A Practical Guide to Model Problems in Quantum Linear Algebra: Theory and Experiment

Alexander Weiss*† and Paul Bruzzi*‡

Project advisor: Fabian M. Faulstich§

Abstract. We present a comprehensive and pedagogical study of two core quantum algorithms in quantum linear algebra: the Hadamard test (HT) and quantum phase estimation (QPE). Building on the lecture notes on Quantum Algorithms for Scientific Computing [1] by Lin, we detail their mathematical formulations, circuit implementations, and convergence behavior. Both algorithms are benchmarked on simulators and IBM's 127-qubit Eagle processor (`ibm_rensselaer`) using a simple eigenvalue problem. While QPE performs well in simulation, its hardware performance degrades beyond $t = 3$ ancilla qubits due to circuit depth and two-qubit gate error. In contrast, HT shows stable sampling-based convergence and greater resilience to hardware noise. These results provide a realistic baseline for implementing phase estimation on noisy intermediate-scale quantum (NISQ) era devices.

1. Introduction.

Quantum linear algebra lies at the heart of many quantum algorithms that promise computational advantages over classical methods [2, 3, 4]. As linear algebra underpins a wide range of scientific computing tasks—from solving systems of equations to eigenvalue problems and signal processing—quantum linear algebra is expected to play a foundational role in quantum-enhanced numerical simulation, machine learning, and optimization (see, e.g., [5, 6, 7]).

While the theoretical development of quantum algorithms has seen rapid progress, their practical realization is limited by the current state of quantum hardware. Today's devices are constrained by short coherence times, gate error, and limited qubit connectivity, which significantly restrict the depth and complexity of executable circuits [8]. In addition, quantum hardware is inherently noisy due to unavoidable disturbances in its internal quantum system. As a result, many of the most promising quantum algorithms, while asymptotically efficient, are not yet viable for deployment on existing quantum processors. Bridging this gap between theoretical algorithm design and practical implementation is essential to realizing quantum utility in the near term.

This manuscript pursues two primary objectives: First, we aim to provide a pedagogical introduction to quantum linear algebra, accessible at the advanced undergraduate level. We focus on two fundamental algorithms, namely, the Hadamard test (HT) and quantum phase estimation (QPE). These serve as the conceptual and technical foundations for many advanced quantum algorithms, including those for Hamiltonian simulation, quantum singular value transformation, and solving linear systems of equations. Second, we provide a realistic comparison between the theoretical predictions and the performance of these algorithms on actual quantum hardware.

Specifically, we benchmark both HT and QPE using (i) Qiskit's AerSimulator [9]—a hypo-

*These authors contributed equally to this work.
†Department of Physics, Rensselaer Polytechnic Institute, Troy, NY (alexxweiss34@gmail.com)
‡Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY (paulbruzzi@gmail.com)
§Department of Mathematics, Rensselaer Polytechnic Institute, Troy, NY

thetical fault-tolerant device—and (ii) experiments on an IBM's 127-qubit Eagle-class quantum processor (`ibm_rensselaer`) [10]. This comparison exposes the strengths and limitations of both methods in the current noisy intermediate-scale quantum (NISQ) era and highlights the critical impact of circuit depth, gate error, and state preparation on algorithmic performance.

This manuscript was inspired by and closely follows the structure of Lin's lecture notes on Quantum Algorithms for Scientific Computing [1]. We supplement his exposition with additional clarifications and visual aids to support undergraduate students from STEM backgrounds who are new to quantum computing. In addition, we provide detailed Jupyter notebook implementations of all major concepts and circuits, which are available on our project GitHub repository [11].

The manuscript is organized as follows. Section 2 introduces the foundational elements of quantum computation, including qubits, single-qubit gates, and quantum phase. Section 3 presents the Hadamard test as a simple method for estimating expectation values. Section 4 discusses the discrete Fourier transform and its quantum implementation via the quantum Fourier transform. Section 5 builds on this to construct the quantum phase estimation algorithm. In section 6, we compare the performance of HT and QPE in terms of accuracy and runtime on both simulated and real hardware. Conclusions are summarized in section 7.

We gratefully acknowledge the use of IBM Quantum services for this work. The views expressed here are those of the authors and do not represent the official views of IBM or the IBM Quantum team.

2. Computation on a Quantum Computer.

2.1. Qubits. Quantum bits, or *qubits*, are the fundamental units of quantum information. Analogous to classical bits, which can be in one of two states (0 or 1), qubits can exist in two computational basis states denoted as $|0\rangle$ and $|1\rangle$ in Dirac notation, often referred to as *spin up* and *spin down* states respectively as in [3, 4].

Definition 2.1. *The computational basis states of a single qubit are defined as*

$$|0\rangle := \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \textit{and} \quad |1\rangle := \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{2.1}$$

A general qubit state is a normalized linear combination (superposition) of these basis states:

$$|\psi\rangle = c_1|0\rangle + c_2|1\rangle, \qquad \textit{where} \qquad c_1, c_2 \in \mathbb{C},\ |c_1|^2 + |c_2|^2 = 1. \tag{2.2}$$

Remark 2.2. Unlike classical bits, qubits may exist in superpositions. For instance,

$$|+\rangle := \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \text{and} \quad |-\rangle := \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ -1 \end{bmatrix}, \tag{2.3}$$

are both valid qubit states. See subsection SM1.1 for visualizations.

For systems involving multiple qubits, the full state is represented by a tensor product of individual qubit states. This is commonly abbreviated using the following notation:

$$|\psi_1\rangle \otimes |\psi_2\rangle = |\psi_1\rangle|\psi_2\rangle = |\psi_1\psi_2\rangle. \tag{2.4}$$

Example 2.3. A system with two qubits in states $|0\rangle$ and $|1\rangle$ is represented as

$$|0\rangle \otimes |1\rangle = \begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}1\begin{bmatrix}0\\1\end{bmatrix}\\0\begin{bmatrix}0\\1\end{bmatrix}\end{bmatrix} = \begin{bmatrix}0\\1\\0\\0\end{bmatrix} = |01\rangle. \tag{2.5}$$

2.2. Measurement and Phase. Measuring a qubit collapses its superposition into one of the computational basis states, with probabilities determined by the squared modulus of its inner product (overlap) with the measured state.

Definition 2.4. *The probability of measuring a state $|\phi\rangle$ and finding it in state $|\psi\rangle$ is*

$$\mathbb{P}(X \in \{|\psi\rangle\}) = |\langle\psi|\phi\rangle|^2, \tag{2.6}$$

where X is a random variable representing the measurement of $|\phi\rangle$.

Example 2.5. The probability of measuring $|+\rangle$ and obtaining $|0\rangle$ is

$$|\langle 0|+\rangle|^2 = \left|\langle 0|\left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right)\right|^2 = \left|\frac{1}{\sqrt{2}}(\langle 0|0\rangle + \langle 0|1\rangle)\right|^2 = \left|\frac{1}{\sqrt{2}}\right|^2 = \frac{1}{2}, \tag{2.7}$$

where we used the orthonormality of the computational basis, i.e., $\langle 0|0\rangle = 1$ and $\langle 0|1\rangle = 0$. In vector notation this reads

$$|\langle 0|+\rangle|^2 = \left|\begin{bmatrix}1\\0\end{bmatrix}^\dagger \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}\right|^2 = \left|\begin{bmatrix}1 & 0\end{bmatrix}\frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}\right|^2 = \left|\frac{1}{\sqrt{2}}\right|^2 = \frac{1}{2}, \tag{2.8}$$

where "†" denotes the adjoint. This means that a qubit initially in the $|+\rangle$ state will collapse to the $|0\rangle$ state upon measurement with a probability of $1/2$.

Remark 2.6. Since the modulus square is invariant under rotation, there exist multiple states that collapse to the same state with the same probability when they are measured in the computational basis, i.e, when multiplying the expansion coefficients of a state by $e^{i2\pi\varphi}$ with $\varphi \in [0, 1)$. The quantity $e^{i2\pi\varphi}$ is called *phase* and $2\pi\varphi$ is called *phase angle* of the state. Here, φ acts as the fraction of a full rotation of 2π around the complex unit circle. There are two types of phase, *global phase* which is the phase between different state vectors and *relative phase* which is the phase between bit strings in the same state vector. The idea of relative and global phase is expanded upon in Example 2.9.

2.3. Single-Qubit Gates. Quantum gates manipulate the state of qubits. The most common single-qubit gates include the Pauli-X (bit flip), Pauli-Z (phase flip), R_z (phase rotation), and the Hadamard gate (superposition).

Definition 2.7. *The matrix representations of the Pauli-X, Pauli-Z, R_z, and Hadamard gate in the computational basis are:*

$$X := \begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}, \quad Z := \begin{bmatrix}1 & 0\\0 & -1\end{bmatrix}, \quad R_z(\theta) = \begin{bmatrix}e^{-i\theta/2} & 0\\0 & e^{i\theta/2}\end{bmatrix} \quad \text{and} \quad H := \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}. \tag{2.9}$$

The following table summarizes their action on the computational basis states:

Input	X	Z	$R_z(\theta)$	H
$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$e^{-i\theta/2}\|0\rangle$	$\frac{1}{\sqrt{2}}(\|0\rangle + \|1\rangle)$
$\|1\rangle$	$\|0\rangle$	$-\|1\rangle$	$e^{i\theta/2}\|1\rangle$	$\frac{1}{\sqrt{2}}(\|0\rangle - \|1\rangle)$

Table 1
The result of the X-gate, Z-gate, $R_z(\theta)$-gate, and H-gate acting on the $|0\rangle$ and $|1\rangle$ states.

Remark 2.8. The effect of applying a phase gate (e.g., Z-gate, $R_z(\theta)$-gate) does not change the outcome of a measurement in the computational basis, but changes the quantum state by a complex phase. These phase factors can be interpreted as eigenvalues:

$$U|\psi\rangle = \lambda|\psi\rangle, \quad \text{with} \quad \lambda = \langle\psi|U|\psi\rangle. \tag{2.10}$$

Since quantum states are normalized, $|\lambda| = 1$, implying $\lambda = e^{i2\pi\varphi}$ (c.f. Remark 2.6).

Example 2.9. Expanding further on Remark 2.6, the action of $R_z(\theta)$ on $|+\rangle$ yields:

$$R_z(\theta)|+\rangle = \frac{1}{\sqrt{2}}\left(e^{-i\theta/2}|0\rangle + e^{i\theta/2}|1\rangle\right) = \frac{e^{-i\theta/2}}{\sqrt{2}}\left(|0\rangle + e^{i\theta}|1\rangle\right), \tag{2.11}$$

revealing a global phase of $e^{-i\theta/2}$ and a relative phase of $e^{i\theta}$ between the basis components. See section SM2 for more information about single-qubit gates.

