (\text{H} \otimes \text{H}R_{-\frac{\pi}{2}})(S_1 \otimes S_2) = HS_1 \otimes HR_{-\frac{\pi}{2}}S_2 = |x_1 x_2\rangle. \tag{27}$$

We treat the result $|x_1 x_2\rangle$ as a binary number $\omega = .x_1 x_2$ or $\omega = x_1/2 + x_2/4$ in decimal.
The inverse property is verified below,

$$\text{QFT}\,\text{QFT}^{-1} = (\text{H} \otimes \text{H}R_{\frac{\pi}{2}})(\text{H} \otimes \text{H}R_{-\frac{\pi}{2}}) = I \otimes I = \text{QFT}^{-1}\,\text{QFT}.$$

Here are some interesting take away conclusions. QFT^{-1} reveals the phase of an arbitrary superposition state. Then it follows that given a classical basis state $|x_1 ... x_n\rangle$ (which can be treated as a binary number) QFT encodes its value as a phase in a new superposition state. In other words, one can think of QFT as an encoder of classical binary information into quantum information.

5 Conclusion

We first introduce some basic concepts from quantum computing, and then leverage them to present two quantum algorithms: the QFT and Quantum Phase Estimation. In this report we introduce the language of quantum computing based on linear algebra. We proceed by defining the QFT algorithm and present several concrete worked examples. We draw a mathematical connection between the QFT and DFT. The theoretical part of the report is supplemented with an actual implementation of the algorithm on the IBM's Quantum Computer. We demonstrate our constructed circuit and show that our particular implementation requires only 3 operations to perform the Quantum Fourier Transformation on a 4 dimensional vector. Finally, we present one of the main applications of the QFT, estimation of the Quantum Phase of an arbitrary state in a uniform superposition. Quantum Phase Estimation algorithm allows us to compute properties of quantum states otherwise inaccessible. We conclude the Phase Estimation section with an intuitive and yet important interpretation of the QFT as an operation that encodes classical information into quantum information and vise versa.

References

[1] Don Coppersmith. "An approximate Fourier transform useful in quantum factoring". In: *arXiv preprint quant-ph/0201067* (2002).

[2] Wim van Dam et al. "Optimal quantum circuits for general phase estimation". In: *Physical review letters* 98.9 (2007), p. 090501.

[3] The Editors of Encyclopaedia Britannica. https://www.britannica.com/science/wave-particle-duality. 2016.

[4] IBM Quantum Experience. https://quantum-computing.ibm.com/. 2020.

[5] Laszlo Gyongyosi and Sandor Imre. "An Improvement in Quantum Fourier Transform". In: *arXiv preprint arXiv:1207.4464* (2012).

[6] Sean Hallgren. "Polynomial-time quantum algorithms for Pell's equation and the principal ideal problem". In: *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*. 2002, pp. 653–658.

[7] Aram W Harrow, Avinatan Hassidim, and Seth Lloyd. "Quantum algorithm for linear systems of equations". In: *Physical review letters* 103.15 (2009), p. 150502.

[8] Sandor Imre and Ferenc Balazs. *Quantum computing and communications*. Wiley Online Library, 2005. Chap. 6: Quantum Fourier Transform and its Applications.

[9] Hidary J.D. *Quantum Computing: An Applied Approach*. Springer, 2019. Chap. 2. A Brief History of Quantum Computing.

[10] Hidary J.D. *Quantum Computing: An Applied Approach*. Springer, 2019. Chap. 1. Superposition, Entanglement and Reversibility.

[11] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. "Quantum principal component analysis". In: *Nature Physics* 10.9 (2014), pp. 631–633.

[12] Ashley Montanaro. "Quantum algorithms: an overview". In: *npj Quantum Information* 2.1 (2016), pp. 1–8.

[13] Marıa Cristina Pereyra and Lesley A Ward. *Harmonic analysis: from Fourier to wavelets*. Vol. 63. American Mathematical Soc., 2012. Chap. A tour of discrete Fourier and Haar analysis.

[14] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. "Quantum support vector machine for big data classification". In: *Physical review letters* 113.13 (2014), p. 130503.

[15] Maximilian Schlosshauer. "Quantum decoherence". In: *Physics Reports* 831 (2019), pp. 1–57.

[16] Peter W Shor. "Algorithms for quantum computation: discrete logarithms and factoring". In: *Proceedings 35th annual symposium on foundations of computer science*. Ieee. 1994, pp. 124–134.