ELEN 162
A. I. Zecevic

Lecture Notes for Week 5

Eigenvalue Estimation

On first glance, estimating the eigenvalues of a given operator seems to be a straightforward task, which can be handled by any classical computer with sufficient resources. It turns out, however, that this is an important problem in quantum computing, because it provides the conceptual framework for algorithms that perform prime factorization. Since our ultimate goal is to understand how these algorithms work, we will need to examine the eigenvalue estimation problem in some detail.

To explain what this problem entails, suppose that we are given a *unitary* operator $\hat{U}$ and that ξ_j is one of its eigenfunctions. We will initially assume that ξ_j is known, and will later show how the algorithm should be modifed when this is not the case.

Since $\hat{U}$ is unitary, we know that the corresponding eigenvalue λ_j will have the form $\lambda_j = e^{i\phi_j}$. Setting $\omega = \phi_j/2\pi$, we can rewrite λ_j as $\lambda_j = e^{2\pi i\omega}$ (which will be a bit more convenent to work with). Given this change of variables, our objective in the following will be to design a quantum circuit that can compute ω as accurately as possible. As we do so, we will have to distinguish between two different scenarios - one in which ω can be computed *precisley*, and another in which it can only be *approximated*.

Scenario 1: Exact Computation of ω

When the number of bits in ω matches the number of available qubits, it is possible to compute the eigenvalue *exactly*. To see how this can be done, suppose that ω has n bits in its binary expansion, and that we have n qubits at our disposal. In that case, the binary representation of ω will have the general form

$$\omega = q + x_1 2^{-1} + x_2 2^{-2} + \ldots + x_n 2^{-n} \tag{1}$$

where q is an integer, and each x_k is either zero or 1.

For our purposes, it will be helpful to rewrite ω as

$$\omega = q + \frac{x}{2^n} \tag{2}$$

where

$$x = x_1 2^{n-1} + x_2 2^{n-2} + \ldots + x_n 2^0 \tag{3}$$

is an integer. When using expression (2) to compute λ_j, it is important to keep in mind that q is actually irrelevant, since

$$\lambda_j = e^{2\pi i\omega} = e^{2\pi iq} \cdot e^{(2\pi ix)/2^n} \tag{4}$$

and $e^{2\pi iq} = 1$ for any integer q. As a result, in the following we will set q to zero, and assume (without any loss of generality) that $\omega = x/2^n$.

The problem that we will be solving can now be stated as follows:

Problem 1. Suppose that we can place a system of $n+1$ qubits into state

$$\Psi_{\text{in}} = \psi_0 \otimes \psi_0 \otimes \ldots \otimes \psi_0 \otimes \xi_j \tag{5}$$

Our task will be to build a quantum circuit that takes Ψ_{in} as its input, and produces state

$$\Psi_{\text{out}} = \Psi_x \otimes \xi_j \equiv \psi_{x_1} \otimes \psi_{x_2} \otimes \ldots \otimes \psi_{x_n} \otimes \xi_j \tag{6}$$

where $\{x_1, x_2, \ldots, x_n\}$ represent the bits that appear in the binary expansion of ω.

Before we explain how this problem can be solved, we first need to introduce the concept of a quantum Fourier transform, and say a few words about its properties.

Quantum Fourier Transforms

Given an orthonormal basis $\{\Psi_0, \Psi_1, \ldots, \Psi_{N-1}\}$ (where $N = 2^n$), the *quantum Fourier transform* is defined as an operator that maps a basis state Ψ_x into

$$\hat{Q}(\Psi_x) = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{2\pi i \frac{x}{N} y} \Psi_y \tag{7}$$

The reason why we refer to expression (7) as a quantum Fourier transform lies in its similarity with the normalized discrete Fourier transform, which is computed as

$$x_n = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i \frac{n}{N} k} \, u_k \tag{8}$$

for a given sequence $\{u_0, u_1, \ldots, u_{N-1}\}$. It is easily verified that this formula can be obtained from (7) by replacing Ψ with u, y with k and x with n.

