Smart Post Processing for Algorithms with QPE

Viraj Dsouza

Quantum Phase Estimation (QPE) has been the key for many quantum algorithms. Suppose a unitary operator U has an eigenvector $|u\rangle$ with eigenvalue $e^{2\pi i\phi}$ where ϕ is unknown, that is, $U|u\rangle = e^{2\pi i\phi}\,|u\rangle$. The goal of the phase estimation algorithm is to estimate ϕ.

To understand the procedure involved in QPE, its useful to review Quantum Fourier Transform (QFT) which is an essential ingredient in enabling QPE.

1 Introduction to Quantum Fourier Transform:

A Discrete Fourier transform (DFT) takes in a input vector of complex numbers and outputs a transformed vector of complex numbers. In the case of QFT, the set of input complex numbers are amplitudes of a quantum state.

Definition 1.1. *DFT of the vector $x = (x_0\ x_1 \ldots\ x_{N-1})^T$ is the complex vector $y = (y_0\ y_1 \ldots y_{N-1})^T$ where*

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{\frac{2\pi ijk}{N}} x_j. \tag{1}$$

Now suppose that we have an $N = 2^n$-dimensional quantum state vector $|\psi\rangle$, represented by the amplitudes $x = (x_0\ x_1 \ldots\ x_{N-1})^T$, that is

$$|\psi\rangle = \sum_{j=0}^{N-1} x_j\,|j\rangle\,.$$

Definition 1.2. *Quantum Fourier Transform QFT of state $|\psi\rangle$ is given by*

$$|\phi\rangle = \sum_{k=0}^{N-1} y_k\,|k\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \sum_{j=0}^{N-1} e^{\frac{2\pi ijk}{N}} x_j\,|k\rangle\,, \tag{2}$$

where y_k is defined as 1 for $k = 0, \ldots, N-1$.

For convenience, we denote $e^{\frac{2\pi i}{N}}$ by ω. With this, the above expression can be expressed as

$$|\phi\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \sum_{j=0}^{N-1} \omega^{jk} x_j |k\rangle .$$

Given the entries of the vector x, the entries of the vector y corresponding to QFT of the vector x is thus given by,

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{\frac{2\pi ijk}{N}} x_j = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \omega^{jk} x_j.$$

As an example to illustrate QFT, let's take QFT of the general single qubit quantum state given by $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$.

The quantum state $|\psi\rangle$ is represented by $(\alpha\ \beta)^T$ where $x_0 = \alpha$ and $x_1 = \beta$. Given x_0 and x_1, we calculate y_0 and y_1 from the above expression.

$$y_0 = \frac{1}{\sqrt{2}} \sum_{j=0}^{1} e^{\frac{2\pi ij\cdot 0}{2}} x_j = \frac{\alpha + \beta}{\sqrt{2}}$$

$$y_1 = \frac{1}{\sqrt{2}} \sum_{j=0}^{1} e^{\frac{2\pi i\cdot j\cdot 1}{2}} x_j = \frac{1}{\sqrt{2}} \left(e^{\frac{2\pi i\cdot 1\cdot 0}{2}} x_0 + e^{\frac{2\pi i\cdot 1\cdot 1}{2}} x_1 \right) = \frac{\alpha - \beta}{\sqrt{2}}$$

Hence the new state is $\frac{\alpha+\beta}{\sqrt{2}} |0\rangle + \frac{\alpha-\beta}{\sqrt{2}} |1\rangle$. One can recognize that this is exactly the same state obtained after applying Hadamard $|\psi\rangle$. Hence, the effect of applying QFT to a single qubit is equivalent to applying a Hadamard gate.

1.1 Circuit Implementation of QFT

Now we will look at the effect of applying QFT to a n-qubit system. Let $N = 2^n$.

Let $|j\rangle$ be a basis state where $j = 0, \ldots, N-1$. We will use the binary representation for j, that is $j = j_1 j_2 \cdots j_n$ for $j_i \in \{0, 1\}$. More formally

$$j = j_1 2^{n-1} + j_2 2^{n-2} + \cdots + j_n 2^0 = 2^n \sum_{l=1}^{n} 2^{-l} j_1 \tag{3}$$

For example if $j = 3$ and we want to represent this using $n = 2$ qubits, then $3 = 1(2^{2-1}) + 1(2^{2-2})$ which implies that $j_1 = 1$ and $j_2 = 1$. Hence the binary representation of 3 using 2 qubits is 11.

We use the representation given by 3 to express the state obtained after applying QFT to $|j\rangle$.