2.4. Circuit Diagrams. A circuit diagram (e.g. Figure 1) is a graphical representation that describes the workflow of a quantum algorithm. Quantum circuits are read from left to right as operations or gates are applied over time on lines representing the qubits. In general, one line in a circuit diagram corresponds to a single qubit; however, if a line has a slash through it, e.g. the third line in Figure 1, it represents multiple qubits. Applying a gate to a qubit is represented as a box on the qubit's line. Some gate operations depend on one or more control qubit states. These controls are denoted by a point drawn over the controlling qubit(s) with a line connected to the gate it is controlling; see $|\pi_2\rangle$ in Figure 1. To denote the swapping of two qubit states, a ($\times$) is drawn over the swapping qubits with a line connecting them; see $|\pi_3\rangle$ in Figure 1. Applying a gate to multiple qubits is shown with a box over all the qubits such as at $|\pi_4\rangle$ in Figure 1. Measurement is depicted with a "meter-like" symbol in a box on the qubit being measured, as shown in the first line at the end of the circuit. Finally, a double line denotes classical information, such as after a measurement has been performed.

3. The Hadamard Test. Although quantum measurements discard global phase information (see Remark 2.6), many quantum algorithms critically rely on its access. The Hadamard Test (HT) is a simple yet powerful circuit primitive to estimate the expectation value $\langle\psi|U|\psi\rangle$ of a unitary operator U with respect to a quantum state $|\psi\rangle$. It is important to note that while our analysis of the HT utilizes the exact target state $|\psi\rangle$, experimental implementations rely on an approximate ansatz. Consequently, the HT yields expectation value estimates with respect to this approximation, emphasizing the importance of an ansatz with sufficiently high

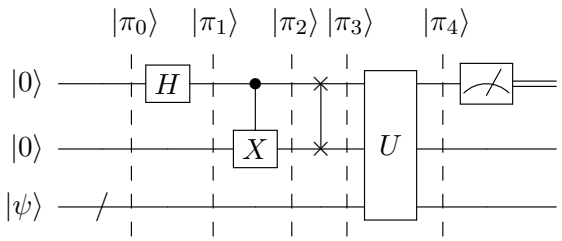


Figure 1. *An example quantum circuit diagram depicting a multi-qubit slash on the third line in the state $|\psi\rangle$, a Hadamard (H) gate on the first qubit (line), a controlled X-gate controlled on the first qubit targeting the second, a swap operation between the first and second qubits, an arbitrary multi-qubit U gate, and a measurement operator on the first qubit.*

overlap with the target state $|\psi\rangle$ (see, e.g., [12, 13, 14]). Though elementary in structure, the HT forms the conceptual backbone of numerous state-of-the-art quantum algorithms for linear algebra, including those for matrix functions, Hamiltonian simulation, and singular value transformation. Because the expectation value is generally complex, its real and imaginary parts must be estimated separately using two circuit variants: the *real Hadamard test* and the *imaginary Hadamard test* as in [15, 16].

3.1. The HT Circuit.

Definition 3.1. *Let $U \in \mathbb{C}^{n\times n}$ be unitary and define the phase gate $S = \sqrt{Z}$. The real and imaginary Hadamard tests (HT) are defined as the circuits depicted below in Figure 2.*

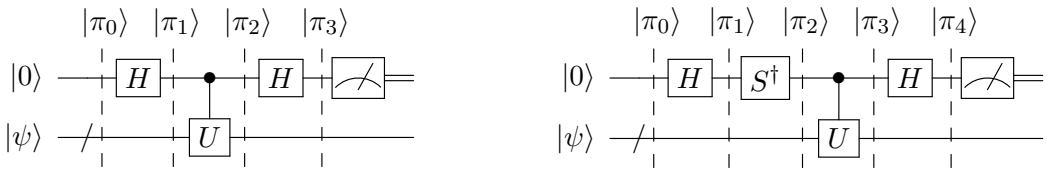


Figure 2. *(Left) The real Hadamard test circuit and (Right) the imaginary Hadamard test circuit. H is the Hadamard gate, S is the phase gate, U is a unitary gate. The topmost qubit starting in the state $|0\rangle$ is the control qubit, and the lower qubit(s) starting in the state $|\psi\rangle$ are the computation bits representing the eigenstate.*

The measurement of the control qubit in the real HT is represented by a random variable X_{Re} which is permitted to be either $|0\rangle$ or $|1\rangle$. Instead of referencing this random variable directly, we use the shorthand notation:

$$\mathcal{P}_{\text{Re}|0\rangle} := \mathbb{P}(X_{\text{Re}} \in \{|0\rangle\}) \quad \text{and} \quad \mathcal{P}_{\text{Re}|1\rangle} := \mathbb{P}(X_{\text{Re}} \in \{|1\rangle\}). \tag{3.1}$$

Similarly, a random variable X_{Im} being $|0\rangle$ or $|1\rangle$ describes the measurement of the control qubit in the imaginary HT, where we use the shorthand notation:

$$\mathcal{P}_{\text{Im}|0\rangle} := \mathbb{P}(X_{\text{Im}} \in \{|0\rangle\}) \quad \text{and} \quad \mathcal{P}_{\text{Im}|1\rangle} := \mathbb{P}(X_{\text{Im}} \in \{|1\rangle\}). \tag{3.2}$$

Proposition 3.2. *The measurement probabilities for the real Hadamard test are*

$$\mathcal{P}_{\mathrm{Re}|0\rangle} = \frac{1}{2}(1 + Re\langle\psi|U|\psi\rangle) \quad \textit{and} \quad \mathcal{P}_{\mathrm{Re}|1\rangle} = \frac{1}{2}(1 - Re\langle\psi|U|\psi\rangle). \tag{3.3}$$

Proof. Consider the real HT (left panel in Figure 2 in Definition 3.1) with initial state

$$|\pi_0\rangle = |0\rangle \otimes |\psi\rangle = |0\rangle|\psi\rangle. \tag{3.4}$$

A Hadamard gate is then applied to the control qubit, which yields the $|+\rangle$ state, i.e.,

$$|\pi_1\rangle = H|0\rangle I|\psi\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)|\psi\rangle = |+\rangle|\psi\rangle. \tag{3.5}$$

Next, a controlled unitary gate applies the U operation to $|\psi\rangle$ if the control qubit is $|1\rangle$, i.e.,

$$|\pi_2\rangle = \left(|0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes U\right)|\pi_1\rangle = \frac{1}{\sqrt{2}}|0\rangle I|\psi\rangle + \frac{1}{\sqrt{2}}|1\rangle U|\psi\rangle. \tag{3.6}$$

The last operation is a Hadamard gate, yielding the final state

$$|\pi_3\rangle = |0\rangle\frac{I+U}{2}|\psi\rangle + |1\rangle\frac{I-U}{2}|\psi\rangle, \tag{3.7}$$

where the probability of measuring the control qubit in $|0\rangle$ is

$$\mathcal{P}_{\mathrm{Re}|0\rangle} = |\langle 0|\pi_3\rangle|^2 = \left|\frac{I+U}{2}|\psi\rangle\right|^2 = \frac{2 + \langle\psi|(U+U^\dagger)|\psi\rangle}{4} = \frac{1}{2}(1 + \mathrm{Re}\langle\psi|U|\psi\rangle). \tag{3.8}$$

Similarly, the probability of measuring the control qubit in $|1\rangle$ is

$$\mathcal{P}_{\mathrm{Re}|1\rangle} = |\langle 1|\pi_3\rangle|^2 = \left|\frac{I-U}{2}|\psi\rangle\right|^2 = \frac{2 - \langle\psi|(U+U^\dagger)|\psi\rangle}{4} = \frac{1}{2}(1 - \mathrm{Re}\langle\psi|U|\psi\rangle). \tag{3.9}$$

■

Having established the measurement probabilities for the real HT, we can characterize the real part of the expectation value.

Corollary 3.3. *The real part of the expectation value is given by*

$$Re\langle\psi|U|\psi\rangle = \mathcal{P}_{\mathrm{Re}|0\rangle} - \mathcal{P}_{\mathrm{Re}|1\rangle} = 2\mathcal{P}_{\mathrm{Re}|0\rangle} - 1 = 1 - 2\mathcal{P}_{\mathrm{Re}|1\rangle}. \tag{3.10}$$

Proof. Note that Proposition 3.2 yields

$$2\mathcal{P}_{\mathrm{Re}|0\rangle} - 1 = (1 + \mathrm{Re}\langle\psi|U|\psi\rangle) - 1 = \mathrm{Re}\langle\psi|U|\psi\rangle, \tag{3.11}$$

and

$$\mathrm{Re}\langle\psi|U|\psi\rangle = 1 - (1 - \mathrm{Re}\langle\psi|U|\psi\rangle) = 1 - 2\mathcal{P}_{\mathrm{Re}|1\rangle}. \tag{3.12}$$

Combining equations (3.11) and (3.12) yields,

$$\mathrm{Re}\langle\psi|U|\psi\rangle = \frac{1}{2}(1 + \mathrm{Re}\langle\psi|U|\psi\rangle) - \frac{1}{2}(1 - \mathrm{Re}\langle\psi|U|\psi\rangle) = \mathcal{P}_{\mathrm{Re}|0\rangle} - \mathcal{P}_{\mathrm{Re}|1\rangle}. \tag{3.13}$$

■

For a more detailed step-by-step proof, we refer the reader to subsection SM3.2.

The analysis for the imaginary HT follows analogously. We here report the measurement probabilities together with the resulting expectation value, and refer the reader to subsection SM3.3 for the detailed proofs.

