Quantum Fourier Transform and Quantum Phase Estimation

P471: Quantum Computation and Quantum Information

Authors:

Mrinali Mohanty, 2011091
Varun Subudhi, 2011188
Hudha P M, 2011071

9 May 2024

Abstract

Abstract

The Quantum Fourier Transform (QFT) and Quantum Phase Estimation (QPE) are foundational algorithms in quantum computing, offering powerful capabilities for analyzing and manipulating quantum states. This report delves into the theory and practical implementation of these transformative techniques using Qiskit, a prominent open-source quantum computing framework. Starting with the preparation of Gaussian states, we discretize Gaussian functions and encode them into quantum states, applying the QFT to obtain their Fourier transforms. Overcoming challenges such as gate availability and measurement integration ensures smooth implementation. Additionally, we explore Quantum Phase Estimation, a crucial algorithm for discerning eigenvalues of unitary operators, demonstrating its practical application in quantum computing through Qiskit. Through a blend of theoretical insights and hands-on implementation, this report offers a comprehensive understanding of QFT and QPE, along with their implications in real-world quantum computing.

Keywords : Quantum Fourier Transform (QFT), Quantum Phase Estimation (QPE), Quantum computing, Qiskit, Gaussian state preparation, Fourier transform, Quantum algorithms.

Contents

1 Introduction

1.1 Fourier Transform

Fourier transform (FT) is an integral transform that takes a function as input and outputs another function that describes the extent to which various components of Fourier basis are present in the original function. The Fourier transform takes us from one basis to another, for example, from position to momentum or from time to frequency. Continuous Fourier transform is defined by:

$$g(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t) e^{i\omega t} dt \tag{1}$$

For analyzing a signal in time basis, the Fourier transform breaks down a function or signal into its sinusoidal parts with different frequencies. It represents a function f(t) in terms of its frequency spectrum g(ω), where ω is the angular frequency. This is done by using complex exponential functions $e^{i\omega t}$, oscillating at different frequencies. Integrating the signal with these complex exponentials over time gives the amplitude and phase of each frequency component present in the signal. Translation (i.e. delay) in the time domain is interpreted as complex phase shifts in the frequency domain. Functions concentrated in time spread out in frequency, and vice versa, a concept known as the uncertainty principle. For example, the Fourier transform of a Gaussian function is another Gaussian function. Let's check that in subsubsection 1.1.1. We are using the classical form of the Fourier transform, applicable to continuous, time-domain signals. It converts a function of time into a function of frequency.

1.1.1 Fourier Transform of a Gaussian function

Gaussian function has the following form:

$$f(t) = ae^{-(t-b)^2/2c^2} \tag{2}$$

where a, b, and c are arbitrary constants representing the height of the curve's peak, the peak's position, and the distribution's width (standard deviation), respectively.
From Equation 1, its Fourier transform is:

$$g(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t) e^{i\omega t} dt = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} ae^{-(t-b)^2/2c^2} e^{i\omega t} dt \tag{3}$$

$$\text{Let } \frac{t-b}{\sqrt{2}c} = y \implies dy = \frac{dx}{\sqrt{2}c}$$

$$g(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} ae^{-y^2} e^{i\omega(\sqrt{2}yc+b)} dy\sqrt{2}c$$

$$g(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} ae^{-(y^2 - i\omega\sqrt{2}yc)} e^{i\omega b} dy\sqrt{2}c$$

$$y^2 - \frac{i\omega yc}{\sqrt{2}} = y^2 - \frac{i\omega yc}{\sqrt{2}} - \frac{\omega^2 c^2}{2} + \frac{\omega^2 c^2}{2} = \left(y - \frac{i\omega c}{\sqrt{2}}\right)^2$$

$$g(\omega) = \frac{\sqrt{2}ca}{\sqrt{2\pi}} e^{i\omega b} e^{-\omega^2 c^2/2} \int_{-\infty}^{\infty} e^{-(y - i\omega yc/\sqrt{2})^2} dy$$

$$\text{Let } y - \frac{i\omega yc}{\sqrt{2}} = z \implies dy = dz$$

$$g(\omega) = \frac{ca}{\sqrt{\pi}} e^{i\omega b} e^{-\omega^2 c^2/2} \int_{-\infty}^{\infty} e^{-z^2} dz$$

$$g(\omega) = \frac{ca}{\sqrt{\pi}} e^{i\omega b} e^{-\omega^2 c^2/2} \sqrt{\pi}$$