Now $|j\rangle = |j_1 j_2 \cdots j_n\rangle$ and after QFT

$$|j\rangle \rightarrow \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{\frac{2\pi ijk}{N}} |k\rangle$$

Replacing k using 3 and using $N = 2^n$ we get,

$$\frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{\frac{2\pi ijk}{N}} |k\rangle = \frac{1}{2^{n/2}} \sum_{k_1=0}^{1} \sum_{k_2=0}^{1} \cdots \sum_{k_n=0}^{1} e^{2\pi ij(\sum_{l=1}^{n} k_l 2^{-l})} |k_1 k_2 \cdots k_n\rangle$$

Writing the quantum state as the tensor product of n qubits, the above expression reduces to,

$$= \frac{1}{2^{n/2}} \sum_{k_1=0}^{1} \sum_{k_2=0}^{1} \cdots \sum_{k_n=0}^{1} \prod_{l=1}^{n} e^{2\pi ijk_l 2^{-l}} |k_1 k_2 \cdots k_n\rangle$$

$$= \frac{1}{2^{n/2}} \otimes_{l=1}^{n} \sum_{k_o=0}^{1} e^{2\pi ijk_l 2^{-l}} |k_o\rangle = \frac{1}{2^{n/2}} \otimes_{l=1}^{n} \Big(|0\rangle + e^{2\pi ij 2^{-l}} |1\rangle \Big)$$

$$= \frac{1}{2^{n/2}} \Big(\big(|0\rangle + e^{2\pi ij2^{-1}} |1\rangle\big) \otimes \big(|0\rangle + e^{2\pi ij2^{-2}} |1\rangle\big) \otimes \cdots \otimes \big(|0\rangle + e^{2\pi ij2^{-n}} |1\rangle\big) \Big) \quad (4)$$

To implement the above operations through a quantum circuit, we define the two qubit CR_k operator which puts a relative phase of $e^{\frac{2\pi i}{2^k}}$ in front of the quantum state if both the controlled and the target qubits are in state $|1\rangle$. The CR_k gate corresponds to a rotation around z-axis and has the following matrix representation:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{\frac{2\pi i}{2^k}} \end{pmatrix} \quad (5)$$

Let $|j_1 j_2 \cdots j_n\rangle$ be the input state. We'll start witht the first qubit.
Applying Hadamard to the first qubit, we get

$$\rightarrow \frac{1}{2^{1/2}} \big(|0\rangle + e^{2\pi ij_1 2^{-1}} |1\rangle\big) |j_2 \cdots j_n\rangle$$

As we can see, if $j_1 = 0$ the sign before $|1\rangle$ is $+$ and if $j_1 = 1$, the sign before $|1\rangle$ is $-$ as needed.

Now let's apply the operator CR_2 where second qubit is the control and the first qubit is the target. The resulting state is,

$$\rightarrow \frac{1}{2^{1/2}} \big(|0\rangle + e^{2\pi i(j_1 2^{-1} + j_2 2^{-2})} |1\rangle\big) |j_2 \cdots j_n\rangle$$

We then apply CR_i operator where qubit i is the control qubit and the first qubit is the target, consecutively for $i = 3 \dots n$. We then obtain the state,

$$\rightarrow \frac{1}{2^{1/2}}\left(|0\rangle + e^{2\pi i(j_1 2^{-1} + j_2 2^{-2} + j_3 2^{-3} \dots j_n 2^{-n})}|1\rangle\right)|j_2 \cdots j_n\rangle \tag{6}$$

$$= \frac{1}{2^{1/2}}\left(|0\rangle + e^{2\pi i j 2^{-n}}|1\rangle\right)|j_2 \cdots j_n\rangle \tag{7}$$

Continuing the above process (applying Hadamard followed by successive CR_i operator) for second qubit, third qubit and so on. We end up with the below state,

$$\frac{1}{2^{n/2}}\left(|0\rangle + e^{2\pi i j 2^{-n}}|1\rangle\right) \otimes \left(|0\rangle + e^{2\pi i j 2^{-n+1}}|1\rangle\right) \otimes \cdots \otimes \left(|0\rangle + e^{2\pi i j 2^{-1}}|1\rangle\right)$$

This looks nearly the same as 4, with only difference being the qubits are in the reverse order. This can be modified by using $SWAP$ gates.