Proposition 3.4. *The measurement probabilities for the imaginary Hadamard test are*

$$\mathcal{P}_{\mathrm{Im}|0\rangle} = \frac{1}{2}(1 + Im\langle\psi|U|\psi\rangle) \quad \textit{and} \quad \mathcal{P}_{\mathrm{Im}|1\rangle} = \frac{1}{2}(1 - Im\langle\psi|U|\psi\rangle). \tag{3.14}$$

Corollary 3.5. *The imaginary part of the expectation value is given by*

$$Im\langle\psi|U|\psi\rangle = \mathcal{P}_{\mathrm{Im}|0\rangle} - \mathcal{P}_{\mathrm{Im}|1\rangle} = 2\mathcal{P}_{\mathrm{Im}|0\rangle} - 1 = 1 - 2\mathcal{P}_{\mathrm{Im}|1\rangle}. \tag{3.15}$$

3.2. Estimating Eigenphases. Returning to the task of phase estimation, recall that the eigenvalues of U are of the form

$$\lambda = e^{i\theta} = \cos\theta + i\sin\theta, \tag{3.16}$$

and thus $\mathrm{Re}\langle\psi|U|\psi\rangle = \cos\theta$. So the real probabilities can be written as

$$\mathcal{P}_{\mathrm{Re}|0\rangle} = \frac{1}{2}(1 + \cos\theta) \quad \text{and} \quad \mathcal{P}_{\mathrm{Re}|1\rangle} = \frac{1}{2}(1 - \cos\theta), \tag{3.17}$$

hence,

$$\theta = \arccos(2\mathcal{P}_{\mathrm{Re}|0\rangle} - 1) = \arccos(1 - 2\mathcal{P}_{\mathrm{Re}|1\rangle}). \tag{3.18}$$

Similarly, $\mathrm{Im}\langle\psi|U|\psi\rangle = \sin\theta$. Thus, θ can also be estimated by

$$\theta = \arcsin(2\mathcal{P}_{\mathrm{Im}|0\rangle} - 1) = \arcsin(1 - 2\mathcal{P}_{\mathrm{Im}|1\rangle}). \tag{3.19}$$

Finally, θ can be estimated using both the real and imaginary parts of the expectation value. Each part can be thought of as a vector on the complex unit circle with the real part on the x-axis and the imaginary part on the y-axis. Taking the arcsin of the imaginary part or the arccos of the real part is effectively calculating the vector of the entire complex expectation value. This means that the arctan of the imaginary part over the real part also gives θ. Therefore, the angle θ can be estimated with the following three methods:

$$\theta = \arccos(\mathrm{Re}\langle\psi|U|\psi\rangle) = \arcsin(\mathrm{Im}\langle\psi|U|\psi\rangle) = \arctan\left(\frac{\mathrm{Im}\langle\psi|U|\psi\rangle}{\mathrm{Re}\langle\psi|U|\psi\rangle}\right). \tag{3.20}$$

Remark 3.6. Note that using both the real and imaginary parts of the expectation value to estimate θ with arctan increases the error of the estimation, since it uses data from two random variables rather than just one.

3.3. HT Phase Estimation Example. To demonstrate the HT in practice, consider the unitary matrix

$$U = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{bmatrix}, \quad \text{with} \quad \theta = \frac{1}{2} + \frac{1}{2^4} = 0.5625. \tag{3.21}$$

We will use this simple matrix as a running example throughout this article. Our numerical tests are performed using Qiskit 1.3.1 alongside AerSimulator [9] for the simulated quantum data. We estimate θ by applying both the real and imaginary HTs and measuring 100 shots, see Figure 3.

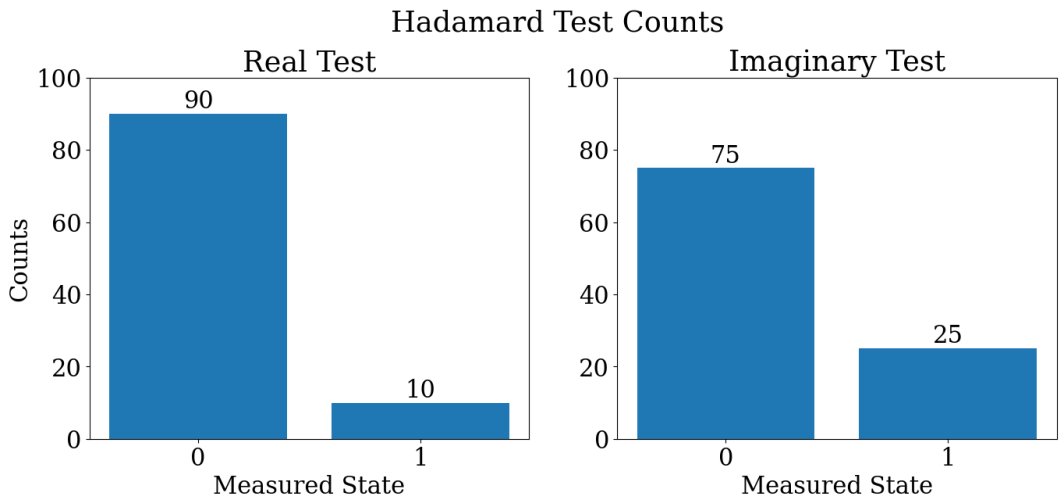


Figure 3. *Counts for the real (left) and imaginary (right) Hadamard tests for* 100 *shots with a phase angle* $\theta = 0.5625$.

Theoretical expectation values (c.f. (3.17)) follow from:

$$\mathcal{P}_{\mathrm{Re}|0\rangle} = \frac{1}{2}(1 + \cos(0.5625)) \approx 0.9230 \quad \text{and} \quad \mathcal{P}_{\mathrm{Re}|1\rangle} = \frac{1}{2}(1 - \cos(0.5625)) \approx 0.0770, \tag{3.22}$$

whereas the measured values from a simulation with 100 shots yield

$$\mathcal{P}_{\mathrm{Re}|0\rangle} = 0.90 \quad \text{and} \quad \mathcal{P}_{\mathrm{Re}|1\rangle} = 0.10. \tag{3.23}$$

Similarly, for the imaginary circuit, the computed probabilities are

$$\mathcal{P}_{\mathrm{Im}|0\rangle} = \frac{1}{2}(1 + \sin(0.5625)) \approx 0.7667 \quad \text{and} \quad \mathcal{P}_{\mathrm{Im}|1\rangle} = \frac{1}{2}(1 - \sin(0.5625)) \approx 0.2333, \tag{3.24}$$

whereas the measured values from a simulation with 100 shots yield

$$\mathcal{P}_{\mathrm{Im}|0\rangle} = 0.75 \quad \text{and} \quad \mathcal{P}_{\mathrm{Im}|1\rangle} = 0.25. \tag{3.25}$$

This illustrates the discrepancy between the theoretical and experimental values using 100 shots for both HTs. As expected, both probabilities always sum to 1 (c.f. subsection SM3.4).

Provided these probabilities, the real and imaginary expectation values can be found using (3.10) and (3.15), respectively. This yields

$$\mathrm{Re}\langle\psi|U|\psi\rangle = 0.80 \quad \text{and} \quad \mathrm{Im}\langle\psi|U|\psi\rangle = 0.50, \tag{3.26}$$

whereas the theoretical expectation values obtained from (3.22) and (3.24) are

$$\mathrm{Re}\langle\psi|U|\psi\rangle = 0.8460 \quad \text{and} \quad \mathrm{Im}\langle\psi|U|\psi\rangle = 0.5334. \tag{3.27}$$

Thus, the experimental and theoretical expectation values are approximately what we expect. Finally, either or both of these values can be used to estimate the phase angle θ using any equation in (3.20).

Remark 3.7. Since trigonometric functions are cyclic, inverse trigonometric functions have multiple possible values of θ that they can return. The standard for these types of functions is to return the value closest to zero, which is not necessarily the true phase. Therefore, the outputs of both arccos and arcsin are used to determine which quadrant the phase is in. The function arccos ranges from 0 to π, and arcsin from $-\pi/2$ to $\pi/2$. Arccos is used to determine whether the angle is in the positive x or negative x part of the graph, and arcsin is used to determine if it is in the positive y or negative y part of the graph. Using the angle returned from arccos, if arcsin gives a negative value, then the angle must be found in the negative y part of the graph. To do this, the estimated angle can be set to $2\pi - \theta$. Using the angle returned from arcsin, if arccos gives a value greater than $\pi/2$, then the angle must be determined in the negative x part of the graph. To do this, the estimated angle is set to $\pi - \theta$. For a graphical explanation we refer the reader to subsection SM3.5.

We emphasize that the above experiment simulated 100 shots, leading to an estimated phase angle $\theta_{est} = 0.6435$ (c.f. $\theta = 0.5625$) when using cos to determine the angle and sin to determine the sign. Note that the accuracy of the estimation depends directly on the number of shots. In the following section, we will connect the convergence behavior to Monte-Carlo type estimators, providing a rate of convergence for the HTs.

3.4. Error Analysis. In quantum computing, repeated executions of a circuit—known as *shots*—are used to estimate the probability of a quantum system collapsing into a given state. This is inherently a statistical process: each measurement samples from an underlying probability distribution that is not directly accessible. To infer this distribution, we employ tools from statistical estimation, most notably the empirical distribution function (EDF), which approximates the true distribution based on a finite number of observations. This forms the foundation for quantifying the accuracy and convergence of algorithms like the HTs.

Each single-qubit measurement of the control qubit yields a Bernoulli random variable $X \in \{0, 1\}$, with success probability $p = P(X = 0)$. For N_s independent shots, the EDF is defined as

$$\hat{F}(x) = \frac{1}{N_s} \sum_{i=1}^{N_s} \mathbb{1}_{\{X_i \leq x\}}, \tag{3.28}$$

where $X_i \sim X$ are i.i.d. and are corresponding to each measurement result. Here, $\mathbb{1}_{\{X_i \leq x\}}$ is the indicator function which returns 1 when $X_i \leq x$ and 0 otherwise.

It is often necessary to estimate the probability of a specific outcome. This is a special case of the EDF where it simplifies to the empirical probability estimator which focuses on the proportion of trials yielding a specific outcome. For example, the probability of measuring 0 on a quantum computer with N_s shots is estimated by

$$\hat{p}_0 = \frac{1}{N_s} \sum_{i=1}^{N_s} \mathbb{1}_{\{0\}}(X_i). \tag{3.29}$$

This is also called a Monte Carlo estimator which uses repeated sampling to predict the probability of a variety of outcomes. Recall that for the HTs, only a single qubit is measured, and thus the X_i's are Bernoulli distributed. Tying single-qubit measurement to common probability theory, the weak law of large numbers (see, e.g., [17, Section 10.2]) yields that $\hat{p}_0$ in (3.29) converges to $p_0 = \mathbb{E}(X)$ in probability.

Proposition 3.8. *The average of N_s single-qubit measurements converges in probability. Moreover, the convergence rate of the empirical estimator $\hat{p}_0$ can be calculated from the variance*

$$\mathbb{V}(\hat{p}_0) = \frac{p_0(1-p_0)}{N_s}, \tag{3.30}$$

which yields the standard deviation

$$\sigma = \sqrt{\frac{p_0(1-p_0)}{N_s}}. \tag{3.31}$$

This translates to taking N_s single-qubit measurements as follows.