$$g(\omega) = cae^{i\omega b} e^{-\omega^2 c^2/2}$$

Thus, the Fourier transform of a Gaussian function results in another Gaussian function. The peak position of the Fourier-transformed Gaussian will be at origin. An oscillatory phase factor $e^{i\omega b}$ encodes the displacement of the peak from the origin in position space. The pulse width (standard deviation) of the Fourier-transformed Gaussian is inversely proportional to the pulse width of the original Gaussian. The amplitude of the Fourier-transformed Gaussian is scaled by a factor compared to the original Gaussian. This scaling ensures that the area under the curve is preserved during the transformation.

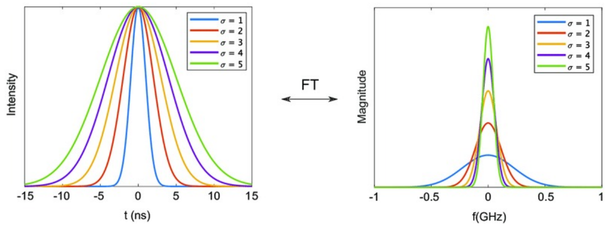

Figure 1: Fourier Transform of a Gaussian function [gau]

1.2 Discrete Fourier transform

Due to the use of digital computers, the continuum of values is replaced by a discrete set, and the integration is replaced by a summation. Discrete Fourier Transform (DFT) converts a finite sequence of equally-spaced samples of a function into another sequence of equally-spaced samples.
Consider a set of N time values:

$$t_k = \frac{kT}{N}, k = 0, 1.., N-1$$

The reciprocal space, ω space, can be represented by:

$$\omega_p = \frac{2\pi p}{T}, p = 0, 1.., N-1$$

So, the function of time in discrete space undergoes DFT to give the function of frequency as shown:

$$g(\omega_p) = \frac{1}{N}\sum_{0}^{N-1} f(t_k)e^{i\omega_p t_k} \tag{4}$$

And the inverse DFT is done by:

$$f(t_k) = \sum_{0}^{N-1} g(\omega_p)e^{-i\omega_p t_k} \tag{5}$$

$f(t_k)$ and $g(\omega_p)$ are discrete Fourier transforms of each other.
The DFT is widely used in digital signal processing for spectral analysis, filtering, and modulation tasks. It allows us to analyze the frequency components present in a discrete signal. [4]

1.3 Quantum Fourier Transform (QFT)

QFT is effectively a change of basis from the computational basis to the Fourier basis. For 1 qubit, the computational basis is $|0\rangle$ and $|1\rangle$, and the Fourier basis is $|+\rangle$ and $|-\rangle$. These states can be represented in a Bloch sphere, with $|0\rangle$ and $|1\rangle$ in the z-axis and $|+\rangle$ and $|-\rangle$ in the equator. So, when we do QFT, we go from top/bottom to being on the equator.

1.3.1 Basics to understand QFT

Qubits can be represented by states $|0\rangle$ and $|1\rangle$.

$$|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}; |1\rangle = \begin{pmatrix}0\\1\end{pmatrix}$$

The quantum states can be manipulated by quantum gates. Gates relevant to our topic are listed below:

- Hadamard gate: It converts $|0\rangle$ and $|1\rangle$ states to $|+\rangle$ and $|-\rangle$ states, which are superposition of $|0\rangle$ and $|1\rangle$ states.

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\1 & -1\end{pmatrix}$$

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |+\rangle \text{ and } H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |-\rangle$$

- X-gate: It rotates the state by 180^0 about the x-axis

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$X|0\rangle = |1\rangle \text{ and } X|1\rangle = |0\rangle$$

- $UROT_k$ gate: It rotates the state by an angle $2\pi i/2^k$ around the z-axis and is controlled by another set of qubits.

$$U = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{pmatrix}$$

$$UROT_k|0\rangle = |0\rangle \text{ and } UROT_k|1\rangle = e^{2\pi i/2^k}|1\rangle$$

The operation of quantum gates should preserve the norm.

1.3.2 Mathematical derivation

Let's consider n-qubits, which means there are $N = 2^n$ basis states. Similar to the DFT, the QFT acts on a quantum state in computational basis to take it to Fourier basis. This can be expressed as:

$$\left|\tilde{X}\right\rangle = QFT\,|X\rangle = \frac{1}{\sqrt{N}}\sum_{Y=0}^{N-1} e^{2\pi i \frac{XY}{N}}\,|Y\rangle \tag{6}$$

The tilde ($\sim$) are used to denote the states in Fourier basis.