Let us briefly look into the QFT circuit complexity. For the first qubit, we apply a single Hadamard gate followed by $(n-1)$ CR gates, which makes n gates in total. For the second qubit we apply a single Hadamard gate followed by $(n-2)$ CR gates. Continuing in this way, we see that we need n Hadamard gates and $(n-1)+(n-2)+...+1+0$ CR gates (note that only one Hadamard acts on the n^{th} qubit). Thus we require a total of $\frac{n(n+1)}{2}$ gate sand furthermore, $\frac{n}{2}$ $SWAP$ gates are required, each of which can be implemented using three $CNOT$ gates. Thus the circuit provides a $\theta(n^2)$ algorithm for applying QFT to an n-qubit system represented by a vector of size $N = 2^n$.

The best known classical algorithm for computing the Discrete Fourier Transform of 2^n entries, such as Fast Fourier Transform (FFT) requires $\Theta(n2^n)$, equivalently $\Theta(N \log N)$ gates, which means that the classical algorithm requires exponentially many more operations to compute DFT.
Nevertheless, this does not mean that we can use QFT directly to accelerate the classical computation process. There are two reasons for this: The first reason is that the amplitudes can not be accessed directly after applying QFT. The second reason is that we may not know how to efficiently prepare the input state to be Fourier Transformed.
As all the gates used in the circuit implementation of QFT are unitary, QFT is a unitary transformation and has an inverse equal to $QFT^\dagger$
Inverse Quantum Fourier Transform ($QFT^\dagger$) is the transformation which satisfies $QFT \cdot QFT^\dagger = I$. Hence to implement $QFT^\dagger$, one should apply all the operations in reverse order to undo the circuit. The $QFT^\dagger$ is defined almost the same as QFT but with the exponents having a negative sign.

$$QFT^\dagger |k\rangle = \frac{1}{\sqrt{N}} \sum_{l=0}^{N-1} e^{-\frac{2\pi ijk}{N}} |l\rangle \,.$$

2 Introduction to Quantum Phase Estimation:

Consider a unitary operator U that has an eigenvector $|u\rangle$ with eigenvalue $e^{2\pi i\phi}$ where ϕ is unknown, that is, $U|u\rangle = e^{2\pi i\phi}|u\rangle$. The goal of the phase estimation algorithm is to estimate ϕ. To perform the estimation, we assume that we have available black boxes (also called *oracles*) capable of preparing the state $|u\rangle$ and performing the controlled-$U^{2^j}(CU^{2^j})$ operation, for suitable non-negative integers j.

The controlled version of the U operator is given by,

$$CU(|0\rangle|u\rangle) \rightarrow |0\rangle|u\rangle \quad \text{and} \quad CU(|1\rangle|u\rangle) \rightarrow e^{2\pi i\phi}|1\rangle|u\rangle$$

Now the matrix representation of the operator CU^{2^j} is same as 5 (if $|u\rangle$ is a one qubit state) with the replacement of k by 2^j

Thus for an arbitrary state,

$$\alpha|0\rangle|u\rangle + \beta|1\rangle|u\rangle \xrightarrow{CU} \alpha|0\rangle|u\rangle + e^{2\pi i\phi}\beta|1\rangle|u\rangle = (\alpha|0\rangle + e^{2\pi i\phi}\beta|1\rangle)|u\rangle$$

Algorithm:

QPE requires two registers where first register contains t qubits which are in state $|0\rangle$. t depends on the number of digits of accuracy and the probability of success while estimating ϕ (more on this later). Suppose that the qubits in the first register are numbered from 1 to t. Second register stores $|u\rangle$ and has as many qubits necessary to store $|u\rangle$. We are given controlled U^{2^j} operators as black-box functions.

1. Apply Hadamard to first register which has t qubits, all initialized to $|0\rangle$. The new quantum state is

$$\frac{1}{2^{t/2}}\big(|0\rangle + |1\rangle\big) \otimes \cdots \otimes \big(|0\rangle + |1\rangle\big)|u\rangle \,.$$

2. Apply CU^{2^j} gate where qubit $t-j$ is the control for $j = 0, \ldots, t-1$ and the qubits representing the state $|u\rangle$ being the target. That is,
For $j = 0$, Apply CU^{2^0} where qubit t is the control qubit.
For $j = 1$, Apply CU^{2^1} where qubit $t-1$ is the control qubit.
Continuing this process until $j = t-1$, where qubit 1 is the control, we get the following state from the first stage of phase estimation,

$$\frac{1}{2^{t/2}}\big(|0\rangle + e^{2\pi i\phi 2^{t-1}}|1\rangle\big) \otimes \cdots \otimes \big(|0\rangle + e^{2\pi i\phi 2^{1}}|1\rangle\big) \otimes \big(|0\rangle + e^{2\pi i\phi 2^{0}}|1\rangle\big)|u\rangle \quad (8)$$