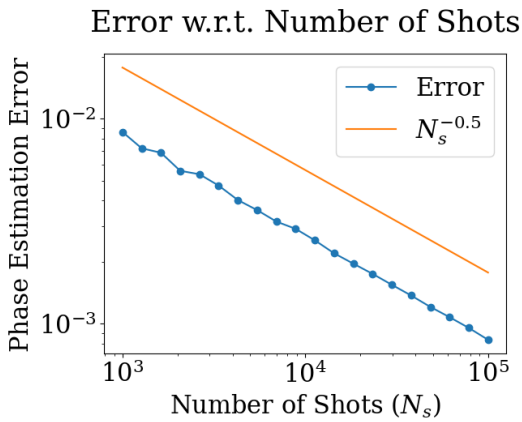
Proposition 3.9. *Let $\epsilon > 0$. The number of shots required to recover p_0 with precision ϵ via $\hat{p}_0$ scales as*

$$\epsilon \geq \sigma = \sqrt{\frac{p_0(1-p_0)}{N_s}} \Rightarrow \epsilon^2 \geq \frac{p_0(1-p_0)}{N_s} \Rightarrow N_s \geq \frac{p_0(1-p_0)}{\epsilon^2} \in \Omega\left(\frac{1}{\epsilon^2}\right). \tag{3.32}$$

The variance of the control qubit's output is dependent on p_0 and thus σ in (3.31). Plotting σ against N_s in Figure 4 for $10^3 - 10^5$ shots reveals that the error approaches zero as the number of shots increases when experimentally tested. Moreover, the estimated angle approaches $\theta = 0.5625$ as the number of shots increases, see the table in Figure 4.

4. Quantum Fourier Transform.

Fourier analysis plays a central role in scientific computing, offering a way to transform problems from the time or spatial domain to the frequency domain, where many tasks—such as filtering, compression, or solving differential equations—become more tractable [18]. In quantum computing, a related transformation known as the quantum Fourier transform (QFT) serves as a key subroutine in algorithms including Shor's algorithm, quantum phase estimation, and quantum signal processing (see, e.g., [15, 19, 20]). This section introduces both the classical Discrete Fourier Transform (DFT) and its quantum counterpart, the QFT, and outlines their mathematical foundations and circuit implementations.



Shot Count	Estim. Angle θ	Error
100	0.4027	0.1598
500	0.4864	0.0761
1000	0.5394	0.0231
5000	0.5830	0.0205
10000	0.5698	0.0073
50000	0.5569	0.0056
100000	0.5620	0.0005

Figure 4. *(Left) A log-log scaled plot of the standard deviation σ of the Hadamard test against the number of shots N_s converging like $N_s^{-1/2}$. (Right) A table showing the estimated angle θ converging to the expected value $\theta = 0.5625$ and its error approaching 0 as the number of shots increases.*

4.1. Discrete Fourier Transform. We begin by recalling the classical discrete Fourier transform, which maps a complex-valued vector in the time domain into the frequency domain.

Definition 4.1. *The (forward) discrete Fourier transform (DFT) is defined as the map,*

$$\mathcal{F} : \mathbb{C}^N \to \mathbb{C}^N \ ; \ x = \begin{bmatrix} x_0 \\ \vdots \\ x_{N-1} \end{bmatrix} \mapsto \tilde{x} = \begin{bmatrix} \tilde{x}_0 \\ \vdots \\ \tilde{x}_{N-1} \end{bmatrix}, \tag{4.1}$$

where

$$\tilde{x}_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{2\pi i \frac{jk}{N}}. \tag{4.2}$$

Example 4.2. Consider the continuous signal

$$x(t) = \sin(2\pi t) + \sin(2\pi(11.2)t) + \sin(2\pi(18.4)t), \tag{4.3}$$

sampled such that $x_j = x(t_j)$ with $t_j = j/100, j \in \mathbb{N}_0$, i.e., 100 samples per second. Taking 1 second of equally spaced samples yields the plot in the left panel of Figure 5. Applying the DFT in (4.2) where x_j are the samples, reveals the frequency amplitudes $\tilde{x}_k$ in the right panel of Figure 5.

Conversely, given a set of frequencies, the inverse discrete Fourier transform reconstructs the corresponding time-domain signal from its frequency-domain representation.

Definition 4.3. *The inverse discrete Fourier transform (IDFT) is defined as the map,*

$$\mathcal{F}^{-1} : \mathbb{C}^N \to \mathbb{C}^N \ ; \ \tilde{x} = \begin{bmatrix} \tilde{x}_0 \\ \vdots \\ \tilde{x}_{N-1} \end{bmatrix} \mapsto x = \begin{bmatrix} x_0 \\ \vdots \\ x_{N-1} \end{bmatrix}, \tag{4.4}$$

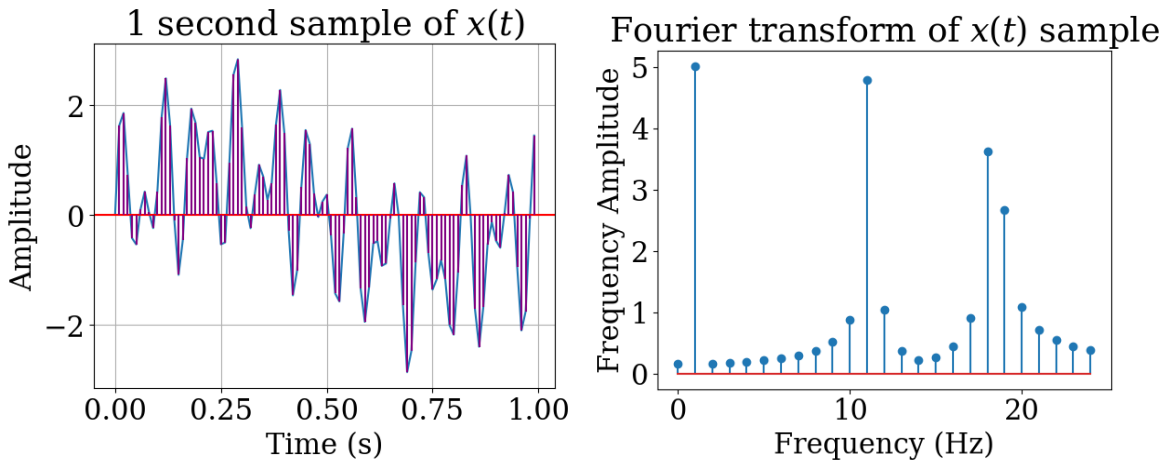


Figure 5. *(Left) A sample of $x(t)$ in (4.3), and (Right) its Fourier transform showing pronounced peaks at 1 Hz, 11.2 Hz and 18.4 Hz.*

where

$$x_j = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \tilde{x}_k e^{-2\pi i \frac{jk}{N}}. \tag{4.5}$$

4.2. QFT Algorithm. The quantum Fourier transform performs an operation similar to the DFT, but acts on the amplitudes of a quantum state. It is defined over computational basis states, which represent integers in binary format.

Definition 4.4. *Let $j \in \mathbb{N}$ and $n \in \mathbb{N}$ be the number of qubits. The (forward) quantum Fourier transform (QFT) is defined as the map*

$$|j\rangle \mapsto \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i \frac{jk}{2^n}} |k\rangle. \tag{4.6}$$

Here we express j and k as binary numbers to represent the computational basis state vectors. See subsection SM1.2 for more details.

Remark 4.5. The QFT can be expressed using the unitary matrix

$$\mathrm{QFT}_n = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} \sum_{k=0}^{2^n-1} e^{2\pi i \frac{jk}{2^n}} |k\rangle\langle j| = \frac{1}{\sqrt{2^n}} \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & \dots & \omega^{2^n-1} \\ 1 & \omega^2 & \omega^4 & \dots & \omega^{2(2^n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{2^n-1} & \omega^{2(2^n-1)} & \dots & \omega^{(2^n-1)^2} \end{bmatrix}, \tag{4.7}$$

where $\omega = e^{2\pi i \frac{1}{2^n}}$.

Example 4.6. Taking $n = 1$ in (4.7), the matrix form of QFT_1 is

$$\mathrm{QFT}_1 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & e^{\frac{2\pi i}{2}} \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & \cos(\pi) + i\sin(\pi) \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = H. \tag{4.8}$$

Therefore, the QFT for a single qubit is simply the Hadamard gate, i.e., this gate transforms the computational basis states $|0\rangle$ and $|1\rangle$ into the Fourier basis states $|+\rangle$ and $|-\rangle$, respectively.

Proposition 4.7. *QFT_n acting on a quantum state $|j\rangle$ yields the following state vector in the computational basis:*

$$\mathrm{QFT}_n|j\rangle = \frac{1}{\sqrt{2^n}} \bigotimes_{l=1}^{n} \left(|0\rangle + e^{2\pi i \frac{j}{2^l}}|1\rangle\right). \tag{4.9}$$

Proof. The map in Definition 4.4 can be rewritten using fractional binary notation where k and j are binary numbers, i.e.,

$$k = k_1 k_2 \dots k_n.0\ , \qquad j = j_1 j_2 \dots j_n.0\ . \tag{4.10}$$

Therefore, QFT_n in (4.6) acting on a state $|j\rangle$ can be written as

$$\mathrm{QFT}_n|j\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i \frac{jk}{2^n}}|k\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} \prod_{l=1}^{n} e^{2\pi i k_l \frac{j}{2^l}}|k\rangle, \tag{4.11}$$

where

$$\frac{jk}{2^n} = k_1\frac{j}{2^1} + k_2\frac{j}{2^2} + \dots + k_n\frac{j}{2^n} = \sum_{l=1}^{n} k_l \frac{j}{2^l}. \tag{4.12}$$

Since k is a binary number, each k_l can only be 0 or 1. Thus the sum over k becomes the direct product of the sum of each individual k_l from 0 to 1, i.e.,

$$\mathrm{QFT}_n|j\rangle = \frac{1}{\sqrt{2^n}} \sum_{k_1=0}^{1} e^{2\pi i k_1 \frac{j}{2^1}}|k_1\rangle \otimes \sum_{k_2=0}^{1} e^{2\pi i k_2 \frac{j}{2^2}}|k_2\rangle \otimes \dots \otimes \sum_{k_n=0}^{1} e^{2\pi i k_n \frac{j}{2^n}}|k_n\rangle. \tag{4.13}$$

This can be rewritten as multiple tensor products

$$\mathrm{QFT}_n|j\rangle = \frac{1}{\sqrt{2^n}} \bigotimes_{l=1}^{n} \left(|0\rangle + e^{2\pi i \frac{j}{2^l}}|1\rangle\right). \tag{4.14}$$

■

4.3. QFT Circuit Implementation. The QFT algorithm has an efficient implementation as a quantum circuit consisting of Hadamard gates and controlled phase rotations.