If $|X\rangle = \sum_{j=0}^{N-1}\alpha_j\,|j\rangle$ and maps it to the quantum state $|Y\rangle = \sum_{k=0}^{N-1}\beta_k\,|k\rangle$. In the QFT, we do a DFT on the amplitudes of a quantum state:

$$\sum_j \alpha_j\,|j\rangle \to \sum_k \beta_k\,|k\rangle \text{ with } \beta_k = \frac{1}{\sqrt{N}}\sum_{j=0}^{N-1} e^{2\pi i \frac{jk}{N}}\alpha_j \tag{7}$$

This can also be expressed by mapping of basis:

$$|j\rangle \to \frac{1}{\sqrt{N}}\sum_{j=0}^{N-1} e^{2\pi i \frac{jk}{N}}\,|k\rangle \tag{8}$$

A unitary matrix can represent the above. The matrix is given as:

$$U_{QFT} = \frac{1}{\sqrt{N}}\sum_{j,k=0}^{N-1} e^{2\pi i \frac{jk}{N}}\,|k\rangle\langle j| \tag{9}$$

For a single qubit, H-gate does the QFT, and it transforms between the Z-basis states $|0\rangle$ and $|1\rangle$ to the Fourier basis states $|+\rangle$ and $|-\rangle$.

$$QFT\,|x\rangle = \frac{1}{\sqrt{2}}\sum_{y=0}^{2-1} e^{2\pi i xy/2}\,|y\rangle \tag{10}$$

$$QFT\,|0\rangle = \frac{1}{\sqrt{2}}[|0\rangle + |1\rangle] = |+\rangle$$

$$QFT\,|1\rangle = \frac{1}{\sqrt{2}}[|0\rangle + e^{i\pi}\,|1\rangle] = |-\rangle$$

For n-qubits, $|x\rangle = |x_1 ... x_n\rangle$ where x_1 is the most significant bit. It can be represented in binary as:

$$x = 2^{n-1}x_1 + 2^{n-2}x_2 + ... + 2^0 x_n = \sum_{k=1}^{n} x_k 2^{n-k}$$

$$QFT|x\rangle = \frac{1}{\sqrt{N}}\sum_{y=0}^{N-1} e^{2\pi ixy/N}|y\rangle \text{ where, } N = 2^n$$
$$= \frac{1}{\sqrt{N}}\sum_{y=0}^{N-1} e^{2\pi ix\sum_{k=1}^{n} y_k 2^{n-k}/2^n}|y\rangle$$
$$= \frac{1}{\sqrt{N}}\sum_{y=0}^{N-1} \Pi_{k=1}^{n} e^{2\pi iy_k x/2^k}|y_1...y_n\rangle$$
$$= \frac{1}{\sqrt{N}}\sum_{y_1=0}^{1} ... \sum_{y_n=0}^{1} \Pi_{k=1}^{n} e^{2\pi iy_k x/2^k}|y_1...y_n\rangle$$
$$= \frac{1}{\sqrt{N}}\Pi_{k=1}^{n}\sum_{y_1=0}^{1} e^{2\pi iy_1 x/2^1}|y_1\rangle \otimes ... \otimes \sum_{y_n=0}^{1} e^{2\pi iy_n x/2^n}|y_n\rangle$$

$$\implies QFT|x\rangle = \frac{1}{\sqrt{N}} \otimes_{k=1}^{n} (|0\rangle + e^{2\pi ix/2^k}|1\rangle)$$

$$QFT|x\rangle = \frac{1}{\sqrt{N}}(|0\rangle + e^{2\pi ix/2}|1\rangle) \otimes (|0\rangle + e^{2\pi ix/2^2}|1\rangle) \otimes ... \otimes (|0\rangle + e^{2\pi ix/2^{n-1}}|1\rangle) \otimes (|0\rangle + e^{2\pi ix/2^n}|1\rangle)$$

1.3.3 Making the circuit

From the above calculation, we can see that the phase is qubit-dependent, and we need to add more components with more "1"s. For this, Hadamard and $UROT_k$ gates are used. The circuit is given in Figure 2 States