The form of this equation is same as 4, hence back tracking the calculation we did to arrive at 4, we can express 8 as,

$$\frac{1}{2^{t/2}}\sum_{k=0}^{2^t-1} e^{2\pi i k\phi}\left|k\right\rangle\left|u\right\rangle$$

To realize why the phase estimation algorithm works, we express ϕ exactly in t bits. That is,

$$\phi = 0.\phi_1\ldots\phi_t = \frac{\phi_1\ldots\phi_t}{N} = \frac{x}{2^t},$$

The resulting state now is,

$$\frac{1}{2^{t/2}}\sum_{k=0}^{2^t-1} e^{\frac{2\pi i k x}{N}}\left|k\right\rangle\left|u\right\rangle.$$

The state of the first register is clearly of the form of Fourier transform of state $\left|x\right\rangle$, hence applying $QFT^{\dagger}$ to the first register, we exactly measure $\left|x\right\rangle\left|u\right\rangle = \left|\phi_1\ldots\phi_t\right\rangle\left|u\right\rangle$.

3. Apply $QFT^{\dagger}$ on the first register we get,

$$\frac{1}{2^{t/2}}\sum_{k=0}^{2^t-1} e^{\frac{2\pi i k x}{N}}\left|k\right\rangle\left|u\right\rangle \xrightarrow{QFT^{\dagger}\otimes I} \left|x\right\rangle\left|u\right\rangle = \left|\phi_1\ldots\phi_t\right\rangle\left|u\right\rangle.$$

Finally the required phase is just $\phi = \frac{x}{2^t}$.

Cost of QPE Algorithm:

The main cost of the algorithm is the implementation of U gates which are $\sum_{j=0}^{n-1} 2^j = 2^n - 1$ in total (note that this is exponential). Apart from this inverse QFT required $O(n^2)$ gates and a total of n Hadamard gates are required (the cost here is not exponential, hence can be ignored for a sufficiently large n). Thus we require $O(2^n)$ queries to know the required phase to n-bits (representing the precision of the phase calculated).
So for now we have an exact measurement of the phase without any errors.

2.1 The case where x is not an integer:

The earlier case applied to the ideal case where ϕ could be written exactly with a t bit binary expression. In this section we show that even if this is not the case (i.e $e^{i2\pi\phi} \neq e^{i2\pi(\frac{x}{2^t})}$ for an integer x) the QPE algorithm would still produce a very good *approximation* of ϕ.

Just as before, after n-Hadamard gates and CU operations will result in the state,

$$\frac{1}{2^{t/2}}\sum_{k=0}^{2^t-1} e^{2\pi i k\phi}\left|k\right\rangle\left|u\right\rangle$$

Note that this is not exactly a Fourier transformed state and applying an inverse QFT will give,

$$\frac{1}{2^t}\sum_{kk'} e^{2\pi i k\phi} e^{\frac{-2\pi i k k'}{2^t}}\left|k'\right\rangle\left|u\right\rangle \tag{9}$$

Now define:

$$\alpha_{k'} = \sum_{k=0}^{2^t-1}\frac{\left[e^{-\frac{2\pi i}{2^t}(k'-\phi 2^t)}\right]^k}{2^t}$$

This is clearly a geometric series and can be easily summed through we get,

$$\alpha_{k'} = \frac{1}{2^t}\left[\frac{1-\left[e^{\left(\frac{-2\pi i}{2^t}(k'-\phi 2^t)\right)}\right]^{2^t}}{1-\left[e^{\left(\frac{-2\pi i}{2^t}(k'-\phi 2^t)\right)}\right]}\right] \leq \frac{1}{2^t}\left[\frac{2}{1-\left[e^{\left(\frac{-2\pi i}{2^t}(k'-\phi 2^t)\right)}\right]}\right]$$

Assuming that the term $(k'-\phi 2^t)$ is bounded between $-\pi$ and $+\pi$, the above inequality can be expressed as,

$$\alpha_{k'} \leq \frac{1}{2}\left(\frac{1}{|j'-\frac{\phi 2^n}{2/pi}|}\right) \tag{10}$$

Plugging this expression back in 9 we get,

$$\frac{1}{2^t}\sum_{k'}\left[\frac{1-\left[e^{\left(\frac{-2\pi i}{2^t}(k'-\phi 2^t)\right)}\right]^{2^t}}{1-\left[e^{\left(\frac{-2\pi i}{2^t}(k'-\phi 2^t)\right)}\right]}\right]\left|k'\right\rangle\left|u\right\rangle \tag{11}$$