Proposition 4.8. *Let $R_n = R_z\left(\frac{\pi}{2^{n-1}}\right)$. The QFT circuit is depicted in Figure 6.*

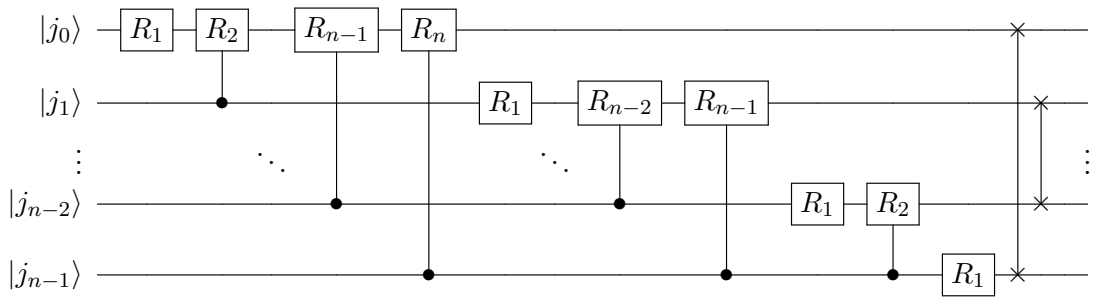


Figure 6. *The QFT circuit where* $R_n = R_z\left(\frac{\pi}{2^{n-1}}\right)$ *(Definition 2.7). Note that* $R_1 = H$.

Proof. If the circuit in Figure 6 performs a Fourier transform, then the final state of the circuit is the expression in (4.14). Therefore, a circuit without the swap gates—which we will call the swap-less QFT circuit ($\overleftarrow{\text{QFT}}$)—will swap the qubit order of the output. This implies that the final state of the swap-less QFT circuit would be

$$\overleftarrow{\text{QFT}}_n|j\rangle = \frac{1}{\sqrt{2^n}} \bigotimes_{l=n}^{1} \left(|0\rangle + e^{2\pi i \frac{j}{2^l}}|1\rangle\right). \tag{4.15}$$

Using the swap-less QFT circuit and (4.15), we construct an inductive proof verifying the circuit in Figure 6 does indeed perform a Fourier transform. Consider the swap-less QFT circuit with a single qubit. Here, the circuit becomes a single R_1 gate which is equivalent to a Hadamard gate. According to (4.15), the initial state of the qubit transforms to,

$$\overleftarrow{\text{QFT}}_1|j\rangle = \frac{1}{\sqrt{2^1}} \bigotimes_{l=1}^{1} \left(|0\rangle + e^{2\pi i \frac{j}{2^l}}|1\rangle\right) = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.j_1}|1\rangle\right) = \begin{cases} |+\rangle, \text{ if } j = 0, \\ |-\rangle, \text{ if } j = 1. \end{cases} \tag{4.16}$$

Since this is exactly the effect of a Hadamard gate, when $n = 1$, the swap-less QFT circuit correctly performs the map in (4.15), showing the induction base.

Since $\overleftarrow{\text{QFT}}_n$ represents the swap-less QFT circuit for n qubits, it then follows from (4.15) that $\overleftarrow{\text{QFT}}_{n+1}$ can be written as

$$\overleftarrow{\text{QFT}}_{n+1}|j\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.j_1 \ldots j_{n+1}}|1\rangle\right) \otimes \overleftarrow{\text{QFT}}_n|j_2 \ldots j_{n+1}\rangle. \tag{4.17}$$

From the inductive hypothesis, the $\overleftarrow{\text{QFT}}_n|j_2 \ldots j_{n+1}\rangle$ part of $\overleftarrow{\text{QFT}}_{n+1}$ is correctly performing the map in (4.15). Thus, if the first bit is being transformed correctly, the entire mapping is correct.

The phase on the $|1\rangle$ state in (4.17) can be rewritten as a combination of rotations specific to each binary digit of j, i.e.,

$$e^{2\pi i 0.j_1 \ldots j_{n+1}} = e^{2\pi i \frac{j_1}{2^1}} e^{2\pi i \frac{j_2}{2^2}} \cdots e^{2\pi i \frac{j_{n+1}}{2^{n+1}}} = e^{j_1(\pi i)} e^{j_2\left(\frac{\pi i}{2}\right)} \cdots e^{j_{n+1}\left(\frac{\pi i}{2^n}\right)}. \tag{4.18}$$

Since the phase rotation can be rewritten as a product of rotations that are dependent only on a single qubit, each rotation can be directly converted to a single controlled rotation gate. The rotation for each of these gates is defined as

$$r_n = \frac{\pi}{2^{n-1}}, \tag{4.19}$$

where n represents the qubit's number/label. Upon inspecting $\overleftarrow{\text{QFT}}_{n+1}$, the gates defined here and in Proposition 4.8 are equivalent. This means that if the swap-less QFT circuit with n qubits performs the map $\overleftarrow{\text{QFT}}_n$, the swap-less QFT circuit with $n+1$ qubits performs the map $\overleftarrow{\text{QFT}}_{n+1}$. Therefore, by induction the swap-less QFT circuit performs the map $\overleftarrow{\text{QFT}}_n$ for all $n \geq 1$. Since the swap-less QFT performs the mapping $\overleftarrow{\text{QFT}}_n$, the circuit in Figure 6 must perform the QFT mapping in Definition 4.4. ∎

We refer the interested reader to section SM4 for a more detailed explanation on this circuit's construction.

Definition 4.9. *Let $j \in \mathbb{N}$. The inverse quantum Fourier transform (IQFT) is defined as the map*

$$|k\rangle \mapsto \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} e^{-2\pi i \frac{kj}{2^n}} |j\rangle. \tag{4.20}$$

Proposition 4.10. *Let $R_n^\dagger = R_z\left(\frac{-\pi}{2^{n-1}}\right)$. The IQFT circuit is given in Figure 7.*

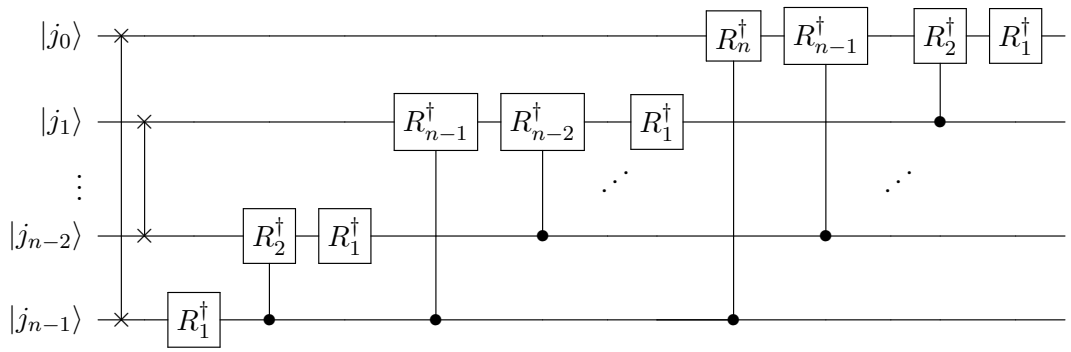


Figure 7. *The IQFT circuit where $R_n^\dagger = R_z\left(\frac{-\pi}{2^{n-1}}\right)$. Note that $R_1^\dagger = H$.*

Proof. Since QFT can be represented as a unitary matrix, its inverse is the conjugate transpose of QFT. Therefore, the IQFT circuit is constructed by applying the dagger operation to the QFT circuit in Figure 7. ∎

5. Quantum Phase Estimation. The quantum phase estimation (QPE) algorithm is one of the most fundamental procedures in quantum computing. Like the HT, it estimates the phase φ in an eigenvalue of a unitary operator U, i.e., when

$$U|\psi\rangle = e^{2\pi i\varphi}|\psi\rangle, \tag{5.1}$$

where $|\psi\rangle$ is a known eigenstate. Similar to that of the HT, our analysis of QPE utilizes the exact target state $|\psi\rangle$, though in practical applications, an approximate ansatz is used. Provided this approximation has its dominant overlap with the target state $|\psi\rangle$, QPE yields the correct phase with probability proportional to the squared overlap of the ansatz on the target state (see, e.g., [3, 14, 21]). QPE generalizes the HT by using a register of ancilla qubits to extract multiple bits of φ, thereby increasing the resolution of the estimate $\theta = 2\pi\varphi$. It forms the foundation of many quantum algorithms, including those for factoring, linear systems, and using Hamiltonian simulation to extract energy spectra (see, e.g., [1, 3, 14, 21, 22]).

5.1. The QPE Algorithm.

Definition 5.1. *Let $U \in \mathbb{C}^{n\times n}$ be unitary. The quantum phase estimation algorithm is defined via the circuit in Figure 8.*

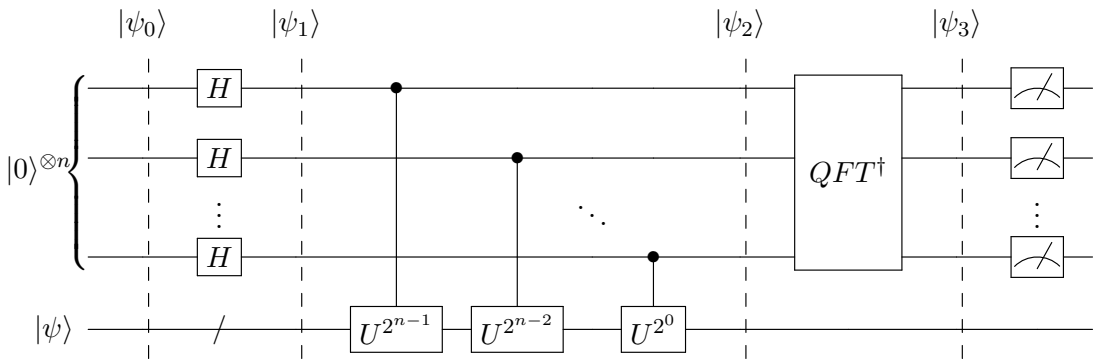


Figure 8. *Quantum phase estimation circuit where H is the Hadamard gate, $U \in \mathbb{C}^{n\times n}$ is unitary, and $QFT^\dagger$ is the inverse QFT. The top qubits initialized to the $|0\rangle$ state are the ancilla qubits and the bottom qubits represent the state $|\psi\rangle$.*

Proposition 5.2. *The final state of the circuit in Figure 8 is*

$$|\psi_3\rangle = \frac{1}{2^n}\sum_{x=0}^{2^n-1}\sum_{k=0}^{2^n-1} e^{2\pi i k(\varphi - \frac{x}{2^n})}|x\rangle|\psi\rangle. \tag{5.2}$$

Proof. The initial state of the circuit in Figure 8 can be represented as

$$|\psi_0\rangle = |0\rangle^{\otimes n}|\psi\rangle. \tag{5.3}$$

A Hadamard gate is then applied to all n ancillas generating a uniform superposition, i.e.,

$$|\psi_1\rangle = \frac{1}{\sqrt{2^n}}(|0\rangle + |1\rangle)^{\otimes n}|\psi\rangle = |+\rangle^{\otimes n}|\psi\rangle. \tag{5.4}$$