In order to solve the eigenvalue estimation problem, we will also make use of the *inverse quantum Fourier transform*, which is defined as

$$\hat{Q}^{-1}(\Psi_y) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{-2\pi i \frac{y}{N} k} \Psi_k \tag{9}$$

It is not difficult to show that this definition is consistent with the one in (7), since the operator described in (9) satisfies

$$\hat{Q}^{-1}\left[\hat{Q}(\Psi_x)\right] = \Psi_x \tag{10}$$

(we won't do that here, but you can find the derivation in the textbook).

We will now describe how the quantum eigenvalue estimation algorithm works by breaking it down into two steps.

Step 1 - The First Quantum Circuit

In the first step, our objective will be to design a quantum circuit that takes

$$\Psi_{\text{in}} = \psi_0 \otimes \psi_0 \otimes \ldots \otimes \psi_0 \otimes \xi_j \tag{11}$$

as its input, and produces state

$$\Psi_{\text{out}} = F(\omega) \otimes \xi_j \tag{12}$$

at the output, where

$$F(\omega) = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{2\pi i \omega y} \Psi_y \tag{13}$$

In order to build such a circuit, it will be useful to introduce a pair of operators $\hat{A}_0$ and $\hat{A}_1$, which are defined as

$$\begin{aligned} \hat{A}_0 \psi &= \langle \psi_0, \psi \rangle \, \psi_0 \\ \hat{A}_1 \psi &= \langle \psi_1, \psi \rangle \, \psi_1 \end{aligned} \tag{14}$$

It is not difficult to see that these operators satisfy

$$\begin{aligned} \hat{A}_0 \psi_0 &= \langle \psi_0, \psi_0 \rangle \, \psi_0 = \psi_0 \\ \hat{A}_0 \psi_1 &= \langle \psi_0, \psi_1 \rangle \, \psi_0 = 0 \\ \hat{A}_1 \psi_0 &= \langle \psi_1, \psi_0 \rangle \, \psi_1 = 0 \\ \hat{A}_1 \psi_1 &= \langle \psi_1, \psi_1 \rangle \, \psi_1 = \psi_1 \end{aligned} \tag{15}$$

since functions ψ_0 are ψ_1 orthonormal.

The four identities described in (15) allow us to express $\hat{A}_0 \psi_+$ and $\hat{A}_1 \psi_+$ as

$$\begin{aligned} \hat{A}_0 \psi_+ &= \frac{1}{\sqrt{2}} \hat{A}_0 (\psi_0 + \psi_1) = \frac{1}{\sqrt{2}} (\hat{A}_0 \psi_0 + \hat{A}_0 \psi_1) = \frac{1}{\sqrt{2}} \psi_0 \\ \hat{A}_1 \psi_+ &= \frac{1}{\sqrt{2}} \hat{A}_1 (\psi_0 + \psi_1) = \frac{1}{\sqrt{2}} (\hat{A}_1 \psi_0 + \hat{A}_1 \psi_1) = \frac{1}{\sqrt{2}} \psi_1 \end{aligned} \tag{16}$$

To explain why the last two transformations are relevant, let us assume that we have a pair of particles that are in state ψ_+ and a third particle which is in state ξ_j. The overall wave function of such a system can then be represented as

$$\Psi = \psi_+ \otimes \psi_+ \otimes \xi_j \tag{17}$$

If we define operator $\hat{W}_0$ as

$$\hat{W}_0 = \hat{I} \otimes \hat{A}_0 \otimes \hat{I} + \hat{I} \otimes \hat{A}_1 \otimes \hat{U} \tag{18}$$

and apply it to function Ψ, we obtain

$$\begin{aligned} \hat{W}_0 \Psi &= \psi_+ \otimes \left(\hat{A}_0 \psi_+ \right) \otimes \xi_j + \psi_+ \otimes (\hat{A}_1 \psi_+) \otimes (\hat{U} \xi_j) = \\ &= \psi_+ \otimes \frac{1}{\sqrt{2}} \psi_0 \otimes \xi_j + \psi_+ \otimes \frac{1}{\sqrt{2}} \psi_1 \otimes e^{2\pi i \omega} \xi_j = \\ &= \psi_+ \otimes \frac{1}{\sqrt{2}} \psi_0 \otimes \xi_j + \psi_+ \otimes \frac{1}{\sqrt{2}} e^{2\pi i \omega} \psi_1 \otimes \xi_j = \\ &= \psi_+ \otimes \varphi_0 \otimes \xi_j \end{aligned} \tag{19}$$