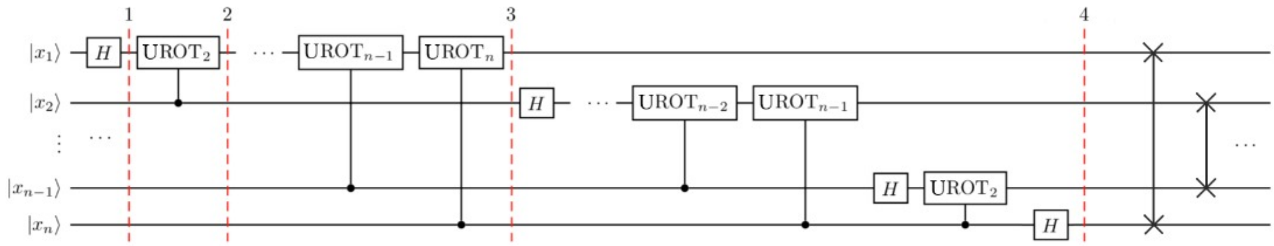

Figure 2: A circuit implementing the quantum Fourier transform. The states at each orange line is written in this section [1]

at different instances in the circuit:

- Input state: $|x_1...x_n\rangle$
- Step1: $\frac{1}{\sqrt{2}}[|0\rangle + |1\rangle] \otimes |x_2...x_n\rangle$
- Step2: $\frac{1}{\sqrt{2}}[|0\rangle + e^{2\pi ix_1/2}|1\rangle] \otimes |x_2...x_n\rangle$
- Step3: $\frac{1}{\sqrt{2}}[|0\rangle + e^{2\pi ix_n/2^n}...e^{2\pi ix_1/2}|1\rangle] \otimes |x_2...x_n\rangle$
- Step4:= $\frac{1}{\sqrt{N}}(|0\rangle + e^{2\pi ix/2^n}|1\rangle) \otimes (|0\rangle + e^{2\pi ix/2^{n-1}}|1\rangle) \otimes ... \otimes (|0\rangle + e^{2\pi ix/2^2}|1\rangle) \otimes (|0\rangle + e^{2\pi ix/2^1}|1\rangle)$

This circuit implements QFT but in reverse order of qubits at the output. So, swap gate is applied at the end of the circuit to return the output in the same order as input. [Benenti et al.]

As the QFT circuit size grows, more time is consumed by progressively smaller rotations. Thus, ignoring rotations below a certain threshold still yields satisfactory results. This is known as approximate QFT, and it is significant in physical setups, where minimizing operations can significantly mitigate decoherence and potential gate errors.

1.4 Quantum Phase Estimation (QPE)

Unitary matrices have eigenvalues in the form of $e^{i\theta}$, which apply phase to the eigenvectors. Quantum Phase Estimation (QPE) is a quantum algorithm designed to estimate the phases acquired by quantum states under these operators' actions. Here, we're using QFT to determine the phase.
Say a state is $(|0\rangle + |1\rangle)/\sqrt{2}$. The probability of measuring $|0\rangle$ and $|1\rangle$ is 50-50. When there is a global phase, say θ, i.e. the state is $e^{i\theta}(|0\rangle + |1\rangle)/\sqrt{2}$, the probability of measuring $|0\rangle$ and $|1\rangle$ is still 50-50. So, this algorithm is important for determining this θ.
This is important to understand the evolution of the system. Since the Hamiltonian is unitary, determining the phase helps us in successful simulation of these systems.

1.4.1 Mathematical understanding

This algorithm takes a unitary operator U and an eigenvector $|\psi\rangle$ as input, where U acts on $|\psi\rangle$ as $U|\psi\rangle = e^{2\pi i\theta}|\psi\rangle$, with θ being the eigenvalue to be estimated. $|\psi\rangle$ resides in one set of qubit registers, while an additional set of n-qubits forms the counting register to store the value $2^n\theta$.

$$|\psi_0\rangle = |0\rangle^{\otimes n}|\psi\rangle$$

Next, a n-bit Hadamard gate operation $H^{\otimes n}$ is applied to the counting register.

$$|\psi_1\rangle = \frac{1}{2^{n/2}}(|0\rangle + |1\rangle)^{\otimes n}|\psi\rangle$$

A quantum circuit is devised to prepare an ancillary qubit in a superposition of states, representing various estimates of the phase θ. This is achieved by applying a series of controlled-unitary operations, i.e., introducing a phase in the system only if the control qubit is $|1\rangle$.