From this, probability of observing the state k' is given as,

$$P(k') = \frac{1}{2^{2t}}\left|\left[\frac{1-\left[e^{\left(\frac{-2\pi i}{2^t}(k'-\phi 2^t)\right)}\right]^{2^t}}{1-\left[e^{\left(-\frac{2\pi i}{2^t}(k'-\phi 2^t)\right)}\right]}\right]\right|^2 \tag{12}$$

Clearly the probability is peaked when $k' \approx \phi 2^t$. Hence we do not have an exact estimate of ϕ, but an approximate estimate.

Let us plot the probability P as a function of ϕ. Let us consider the case of $t = 3$ that is 3 target qubits in the QPE algorithm. Now the value of k' can take are integers in the range from 0 to 7.

2.2 Phase Estimation for a superposition of eigenstates

Now let's consider the case where $|u\rangle$ is a linear superposition of states $|u_p\rangle$, that is $|u\rangle = \sum_p c_p |u_p\rangle$ and the unitary operator U is such that
$U|u_p\rangle = e^{\frac{2\pi i\phi_p}{2^n}} |u_p\rangle$. Now,

$$|\psi\rangle = QPE(|0\rangle^{\otimes n}|u\rangle) = \sum_p c_p QPE(|0\rangle^{\otimes n}|u_p\rangle) = \sum_p c_p(|\phi_p\rangle |u_p\rangle)$$

Now projecting this state onto $|\phi_p\rangle |u_p\rangle$ gives the corresponding phase ϕ_p with probability $|c_p|^2$.

Calculation:

$$\begin{aligned}\langle\psi| (|\phi_p\rangle\langle\phi_p| \otimes I) |\psi\rangle &= \langle\psi| (\sum_p c_p w^* |\phi_p\rangle \langle\phi_p|\phi_p\rangle |u_p\rangle) \\ &= (\sum_{p'} c^*_{p'} \langle\phi_{p'} u_{p'}|)(\sum_p c^*_p |\phi_p\rangle \langle\phi_p|\phi_p\rangle |u_p\rangle) \\ &= \sum_{p'}\sum_p c_p{}^* c_{p'}{}^* \delta_{pp'}\delta_{pp'} = |c_p|^2\end{aligned}$$

We have used the fact that $\langle\phi_{p'}|\phi_p\rangle = \langle\psi_{p'}|\psi_p\rangle = \delta_{pp'}$

It's important to note the particular case of $\phi_p = \phi_p^{'}$ (the case where its not practical to distinguish between ϕ_p and $\phi_p^{'}$), where $p \neq p^{'}$. Doing the projection calculation as done earlier, we will find that we would get the required phase ϕ_p (or $\phi_{p'}$) with probability $|c_p|^2 + |c_{p'}|^2$.
Hence, just as before, phase associated with a given state of the superposition is given by $\phi_p = \frac{x}{2^t}$, where x is a positive integer (representing the bit string encoding the control register of the QPE circuit) and it's obtained when measurement is done on the first register in the QPE algorithm.

It's also useful to note that measuring $|\phi_p\rangle$ not just helps us find the eigenvalues of U but also the measurement causes the projection onto the eigenvector $|u_p\rangle$ of U.

From the insights gained in calculating the probability P as function of state k' in 12 we get a similar expression for probability P as function of state k in the superposition as,

$$P(k') = \frac{1}{2^{2t}} \sum_p |c_p|^2 \left| \left[\frac{1 - \left[e^{\left(\frac{-2\pi i}{2^t}(k' - \phi_p 2^t)\right)} \right]^{2^t}}{1 - \left[e^{\left(-\frac{2\pi i}{2^t}(k' - \phi_p 2^t)\right)} \right]} \right] \right|^2 \tag{13}$$

If 13 is plotted for all k', then for all ϕ_p (corresponding to the states in the superposition), the probability in 13 peaks at all k' for which $k' \approx \phi_p 2^t$.

2.3 Error in estimated phase :

Let ϵ denote the uncertainty or value of error in the estimated phase and let δ denote the probability of erroneous phase.

Let $round\left(\frac{\phi 2^n}{2\pi}\right) = b$ and let e denote the maximum tolerable error. That is we are good with the phase values in the interval $[b+e, b-e]$.