where

$$\varphi_0 = \frac{1}{\sqrt{2}}(\psi_0 + e^{2\pi i\omega}\psi_1) \tag{20}$$

A second operator of the form

$$\hat{W}_1 = \hat{A}_0 \otimes \hat{I} \otimes \hat{I} + \hat{A}_1 \otimes \hat{I} \otimes \hat{U}^2 \tag{21}$$

will then transform function $\hat{W}_0\Psi$ into

$$\begin{aligned}
&\hat{W}_1\hat{W}_0\Psi = \hat{W}_1(\psi_+ \otimes \varphi_0 \otimes \xi_j) = \left(\hat{A}_0\psi_+\right) \otimes \varphi_0 \otimes \xi_j + \\
&+(\hat{A}_1\psi_+) \otimes \varphi_0 \otimes (\hat{U}^2\xi_j) = \frac{1}{\sqrt{2}}\psi_0 \otimes \varphi_0 \otimes \xi_j + \frac{1}{\sqrt{2}}\psi_1 \otimes \varphi_0 \otimes e^{2\pi i\omega\cdot 2}\xi_j = \\
&= \frac{1}{\sqrt{2}}\psi_0 \otimes \varphi_0 \otimes \xi_j + \frac{1}{\sqrt{2}}e^{2\pi i\omega\cdot 2}\psi_1 \otimes \varphi_0 \otimes \xi_j = \\
&= \varphi_1 \otimes \varphi_0 \otimes \xi_j
\end{aligned} \tag{22}$$

where

$$\varphi_1 = \frac{1}{\sqrt{2}}(\psi_0 + e^{2\pi i\omega\cdot 2}\psi_1) \tag{23}$$

Remark 1. Note that the rules for tensor products allow us to move the terms $e^{2\pi i\omega}$ and $e^{2\pi i\omega\cdot 2}$ next to ψ_1. Doing so results in more compact expressions, while leaving functions $\hat{W}_0\Psi$ and $\hat{W}_1\hat{W}_0\Psi$ unchanged.

Recalling that $\psi_+ = \hat{H}\psi_0$ (where $\hat{H}$ is the Hadamard operator), the procedure that we just described can be schematically represented in the manner shown in Fig. 1. Note that both φ_1 and φ_2 depend on ω, so we will refer to them as $\varphi_1(\omega)$ and $\varphi_2(\omega)$ whenever it is important to emphasize this fact.

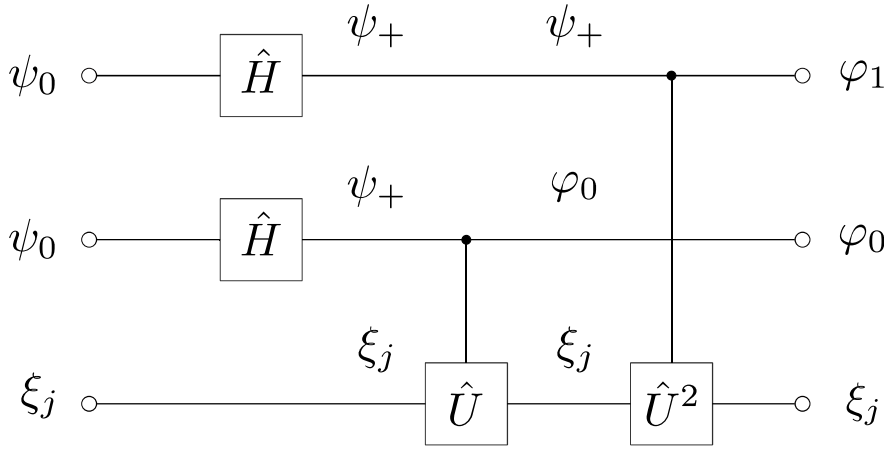

Figure 1: A quantum circuit that transforms $\psi_0 \otimes \psi_0$ into $\varphi_1 \otimes \varphi_0$.