$$\begin{aligned}
U^{2^j}|\psi\rangle &= U^{2^j-1}U|\psi\rangle \\
&= U^{2^j-1}e^{2\pi i\theta}|\psi\rangle \\
&= \dots \\
&= e^{2\pi 2^j\theta}|\psi\rangle
\end{aligned}$$

By applying all n controlled operations CU^{2^j} with $0 \le j \le n-1$, and utilizing the relation $|0\rangle \otimes |\psi\rangle + |1\rangle \otimes e^{2\pi i\theta}|\psi\rangle = (|0\rangle + e^{2\pi i\theta}|1\rangle) \otimes |\psi\rangle$,

$$\begin{aligned}
|\psi_2\rangle &= \frac{1}{2^{n/2}}\left(|0\rangle + e^{2\pi i\theta 2^{n-1}}|1\rangle\right) \otimes \dots \otimes \left(|0\rangle + e^{2\pi i\theta 2^1}|1\rangle\right) \otimes \left(|0\rangle + e^{2\pi i\theta 2^0}|1\rangle\right) \\
&= \frac{1}{2^{n/2}}\sum_{k=0}^{2^n-1}|k\rangle \otimes |\psi\rangle
\end{aligned}$$

where k represents the integer representation of n-bit binary numbers. The phase θ is encoded into the state of the ancillary qubit through a phenomenon known as phase kickback. This occurs when the eigenvalue θ "kicks back" to the ancillary qubit during the controlled-unitary operations.
QPE applies an inverse Quantum Fourier Transform to the ancillary qubits. This transforms the phase-encoded state into a superposition of computational basis states, where the probability amplitudes correspond to estimates of the phase θ.

$$QFT|x\rangle = \frac{1}{\sqrt{N}}\left(|0\rangle + e^{2\pi ix/2}|1\rangle\right) \otimes \left(|0\rangle + e^{2\pi ix/2^2}|1\rangle\right) \otimes \dots \otimes \left(|0\rangle + e^{2\pi ix/2^{n-1}}|1\rangle\right) \otimes \left(|0\rangle + e^{2\pi ix/2^n}|1\rangle\right)$$

Substituting x with $2^n\theta$ in the above expression yields the result obtained in step 1. Consequently, to retrieve the state $|2^n\theta\rangle$, apply an inverse Fourier transform on the auxiliary register, resulting in:

$$|\psi_3\rangle = \frac{1}{2^{n/2}}\sum_{k=0}^{2^n-1}e^{2\pi i\theta k}|k\rangle \otimes |\psi\rangle \xrightarrow{QFT_n^{-1}} \frac{1}{2^n}\sum_{x,k=0}^{2^n-1}e^{-2\pi ik(x-2^n\theta)/N}|x\rangle \otimes |\psi\rangle$$

Finally, the ancillary qubits representing the phase information are measured, yielding an estimate of the phase θ with high probability. The expression peaks near $x = 2^n\theta$. In cases where $2^n\theta$ is an integer, measuring in the computational basis provides the phase in the auxiliary register with high probability:

$$|\psi_4\rangle = |2^n\theta\rangle \otimes |\psi\rangle$$

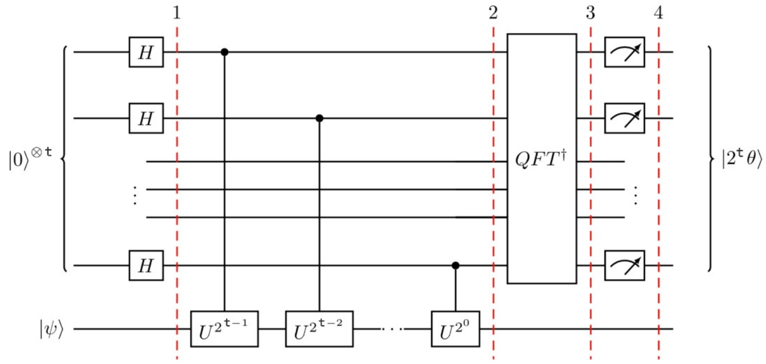

Figure 3: A general quantum circuit for quantum phase estimation with the top register contains t 'counting' qubits, and the bottom contains qubits in the state $|\psi\rangle$ [2]

The measurement outcomes offer an estimate of the phase, typically in the form of a binary fraction.[Benenti et al.]