The failure probability now can be written as,

$$P_f = \sum_{j'<b-e} |\alpha_{j'}|^2 + \sum_{j'>b+e} |\alpha_{j'}|^2$$

Using 12 we get an upper bound for failure probability as,

$$P_f \leq \frac{1}{2(e-1)}$$

We now want our total probability of failure $\leq \delta$, hence

$$\frac{1}{2e} \leq \frac{1}{2(e-1)} \leq \delta$$

Hence the choice of e such $e \geq \frac{1}{2\delta}$ would suffice. With this our estimate of phase would be between,

$$\left(\frac{2\pi}{2^n}(b-e), \frac{2\pi}{2^n}(b+e)\right)$$

The uncertainty or error value (maximum error we can obtain in the estimated phase),

$$\epsilon = \frac{2\pi}{2^n}(b+e-b) = \frac{2\pi}{2^n}(e)$$

For a given quantum state $|\psi\rangle$, we can form the density matrix (ρ) of the state, by taking the outer product of the state with itself. That is,

$$\rho = |\psi\rangle \langle\psi|$$

Using density matrix formulation for the output state of the QPE circuit we obtain,

$$\rho = \left(\frac{1}{2^t}\sum_{kk'} e^{2\pi ik\phi} e^{\frac{-2\pi ikk'}{2^t}} |k'\rangle |u\rangle\right) \left(\frac{1}{2^t}\sum_{ll'} e^{-2\pi il\phi} e^{\frac{2\pi ill'}{2^t}} \langle l'| \langle u|\right)$$

This reduces to,

$$\rho = \frac{1}{2^{2t}} \sum_{kk'll'} e^{\frac{2\pi i}{2^t}(-kk'+ll')} |k'l'\rangle \langle u_k u_l| \tag{14}$$

where

$$\langle u_k| = \langle e^{-\frac{2\pi i}{2^t}(k2^t\phi)} u|$$
$$\langle u_l| = \langle e^{\frac{2\pi i}{2^t}(l2^t\phi)} u|$$

Need to write more here, what's the point of doing DM formulation?

3 Amplitude Estimation (AE) with QPE :

AE combines two ideas: Amplitude Amplification (AA) and Phase Estimation (PE). AA is usually used to boost the probability of success (in other words probability of getting a desired good state). But in AE, the AA technique is used to encode the probability of success as a phase and then QPE is used to estimate that phase.

Let's say we have a unitary operator A acting on a register of $(n+1)$ qubits such that

$$A|0\rangle_{n+1} = |\Psi\rangle = \sqrt{1-a}\,|\psi_0\rangle_n |0\rangle + \sqrt{a}\,|\psi_1\rangle_n |1\rangle \tag{15}$$

where $|\psi_0\rangle_n$ and $|\psi_1\rangle_n$ are some n-qubit normalized states and $a \in [0,1]$ is unknown and with AE we need to efficiently estimate a.

Technique from Amplitude Amplification:
We use the Grover Operator defined by $Q = AS_0A^\dagger S_{|\psi_0\rangle}$, where $S_0 = 1 - 2|0\rangle\langle 0|$ and $S_{|\psi_0\rangle} = 1 - 2|\psi_0\rangle|0\rangle\langle\psi_0|\langle 0|$, represent the reflection operators about the state $|0\rangle$ and the good state $|\psi_0\rangle$ respectively. The full action of the operator Q is to rotate the state $A|0\rangle_{n+1}$ by an angle 2θ in the $2D$ space spanned by $|\psi_0\rangle_n|0\rangle$ and $|\psi_1\rangle_n|1\rangle$, where $a = \sin^2(\theta)$.

The matrix representation of the operator Q is

$$Q = \begin{pmatrix} \cos 2\theta & \sin 2\theta \\ -\sin 2\theta & \cos 2\theta \end{pmatrix} = (\cos 2\theta)\mathbb{1} + i(\sin 2\theta)\sigma_y \tag{16}$$

From this, the eigenvalues of the Q operator can be evaluated and are found to be $\exp(+i2\theta)$ and $\exp(-i2\theta)$ with the eigenvectors say, $|\psi_+\rangle$ and $|\psi_-\rangle$ respectively. In the eigenbasis of Q we can express our state Ψ as[1]

$$\Psi = a_1|\psi_+\rangle + a_2|\psi_-\rangle$$

[1] This is because every Hermitian or Unitary matrix has a complete set of eigenvectors, implying their eigenvectors can be used as a basis to represent any state $|\Psi\rangle$

Also, the action of Q on $|\Psi\rangle$ is,

$$Q|\Psi\rangle = a_1 e^{+i2\theta}|\psi_+\rangle + a_2 e^{-i2\theta}|\psi_-\rangle \tag{17}$$

Technique from Phase Estimation:
Denote $2\theta = \frac{2\pi q}{2^m}$, where m is the size of the estimation register in the QPE algorithm.