When interpreting the diagram in Fig. 1, we should bear in mind that it represents *a two step process*. In the first step (which corresponds to operator $\hat{W}_0$), one of the particles that was in state ψ_+ is transformed into state $\varphi_0(\omega)$, and in the second one (which corresponds to $\hat{W}_1$), the state of the other particle changes from ψ_+ to $\varphi_1(\omega)$. The nodes in this

figure represent points in the process where these transformations occur. Since they involve applying different powers of operator $\hat{U}$ to function ξ_j, each node is connected to the block that is active in that step

This diagram can be easily generalized to the case when we have n qubits in state ψ_+, and the powers of operator $\hat{U}$ range from 1 to 2^{n-1}. Such a configuration is illustrated in Fig. 2, in which functions $\varphi_k(\omega)$ have the form

$$\varphi_k(\omega) = \frac{1}{\sqrt{2}}(\psi_0 + e^{2\pi i \cdot 2^k \omega}\psi_1) \quad (k = 0, 1, \ldots, n-1) \tag{24}$$

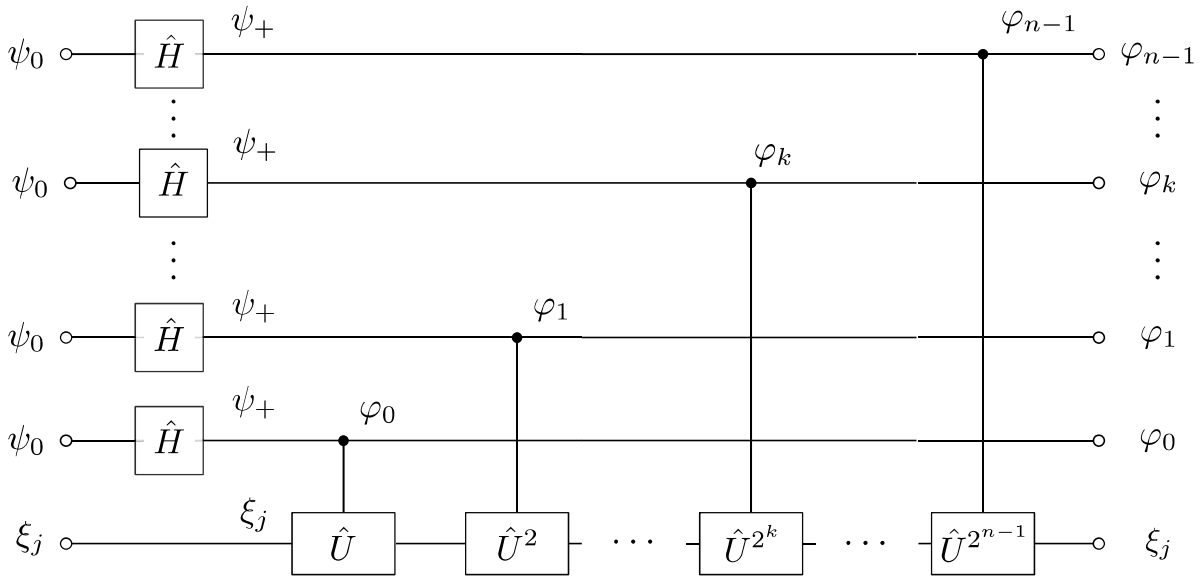

Figure 2: A generalization of the circuit in Fig. 1.

As in Fig. 1, the nodes in this diagram correspond to points in the process where a particle is transformed from state ψ_+ into one of the states $\varphi_k(\omega)$ $(k = 0, 1, \ldots, n-1)$. Since such transformations involve applying different powers of operator $\hat{U}$ to function ξ_j, each node is connected to the block $\hat{U}^{2^k}$ that is active in that step.