Next, the controlled U gates are applied. Since $U|\psi\rangle = e^{2\pi i\varphi}|\psi\rangle$, this yields

$$U^{2^j}|\psi\rangle = (e^{2\pi i\varphi})^{2^j}|\psi\rangle = e^{2\pi i 2^j\varphi}|\psi\rangle. \tag{5.5}$$

Therefore, by controlling this gate and applying it to all n qubits, it only affects the state for the case where the ancilla is $|1\rangle$. Applying the controlled U^{2^j}, with $0 \leq j \leq n-1$, yields

$$|\psi_2\rangle = \frac{1}{\sqrt{2^n}}\left(|0\rangle + e^{2\pi i\varphi 2^{n-1}}|1\rangle\right)\otimes\cdots\otimes\left(|0\rangle + e^{2\pi i\varphi 2^0}|1\rangle\right)\otimes|\psi\rangle = \frac{1}{\sqrt{2^n}}\sum_{k=0}^{2^n-1} e^{2\pi i\varphi k}|k\rangle|\psi\rangle, \tag{5.6}$$

where k is the integer representation of an n-bit binary number. The IQFT is then applied resulting in the state

$$\frac{1}{\sqrt{2^n}}\sum_{k=0}^{2^n-1} e^{2\pi i\varphi k}|k\rangle|\psi\rangle \xrightarrow{\mathrm{QFT}_N^\dagger} \frac{1}{2^n}\sum_{x=0}^{2^n-1}\sum_{k=0}^{2^n-1} e^{2\pi ik(\varphi-\frac{x}{2^n})}|x\rangle|\psi\rangle = |\psi_3\rangle. \tag{5.7}$$

■

Remark 5.3. QPE with one ancilla qubit yields the same circuit as the real HT.

Example 5.4. To test QPE, 4 ancilla qubits with 10^3 shots were used on the noiseless Qiskit AerSimulator [9], estimating the phase angles $\varphi = 1/4$ and $\varphi = 2/3$ for the unitary matrix

$$U = \begin{bmatrix} 1 & 0 \\ 0 & e^{i2\pi\varphi} \end{bmatrix}. \tag{5.8}$$

After a sampling, the probability of measuring each specific state was calculated by dividing the number of times each state was counted by the total number of shots.

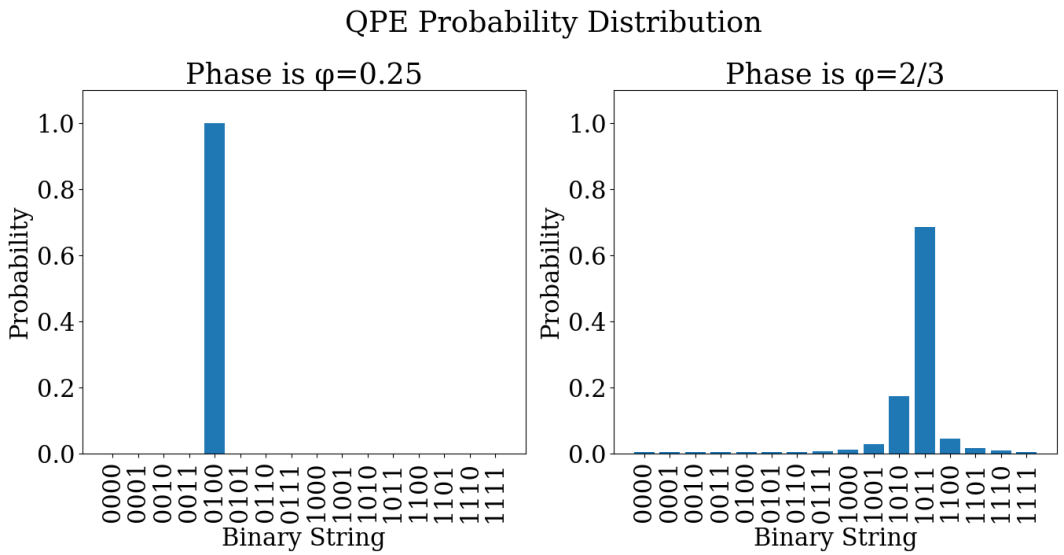


Figure 9. *QPE probability distribution obtained from the Qiskit AerSimulator applied to a state with a phase angle of $\varphi = 1/4$ (left) and $\varphi = 2/3$ (right).*

Since the phase angle $\varphi = \frac{1}{4}$ can be exactly represented with 2 binary digits, using 4 ancilla qubits in QPE returns the phase angle deterministically, see left panel in Figure 9.

However, since the phase angle $\varphi = \frac{2}{3}$ cannot be exactly represented with any number of binary digits, the outcome is probabilistic, and thus multiple measurements are required, see right panel in Figure 9. In such cases where there are multiple bit-strings with non-zero probability, one can use the maximum-likelihood estimate. If greater precision is desired, one can take the expectation value of the output bit-strings. For phases without an exact binary bit expansion, the variance of both the maximum-likelihood and expectation value estimation methods decrease as the number of shots increase. For the sake of simplicity, in the following sections we use the maximum-likelihood estimate.

5.2. Error Analysis. QPE yields a distribution over estimates centered around the true phase. However, when the targeted phase has no exact binary bit expansion, QPE returns the nearest binary approximation with high probability. Quantifying this error is the subject of this section. Throughout this section, we make the assumption that

$$U|\psi_j\rangle = e^{i2\pi\varphi_j}|\psi_j\rangle, \tag{5.9}$$

where $0 \leq \varphi_0 \leq \varphi_1 \cdots \leq \varphi_{N-1} \leq 1$, and that φ_0 does not have an exact d-bit representation. QPE is moreover applied to the initial state $|0^t\rangle|\psi_0\rangle$ where $t > d$.

Proposition 5.5. *In order to obtain the phase φ_0 with an accuracy of $\epsilon = 2^{-d}$ and a success probability of at least $1-\delta$, a total of $t = d + \log_2(\delta^{-1})$ ancilla qubits are necessary to store the phase value. Furthermore, because $T\epsilon = \delta^{-1}$, the simulation run time must be $T = (\epsilon\delta)^{-1}$.*

Proof. Performing QPE on the initial state $|0\rangle^{\otimes t}|\psi_0\rangle$ gives

$$\begin{aligned} |0\rangle^{\otimes t}|\psi_0\rangle &\xrightarrow{H^{\otimes t}\otimes I} \frac{1}{\sqrt{2^t}}\sum_{j\in[2^t]}|j\rangle|\psi_0\rangle \\ &\xrightarrow{U} \frac{1}{\sqrt{2^t}}\sum_{j\in[2^t]}|j\rangle e^{i2\pi j\varphi_0}|\psi_0\rangle \\ &\xrightarrow{\mathrm{QFT}^\dagger} \sum_{k'\in[2^t]}\left(\frac{1}{2^t}\sum_{j\in[2^t]}e^{i2\pi j\left(\varphi_0-\frac{k'}{2^t}\right)}\right)|k'\rangle|\psi_0\rangle. \end{aligned} \tag{5.10}$$

For simplicity, it can be said that

$$|0\rangle^{\otimes t}|\psi_0\rangle \xrightarrow{\mathrm{QPE}} \sum_{k'}\gamma_{0,k'}|k'\rangle|\psi_0\rangle, \quad \text{with} \quad \gamma_{0,k'} = \frac{1}{2^t}\sum_{j\in[2^t]}e^{i2\pi j\left(\varphi_0-\frac{k'}{2^t}\right)}. \tag{5.11}$$

Recognizing that $\gamma_{0,k'}$ is a geometric series with the common ratio $r = e^{i2\pi\left(\varphi_0-\frac{k'}{2^t}\right)}$ and the coefficient $a = \frac{1}{2^t}$, we find that

$$\gamma_{0,k'} = \frac{1}{2^t}\frac{1-r^{2^t}}{1-r} = \frac{1}{2^t}\frac{1-e^{i\pi 2^{t+1}(\varphi_0-\tilde{\varphi}_{k'})}}{1-e^{i2\pi(\varphi_0-\tilde{\varphi}_{k'})}}, \quad \text{with} \quad \tilde{\varphi}_{k'} = \frac{k'}{2^t}, \tag{5.12}$$

Since $e^{i2\pi x}$ is a 1-periodic function, $x \mod 1$ is the only measurable value. Therefore, when estimating the phase k'_0, we seek the precision $\epsilon = \frac{1}{2^d} = \frac{2^{t-d}}{2^t}$ such that

$$\left|\varphi_0 - \tilde{\varphi}_{k'_0}\right|_1 < \epsilon. \tag{5.13}$$

Since for any $\theta \in [-\pi, \pi]$

$$\left|1 - e^{i\theta}\right| = \sqrt{(1-\cos\theta)^2 + \sin^2\theta} = \sqrt{2(1-\cos\theta)} = 2\left|\sin\frac{\theta}{2}\right| \geq \frac{2}{\pi}|\theta|, \tag{5.14}$$

and

$$\left|1 - e^{i\theta}\right| \leq |1| + \left|e^{i\theta}\right| = \sqrt{1^2} + \sqrt{\cos^2\theta - (i\sin\theta)^2} = 1 + \sqrt{\cos^2\theta + \sin^2\theta} = 1 + 1 = 2, \tag{5.15}$$

we find that

$$\left|\gamma_{0,k'}\right| \leq \frac{2}{2^t 2\pi \frac{2}{\pi}\left|\varphi_0 - \tilde{\varphi}_{k'}\right|_1} = \frac{1}{2^{t+1}\left|\varphi_0 - \tilde{\varphi}_{k'}\right|_1}. \tag{5.16}$$

The binary measurement outcome k_0' can be viewed as a random variable where the probability of obtaining $\tilde{\varphi}_{k_0'}$ at least a distance ϵ away from the expected value φ_0 is

$$\mathbb{P}\left(\left|\varphi_0 - \tilde{\varphi}_{k_0'}\right|_1 \geq \epsilon\right) = \sum_{|\varphi_0 - \tilde{\varphi}_{k'}| \geq \epsilon} \left|\gamma_{0,k'}\right|^2 \leq \left(\frac{1}{2^{t+1}\left|\varphi_0 - \tilde{\varphi}_{k'}\right|_1}\right)^2 = \frac{1}{4T^2\left|\varphi_0 - \tilde{\varphi}_{k'}\right|_1^2}, \tag{5.17}$$

where $T = 2^t$. In other words, this expression represents the probability that the estimated phase is not within the desired error. Setting $x = \left|\varphi_0 - \tilde{\varphi}_{k'}\right|_1$, this yields

$$\mathbb{P}\left(|x| \geq \epsilon\right) = \sum_{|x|\geq\epsilon} \frac{1}{4T^2x^2} =: \sum_{|x|\geq\epsilon} f(x). \tag{5.18}$$