Quantum Phase Estimation (QPE) is pivotal in quantum computing, finding applications in various fields. In Shor's algorithm, QPE efficiently determines the period of a function, aiding in the factorization of large composite numbers—an arduous task for classical computers. Moreover, in quantum chemistry, QPE estimates molecular system energies, contributing to research in quantum chemistry and materials science. Furthermore, QPE facilitates quantum simulation by estimating Hamiltonian operator eigenvalues, enabling the exploration of complex quantum phenomena and simulating quantum system time evolution.

2 Numerical Procedures and Computational Setup

2.1 Qiskit Implementation

2.1.1 Qiskit

Qiskit, developed by IBM, is an open-source quantum computing software framework. It offers tools for designing quantum circuits, algorithms, and simulators, along with access to real quantum hardware via the IBM Quantum Experience platform. Utilizing a Python-based high-level programming language, users can specify quantum gates, qubit operations, and measurements to construct quantum algorithms. Qiskit's built-in simulators allow users to simulate quantum circuit behavior, aiding in testing and debugging before deploying algorithms on real quantum hardware. Through the IBM Quantum Experience platform, users gain access to various quantum processors with different qubit counts and error rates, facilitating quantum experimentation and research.

2.2 Execution of Algorithms

(Circuits with their Executions and Outputs can be accessed in the GitHub links.) Link to the Repository: https://github.com/MrinaliMohanty/QIQC-Project-

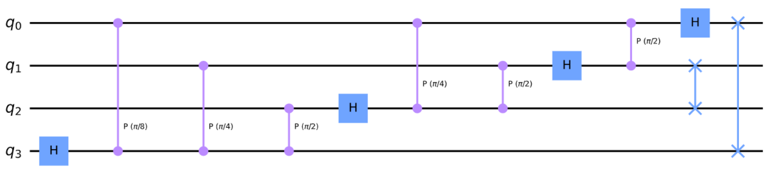

Figure 4: A general quantum circuit for quantum Fourier transform using 4 qubits. [1]

2.2.1 Quantum Phase Estimation (QPE)

https://github.com/varun-subudhi/QIQC-Project-P471/blob/main/QPE_Implementation.ipynb

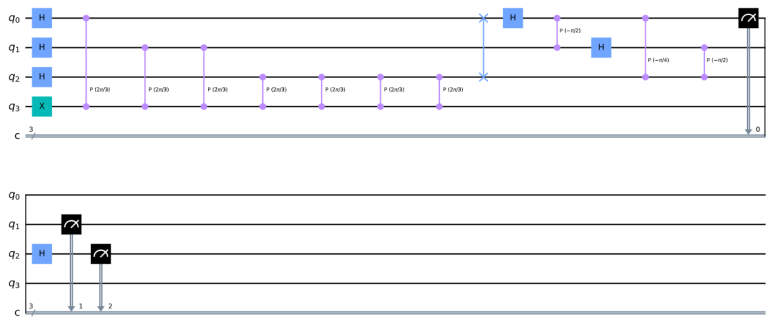

Figure 5: A general quantum circuit for quantum phase estimation using 4 qubits. [2]