Now applying Quantum Phase Estimation,

$$QPE(|0\rangle|\Psi\rangle) = a_1|q\rangle|\psi_+\rangle + a_2|q+2^{m-1}\rangle|\psi_-\rangle$$

Note that the phases corresponding to the eigenvectors $|\psi_+\rangle$ and $|\psi_-\rangle$ in 17 differ by angle π; hence the term $|q+2^{m-1}\rangle$ in the above expression makes sense as,

$$\frac{2\pi(q+2^{m-1})}{2^m} = \frac{2\pi q}{2^m} + \pi$$

If upon measurement of the estimation register we obtain q' and $q' > 2^{m-1}$ then we get the phase corresponding to $|\psi_-\rangle$ branch, hence our estimated q (say q_p) from the QPE algorithm is,

$$q_p = q' - 2^{m-1}$$

Thus the value of 2θ is $\frac{2\pi q_p}{2^m}$ and hence the the value of a is simply,

$$a = \sin^2\theta = \sin^2(\frac{\pi q_p}{2^m})$$

This result is exact, for the case where q_p is an integer.

3.1 The case where q_p is not an integer:

As explained in an earlier section, in such cases QPE will yield an estimate for θ (say, $\tilde{\theta}$) such that $|\theta - \tilde{\theta}| \leq \epsilon$ with a probability greater than equal to $(1-\delta)$, where ϵ is the maximum tolerable error in the phase.

It can be shown that the estimator $\tilde{a} = \sin^2\tilde{\theta}$ satisfies the following relation with a probability of atleast $\frac{8}{\pi^2}$,

$$|a - \tilde{a}| \leq \frac{\pi}{2^m} + \frac{\pi}{2^{2m}}$$

4 Option Pricing with Quantum Amplitude Estimation:

Financial options are *contracts* that give the holder the right (but not the obligation) to buy (call option) or sell (put option) a financial instrument (such as stocks) at a predetermined price (strike) before or at the expiration date.

Let's understand this with a simple example: Suppose an option contract (containing a fixed number of shares of a certain stock) is bought when the underlying stock is worth X with a target/strike price for the stock at K (here $K > X$) and an expiration date of 3 years. The buyer pays a premium to the seller to purchase this contract. Within these three years, the price of this option changes due to the stochastic nature of the stock prices underlying it. If on or before the period of 3 years the stock price crosses the threshold of K and reaches the price S (S is called the payoff and here $S > K$) the contract buyer exercises his right to buy the stock at the predetermined target price Y. In this process the buyer makes a profit of $S - K$ (minus the premium paid, for net profit). If the stock price remains below K on or before the expiry date, the buyer won't exercise the contract and would suffer a loss of the premium amount paid to the seller.

The aim of option pricing is to figure out how much an option will be worth when it reaches its end date in the future. After that, we take this future value and adjust it to find the fair value for the options contract. This fair value tells us the right amount of money we should pay now to start the option contract.

In practice, options are numerically priced using Monte Carlo methods that includes- Modelling asset price of the option's underlying as random variables $\mathbf{X} = \{X_1, X_2, ..., X_N\}$; Generating a large number M of random price paths $\{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_M\}$ from the probability distribution P followed by X. Note that $\mathbf{X_i}$ is a column vector with N rows and indicates a specific price path; Calculating option's payoff $f(\mathbf{X}_i)$ on each price path i and compute the expectation value of the payoff $E_P[f(X)]$ as an average across all paths given by:

$$\mathbb{E}_{\mathbb{P}}[f(\mathbf{X})] = \sum_{i=1}^{M} p_i f(\mathbf{X}_i)$$

where p_i is the probability of realizing the price path i; The final step is to discount the calculated expectation value to get the option's fair value.

4.1 Option Pricing Methodology for gate based Quantum Computers

Quantum Amplitude Estimation algorithm provides an advantage of pricing options on a quantum computer. For an option with payoff f, let's suppose the $\mathcal{A}$ operator creates the following state:

$$\sum_{i=0}^{2^n-1} \sqrt{1-f\left(S_i\right)}\sqrt{p_i}\left|S_i\right\rangle|0\rangle+\sum_{i=0}^{2^n-1} \sqrt{f\left(S_i\right)}\sqrt{p_i}\left|S_i\right\rangle|1\rangle.$$

where S_i represent the possible values the underlying asset can take and the corresponding probabilities are p_i. Here we have assumed that the prices can take 2^n possible values on expiration and on a quantum circuit n qubits can be used to represent all these price paths.