Since f is a symmetric function,

$$\sum_{|x|\geq\epsilon} \frac{1}{4T^2x^2} = \sum_{x\geq\epsilon} \frac{2}{4T^2x^2} = \sum_{x\geq\epsilon} \frac{1}{2T^2x^2}. \tag{5.19}$$

The sum in (5.19) is upper bounded by the integral from ϵ onwards, we visualize this in Example 5.7. Noting that

$$\sum_{x\geq\epsilon} \frac{1}{x^2} \leq \int_\epsilon^\infty \frac{1}{x^2}dx = \frac{1}{\epsilon}, \tag{5.20}$$

the expression in (5.19) can be bounded, i.e., $T \geq 1$ yields

$$\sum_{|x|\geq\epsilon} \frac{1}{4T^2x^2} \leq \frac{1}{2T^2}\frac{1}{\epsilon} \leq \frac{1}{2T\epsilon}. \tag{5.21}$$

Since $T\epsilon = 2^{t-d}$, we set $t - d = \log_2(\delta^{-1})$ so that $T\epsilon = \delta^{-1}$. For $0 < \delta < 1$, we then find

$$\mathbb{P}\left(\left|\varphi_0 - \tilde{\varphi}_{k_0'}\right|_1 \geq \epsilon\right) \leq \frac{1}{2T\epsilon} = \frac{\delta}{2} \leq \delta. \tag{5.22}$$

Therefore, in order to obtain the phase φ_0 to an accuracy of $\epsilon = 2^{-d}$ with a success probability of at least $1 - \delta$, a total of $t = d + \log_2(\delta^{-1})$ ancilla qubits are necessary to store the phase value. Furthermore, since $T\epsilon = \delta^{-1}$, the simulation run time must be $T = (\epsilon\delta)^{-1}$. ■

Example 5.6. Set $\varphi_0 = 0.35$ (not exactly representable in binary). For $t = 6$ and $t = 10$, the QPE distribution $|\gamma_{0,k'}|$ in (5.16) becomes increasingly peaked around the closest binary approximation to φ_0, see Figure 10. As $t \to \infty$, this distribution converges to a Dirac delta. In fact, for the case where φ_0 has an exact d-bit representation (meaning that $\varphi_0 = \tilde{\varphi}_{k'_0}$ for some k'_0), then $\gamma_{0,k'} = \delta_{k',k'_0}$ (the Dirac delta function). Inspecting the function in (5.7), we see that it peaks at $x = 2^n\varphi$, which is the input state.

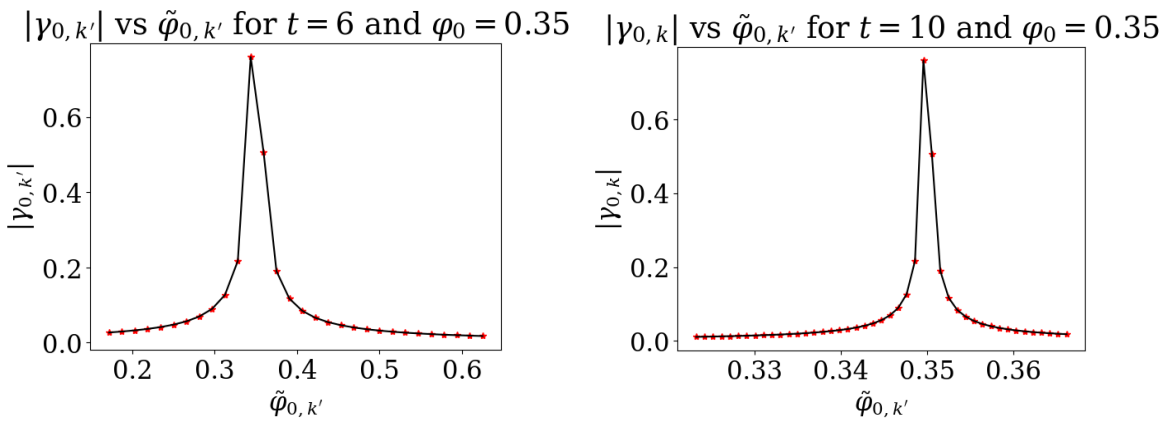


Figure 10. $|\gamma_{0,k'}|$ *plotted against* $\tilde{\varphi}_{k'_0}$ *for* $\varphi_0 = 0.35$. *(Left) With* $t = 6$, *and (Right) with* $t = 10$ *ancilla qubits.*

Example 5.7. Let $\epsilon = 2^{-5}$, in Figure 11 we visualize $f(x)$ defined in (5.18).

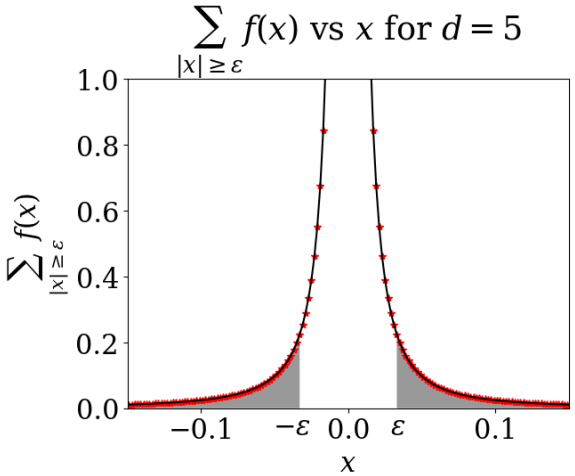


Figure 11. $\sum_{|x|\geq\epsilon} f(x)$ *plotted against* x *for* $\epsilon = 2^{-5}$.

6. Hadamard Test vs. Quantum Phase Estimation. The Hadamard Test (HT) and Quantum Phase Estimation (QPE) both serve to estimate the phase φ in an eigenvalue of the form $e^{i2\pi\varphi}$ of a unitary operator U. In this section, we compare the two algorithms with

respect to accuracy and runtime, using both classical simulations and real quantum hardware (IBM Eagle One system at RPI, known as `ibm_rensselaer`).

To ensure a fair comparison, both algorithms are used to estimate the same eigenphase $\varphi = 1/3$ of the unitary matrix

$$U = \begin{bmatrix} 1 & 0 \\ 0 & e^{i2\pi\varphi} \end{bmatrix}, \tag{6.1}$$

to precision ϵ.

6.1. Accuracy Comparison. A direct comparison of the error between the Hadamard Test (HT) and Quantum Phase Estimation (QPE) is nontrivial due to the fundamentally different nature of their errors: HT exhibits sampling error, while QPE exhibits approximation error based on binary resolution. If the desired phase has an exact d bit binary expansion, QPE with $t = d$ ancilla qubits will return the phase deterministically. To compare the convergence of these algorithms, we choose the phase $\varphi = 1/3$ as it does not have an exact binary bit expansion. Therefore, the output of QPE will asymptotically approach the true phase, similar to the sampling based convergence of HT. To visualize the convergence behavior of each algorithm as a function of its quantum resources, we plot their respective errors on a shared graph. For this comparison, we run each algorithm over 20 independent trials. The HT is tested with shot counts ranging from 10^3 to 10^5, while QPE is tested using 1 to 20 ancilla qubits, each run with 1000 shots. All experiments were conducted both on Qiskit's AerSimulator [9] and on `ibm_rensselaer`, an IBM Quantum System One Eagle processor [9] (see Figure 12). These parameters were selected to span a realistic range of quantum resources given current simulation and hardware constraints.

Up to this point, the reported data has been gathered by classical simulation corresponding to a hypothetical fault-tolerant quantum computer, but does this still hold on a real quantum device?

As shown in the left panel of Figure 12, the Hadamard test (HT) exhibits excellent agreement between the observed and theoretically predicted error across all tested configurations, even when executed on real quantum hardware. In contrast, quantum phase estimation (QPE) deviates significantly from theoretical predictions when the number of ancilla qubits exceeds $t = 3$. For $t \leq 3$, QPE performs as expected, with error decreasing according to the predicted $\mathcal{O}(2^{-t})$ scaling. However, for $t > 3$, the error no longer decreases and may even worsen with additional ancilla qubits.

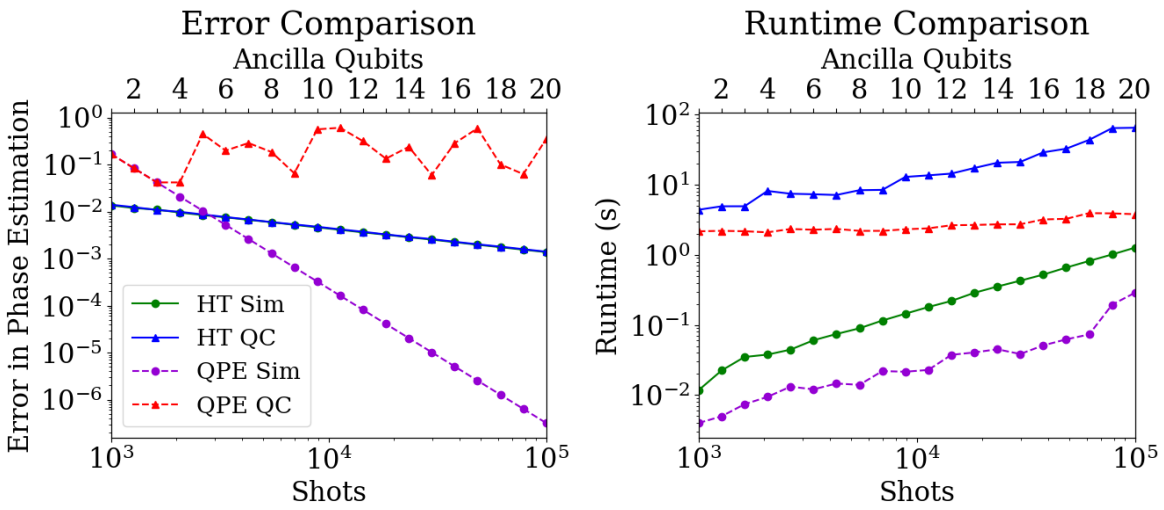


Figure 12. *(Left) A comparison of the approximation error, and (Right) a comparison of the runtime. We present data for the HT and QPE algorithms obtained from a quantum emulator and an IBM Eagle 127-qubit quantum processor.*

This degradation is attributed to the growing circuit depth and gate count: Each added ancilla qubit introduces a new layer of controlled U^{2^j} operations, many of which are decomposed into multiple two-qubit gates. On current noisy intermediate-scale quantum (NISQ) hardware, this increased gate complexity significantly amplifies noise and decoherence, undermining QPE's theoretical advantages. To enable execution of our circuit on a real quantum processing unit (QPU), we employ Qiskit's *transpiler*. A transpiler is a program that transforms an abstract quantum circuit into a physically executable one by mapping logical gates to the target hardware's basis gates and connectivity graph while optimizing the circuit to minimize gate depth and error rates. To reduce the impact of noise, we used various `optimization_level` settings (`0-3`) of the transpiler, which aim to simplify the circuit and reduce the total gate count to varying degrees. Although higher optimization levels (e.g., `optimization_level = 3`) are effective at minimizing noise in many applications, in this experiment they were insufficient to recover QPE performance beyond $t = 3$. This suggests that current hardware remains too noisy to realize the full potential of QPE in regimes that require deeper circuits.