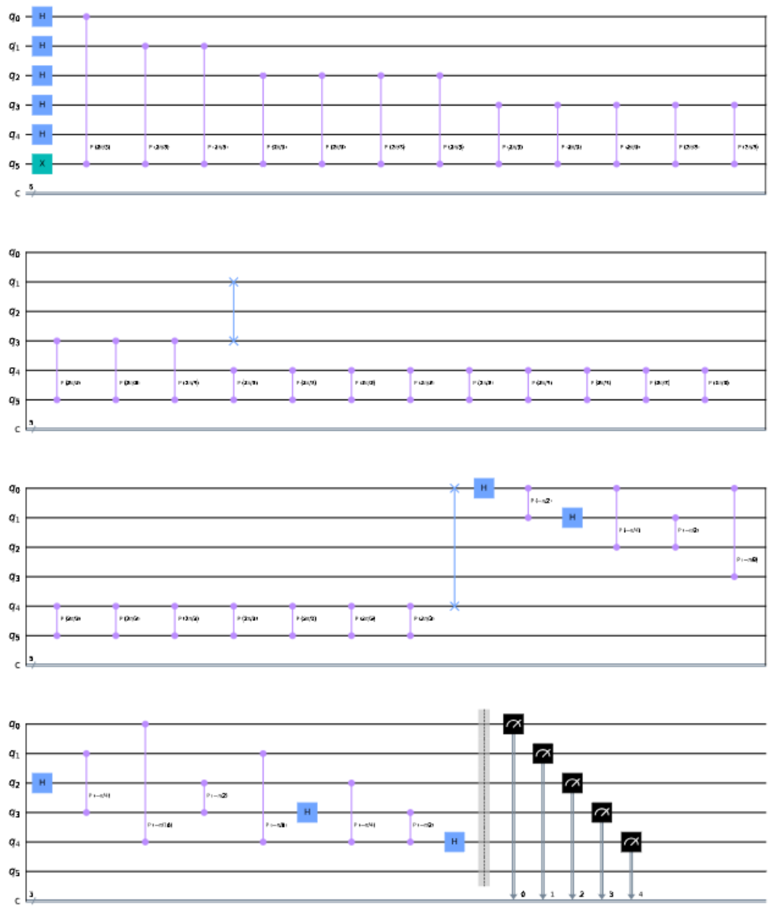

Figure 6: A general quantum circuit for quantum phase estimation using 6 qubits. [2]

3 Results

Our project successfully executed the quantum Fourier transform (QFT) and quantum phase estimation (QPE) algorithms, constructing quantum circuits using Qiskit. Some tests were conducted to verify circuit accuracy.
In the QFT circuit, a Gaussian function served as input, producing a Gaussian function as output. Even with changes in the mean position, the peak of the Fourier-transformed Gaussian remained at zero. Increasing the input Gaussian's standard deviation led to a corresponding increase in the Fourier-transformed Gaussian's amplitude, consistent with analytical calculations. For a smoother graph, the number of qubits in the circuit has to be increased.
The global phase of the state was determined by applying an input state to the QPE circuit. The probability distribution provided information about the global phase. Increasing the number of qubits resulted in enhanced precision in phase estimation.

4 Future Scope

- Qiskit's simulators provide a cost-effective means to simulate quantum circuits, facilitating algorithm development and testing without the need for quantum hardware. These simulators allow for scalable exploration of circuits with varying qubits and gates, ensuring both flexibility and speed for efficient quantum computing experimentation. While simulations offer scalability and speed advantages, accessing real quantum hardware through IBM Q on the IBM Quantum Experience platform is invaluable for validating algorithms under real-world conditions, providing essential feedback for refinement.
- Conducting Quantum Fourier Transform (QFT) implementations on diverse functions and verifying them classically enables benchmarking of calculation speeds, offering insights into the efficiency of quantum algorithms across different scenarios. Comparative analysis of QFT performance across various functions aids in evaluating the effectiveness and applicability of quantum computing in addressing specific problems, guiding algorithm optimization, and enhancement efforts.
- Exploring quantum equivalents of transforms such as the Laplace transform, and Chirp-Z transform presents an opportunity to assess their computational speed and effectiveness in quantum computing. Benchmarking these quantum transformations against their classical counterparts provides insights into their computational efficiency and suitability across different domains. This comparative assessment helps understand the strengths and limitations of quantum transforms, facilitating the optimization and adaptation of quantum algorithms for specific tasks.
- While the quantum phase estimation algorithm may seem constrained by the need to know how to perform controlled operations on a quantum computer, it is feasible to create circuits for which this knowledge is unknown. Learning theta in such circuits can yield valuable insights, serving as a fundamental subroutine for renowned algorithms like Shor's algorithm, which is pivotal for identifying the period of a function relevant to integer factorization.

Acknowledgements

I extend my sincere gratitude to Dr. Anamitra Mukherjee, our professor at NISER, Bhubaneswar, for his invaluable guidance and contributions that significantly shaped this project. I also wish to express my appreciation to my project teammates, Varun Subudhi and Hudha P M, as well as to my friends, for their insightful discussions and feedback, which greatly enhanced the quality of this report.

References

[1] Qft. https://www.youtube.com/watch?v=pq2jkfJlLmY.

[2] Qpe. https://www.youtube.com/watch?v=5kcoaanYyZw&t=1234s.

[gau] Sample gaussian ft. https://www.researchgate.net/figure/Left-Several-Gaussian-functions-with-differen
fig3_339376506.

[4] Arfken, G. (1985). *Mathematical Methods for Physicists*. Academic Press, Inc., San Diego, third edition.

[Benenti et al.] Benenti, G., Casati, G., Rossini, D., and Strini, G. *Principles of Quantum Computation and Information*. 2 edition.