Comparing with 15, we find that

$$a=\sum_{i=0}^{2^n-1} f\left(S_i\right) p_i=\mathbb{E}[f(S)]$$

The value of a which can be estimated from the method described in section 3 will lead us to the expectation value of the payoff.

These are the following critical steps involved:

1. Load the probability distribution $\mathbb{P}$ of the random variables $\mathbf{X}$
   The price of a path-independent option depends only on the distribution of the underlying asset price S_t at the option maturity t and the payoff function f of the option. The distribution of S_t is truncated to the range $[S_{t,\min}, S_{t,\max}]$ and this interval is discretized to $\{0,\ldots,2^n-1\}$ to encode 2^n potential price values the asset can take upon expiration, utilizing n qubits. The distribution loading operator $\mathcal{P}$ creates a state

$$|0\rangle_n \xrightarrow{\mathcal{P}} |\psi\rangle_n=\sum_{i=0}^{2^n-1} \sqrt{p_i}|i\rangle_n \tag{18}$$

   with $i\in\{0,\ldots,2^n-1\}$ to represent S_t. The n-qubit state $|i\rangle_n=|i_{n-1}\ldots i_0\rangle$ encodes the integer $i=2^{n-1}i_{n-1}+\ldots+2i_1+i_0\in\{0,\ldots,2^n-1\}$ with $i_k\in\{0,1\}$ and $k=0,\ldots,n-1$. We note that the outcomes of a random variable X can be mapped to the integer set $\{0,\ldots,2^n-1\}$ using an affine mapping.

2. Construct a quantum circuit which computes the payoff $f(\mathbf{X})$ An example for a call option payoff is the European Call Option $f\left(S_t\right)$ given as

$$f\left(S_t\right)=\max\left(0, S_t-K\right)$$

To load this pay-off function into the quantum circuit and compute the expectation value of the pay-off, the state in 18 along with an ancilla qubit the following procedure as described in **reference** is followed:

The payoff function of European call-option is piece-wise linear and hence for $f : \{0, \dots, 2^n - 1\} \to [0, 1]$ we write $f(i) = f_1 i + f_0$. We create a circuit that performs

$$|i\rangle_n|0\rangle \to |i\rangle_n(\cos[f(i)]|0\rangle + \sin[f(i)]|1\rangle)$$

for all i using controlled Y-rotations implemented with CNOT and single-qubit gates.

Using this technique, the following transformation can be achieved on the state $|\psi\rangle_n|0\rangle$

$$\sum_i \sqrt{p_i}|i\rangle_n|0\rangle \to \sum_{i=0}^{2^n-1} \sqrt{p_i}|i\rangle_n \left[\cos\left(c\tilde{f}(i) + \frac{\pi}{4}\right)|0\rangle + \sin\left(c\tilde{f}(i) + \frac{\pi}{4}\right)|1\rangle\right]$$

Here $\tilde{f}(i)$ is a scaled version of $f(i)$ given by

$$\tilde{f}(i) = 2\frac{f(i) - f_{\min}}{f_{\max} - f_{\min}} - 1$$

with $f_{\min} = \min_i f(i)$ and $f_{\max} = \max_i f(i)$, and $c \in [0, 1]$ is an additional scaling parameter. The relation is chosen so that $\tilde{f}(i) \in [-1, 1]$.

3. Calculate the expectation value of the payoff $\mathbb{E}_{\mathbb{P}}[f(\mathbf{X})]$.

The probability to find the ancilla qubit in state $|1\rangle$, is given by

$$P_1 = \sum_{i=0}^{2^n-1} p_i \sin^2\left(c\tilde{f}(i) + \frac{\pi}{4}\right)$$

Using the approximation $\sin^2\left(c\tilde{f}(i) + \frac{\pi}{4}\right) \approx c\tilde{f}(i) + \frac{1}{2}$, leads to
$P_1 \approx \sum_{i=0}^{2^n-1} p_i \left(c\tilde{f}(i) + \frac{1}{2}\right) = c\frac{2\mathbb{E}[f(X)] - f_{\min}}{f_{\max} - f_{\min}} - c + \frac{1}{2}$.
Thus we could recover $\mathbb{E}[\max(0, i - K)]$ from P_1 up to a scaling factor and a constant.