To better understand the performance degradation of QPE on hardware, it is useful to analyze the circuit depth, which serves as a proxy for accumulated gate error and decoherence effects. While circuit depth captures the overall number of sequential operations, the primary contributors to hardware error are two-qubit gates, which are more error-prone than single-qubit gates and dominate the noise profile in most NISQ devices [9]. As illustrated in Figure 13, the number of two-qubit gates in QPE circuits grows rapidly with each additional ancilla qubit. Although Qiskit's higher transpiler optimization levels (2 and 3) significantly reduce this gate count compared to lower levels, the improvement is not sufficient beyond $t = 3$. In particular, circuits for $t \geq 4$ remain too deep—despite aggressive optimization—to yield reliable output

on current quantum hardware, underscoring the practical limitations of QPE in its standard form on today's machines.

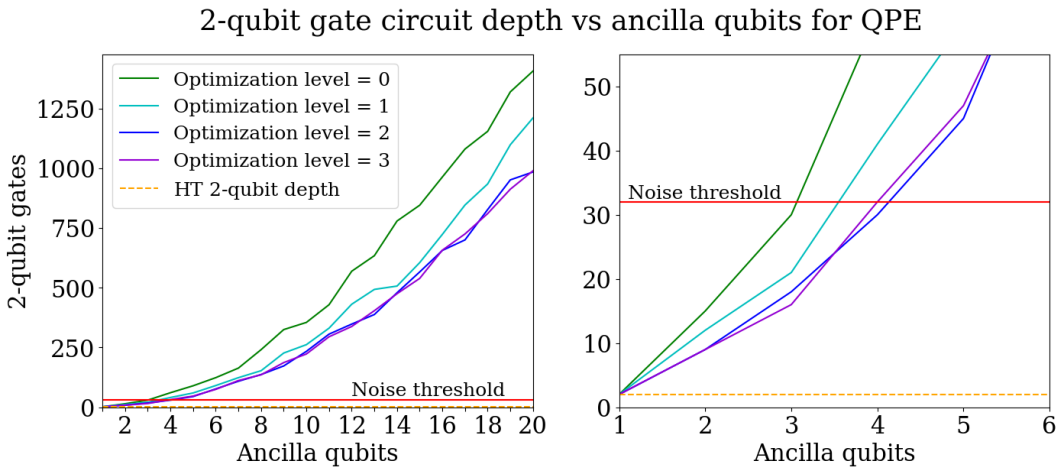


Figure 13. *2-qubit gate circuit depth plotted against the number of ancilla qubits for QPE. The "noise threshold" represents the shortest 2-qubit circuit depth where the qubit operations create too much noise to correctly perform their operations.*

6.2. Runtime Comparison. Runtime is measured as the total execution time for a fixed accuracy target. Although HT and QPE have different resource requirements (shots vs. ancilla qubits), they share a dependence on the physical capabilities of the backend.

The right panel in Figure 12 shows a runtime comparison of HT and QPE on a quantum emulator and an IBM Eagle 127-qubit quantum processor. For HT, the simulated experiment requires less runtime per shot compared to its quantum counterpart. Since the total runtime of HT scales directly with the runtime per shot, we observe that the runtime of HT on the quantum computer exhibits an offset, resulting in a runtime consistently higher than that of the quantum emulator. For QPE, the classical simulation runtime is initially lower than the runtime on the quantum computer but increases exponentially. On the quantum computer, the QPE runtime remains close to constant.

Note that for $t = 1$ ancilla qubits, the QPE circuit is structurally equivalent to the real HT circuit. However, in practice, the total runtime of HT is observed to be approximately twice that of QPE at this setting. This discrepancy arises because HT requires two separate circuits—one to estimate the real part and another to estimate the imaginary part of the expectation value—in order to reconstruct the full complex phase. In contrast, QPE encodes the phase directly into a binary register, requiring only one circuit to extract the phase information.

We emphasize that to increase the precision of the estimated phase on the quantum emulator, the runtime of both the HT and QPE will increase. However, on the quantum computer, regardless of the value of t, QPE's runtime remains nearly constant with respect to the number of qubits, while HT's runtime linearly increases with the number of shots.

Remark 6.1. Even with QPE's inaccuracy, its runtime is always very low on a quantum

machine. Therefore, if the noise can somehow be mitigated better as the ancillary qubits increase to $d > 3$, the QPE algorithm will become extremely effective in estimating phases with high binary bit precision. Since QPE is unreliable for binary representations of $d > 3$ bits, unless the phase can be exactly represented with $d \leq 3$ bits, the HT will give higher precision. Since HT is always slower than QPE, if the phase can be exactly represented with $d \leq 3$ bits, QPE will give higher precision and lower runtime.

7. Conclusion. This work was guided by the lecture notes of Lin on Quantum Algorithms for Scientific Computing [1], which provided a conceptual foundation for our exploration of quantum phase estimation and its practical implementation. Building on this framework, we presented a pedagogical yet rigorous initial study of two essential quantum algorithms: the *Hadamard Test* (HT) and *Quantum Phase Estimation* (QPE). After introducing the mathematical structure and circuit implementations of both algorithms, we evaluated their performance in realistic computational settings.

A key contribution of this work is a detailed comparison of HT and QPE on both simulated and real quantum hardware, in our case, the IBM Quantum System One at RPI (`ibm_rensselaer`) [9], a 127-qubit Eagle-class quantum processor. Our main findings are:

- On ideal fault-tolerant simulators, HT and QPE perform in accordance with theoretical expectations: QPE shows exponential accuracy scaling in the number of ancilla qubits, while HT exhibits convergence consistent with Monte Carlo sampling error.
- On real hardware, HT remains robust across a wide range of shot counts, closely matching simulated behavior. QPE, however, deviates significantly from its ideal performance once the number of ancilla qubits exceeds $t = 3$, due to increased circuit depth and accumulation of two-qubit gate errors.
- Analysis of gate depth and transpiler optimization revealed a steep rise in the number of two-qubit gates with increasing QPE register size. Despite leveraging Qiskit's highest transpilation optimization level, QPE circuits with $t \geq 4$ proved too deep to maintain fidelity on Eagle-class hardware.

Together, these results provide a realistic benchmark for basic quantum linear algebra methods on near-term quantum devices. While QPE remains a cornerstone of long-term algorithmic development, our experiments suggest that sampling-based methods like HT currently offer greater resilience to hardware noise, making them more suitable for early applications of quantum computing in scientific computing. As hardware matures and deeper circuits become feasible, these findings will serve as a reference point for re-evaluating algorithmic choices in the quantum simulation of physical systems.

Acknowledgments. The authors are thankful for the useful discussions with Dr. Cameron Cogburn, Dr. Osama Raisuddin, and Mr. Ibrahim Abdulazeez.

REFERENCES

[1] Lin Lin. Lecture notes: Quantum algorithms for scientific computing, 2024. Available at https://arxiv.org/abs/2201.08309.

[2] Mauro E S Morales, Lirandë Pira, Philipp Schleich, et al. Quantum linear system solvers: A survey of algorithms and applications. *arXiv preprint arXiv:2411.02522v1*, 2024.

[3] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 10th anniversary edition, 2010.

[4] Thomas G. Wong. *Introduction to Classical and Quantum Computing*. Rooted Grove, Omaha, Nebraska, 2022.

[5] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.*, 103:150502, 2009.

[6] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond. In *Proc. 51st ACM Symposium on Theory of Computing (STOC)*, pages 193–204, 2019.

[7] Shi Jin and Nana Liu. Quantum simulation of discrete linear dynamical systems and simple iterative methods in linear algebra. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 480(2292):20230370, 2024.

[8] John Preskill. Quantum computing in the NISQ era and beyond. *Quantum*, 2:79, 2018.

[9] Ali Javadi-Abhari, Matthew Treinish, Kevin Krsulich, Christopher J. Wood, Jake Lishman, Julien Gacon, Simon Martiel, Paul D. Nation, Lev S. Bishop, Andrew W. Cross, Blake R. Johnson, and Jay M. Gambetta. Quantum computing with Qiskit, 2024.

[10] IBM Quantum. IBM unveils 127-qubit *eagle* quantum processor, 2021. IBM Research Blog.

[11] Alexander Weiss and Paul Bruzzi. Intro to quantum linear algebra, 2025. https://github.com/Quantum-Linear-Algebra/Intro-to-Quantum-Linear-Algebra.

[12] Zhiyan Ding and Lin Lin. Even shorter quantum circuit for phase estimation on early fault-tolerant quantum computers with applications to ground-state energy estimation. *PRX Quantum*, 4:020331, Jun 2023.

[13] Sam McArdle, Suguru Endo, Alán Aspuru-Guzik, Simon C. Benjamin, and Xiao Yuan. Quantum computational chemistry. *Rev. Mod. Phys.*, 92:015003, Mar 2020.

[14] Alán Aspuru-Guzik, Anthony D. Dutoi, Peter J. Love, and Martin Head-Gordon. Simulated quantum computation of molecular energies. *Science*, 309(5741):1704–1707, 2005.

[15] Richard Cleve, Artur Ekert, Chiara Macchiavello, and Michele Mosca. Quantum algorithms revisited. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 454(1969):339–354, 1998.

[16] Kosuke Mitarai and Keisuke Fujii. Methodology for replacing indirect measurements with direct measurements. *Phys. Rev. Res.*, 1:013006, Aug 2019.

[17] Joseph K. Blitzstein and Jessica Hwang. *Introduction to Probability*. CRC Press, Boca Raton, Florida, second edition, 2019.

[18] Ronald N. Bracewell. *The Fourier Transform and its Applications*. McGraw-Hill, 3rd edition, 2000.

[19] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, October 1997.

[20] Guang Hao Low and Isaac L. Chuang. Optimal Hamiltonian simulation by quantum signal processing. *Physical Review Letters*, 118(1):010501, 2017.

[21] Daniel S. Abrams and Seth Lloyd. Quantum algorithm providing exponential speed increase for finding eigenvalues and eigenvectors. *Phys. Rev. Lett.*, 83:5162–5165, Dec 1999.

[22] various authors. *Qiskit Textbook*. Github, 2023.