Remark 2. That fact that we need high powers of $\hat{U}$ when n is large poses a practical problem which we will address later. For now, it suffices to say that it can be solved in an elegant way.

The quantum circuit in Fig. 2 is useful because it turns out that its output

$$\Psi_{\text{out}} = \varphi_{n-1}(\omega) \otimes \varphi_{n-2}(\omega) \otimes \ldots \otimes \varphi_1(\omega) \otimes \varphi_0(\omega) \tag{25}$$

can be equivalently represented as

$$\Psi_{\text{out}} = F(\omega) = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{2\pi i \omega y} \Psi_y \tag{26}$$

where $F(\omega)$ is the function defined in (13) (a derivation of this property is provided in the textbook). This sets the stage for the second step of the algorithm, in which we will design a quantum circuit that takes (25) as its input, and produces function

$$\Psi_x = \psi_{x_1} \otimes \psi_{x_2} \otimes \ldots \otimes \psi_{x_n} \tag{27}$$

at the output (where $\{x_1, x_2, \ldots, x_n\}$ are the bits that appear in the binary expansion of ω).

Step 2 - The Second Quantum Circuit

It is not difficult to see that $F(\omega)$ represents the quantum Fourier transform of Ψ_x, since

$$F(\omega) = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{2\pi i \omega y} \Psi_y = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{2\pi i \frac{x}{2^n} y} \Psi_y \tag{28}$$

when

$$\omega = \frac{x}{2^n} \tag{29}$$

As a result, we can retrieve x by simply applying operator $\hat{Q}^{-1}$ to $F(\omega)$.

To see how the inverse Fourier transform can be implemented using a quantum circuit, we should first observe that functions φ_k $(k = 0, 1, \ldots, n-1)$ can be equivalently expressed as

$$\varphi_k(\omega) = \frac{1}{\sqrt{2}} \left[\psi_0 + e^{2\pi i (2^k \omega)} \psi_1\right] = \frac{1}{\sqrt{2}} \left[\psi_0 + e^{2\pi i \alpha_k(x)} \psi_1\right] \tag{30}$$

where

$$\alpha_k(x) = x_{k+1} 2^{-1} + x_{k+2} 2^{-2} \ldots + x_n 2^{k-n} \tag{31}$$

(this is shown in the textbook). That allows us to represent function

$$F(\omega) = \varphi_{n-1}(\omega) \otimes \varphi_{n-2}(\omega) \otimes \ldots \otimes \varphi_1(\omega) \otimes \varphi_0(\omega) \tag{32}$$

as

$$\begin{aligned} F(\omega) = {} & \frac{1}{\sqrt{2}} \left[\psi_0 + e^{2\pi i \alpha_{n-1}(x)} \psi_1\right] \otimes \frac{1}{\sqrt{2}} \left[\psi_0 + e^{2\pi i \alpha_{n-2}(x)} \psi_1\right] \otimes \ldots \\ & \ldots \otimes \frac{1}{\sqrt{2}} \left[\psi_0 + e^{2\pi i \alpha_0(x)} \psi_1\right] \end{aligned} \tag{33}$$

Remark 3. Note that F becomes a function of x after replacing $2^k\omega$ with $\alpha_k(x)$. We will continue to refer to it as $F(\omega)$, however, since ω and x are related in the manner shown in (29) (and are therefore interchangeable).

To gain some additional insight into the structure of function $F(\omega)$, it is helpful to show what coefficients α_k look like for different values of k. For the sake of simplicity, we will only consider the case when $k \in \{0, 1, 2\}$.

CASE 1. If $k = 0$, the first term in

$$\alpha_k(x) = x_{k+1} 2^{-1} + x_{k+2} 2^{-2} \ldots + x_n 2^{k-n} \tag{34}$$

is $x_1 2^{-1}$ and the last one is $x_n 2^{-n}$. Consequently, we have that

$$\alpha_0(x) = x_1 2^{-1} + x_2 2^{-2} + \ldots + x_n 2^{-n} \tag{35}$$

CASE 2. If $k = 1$, the first term in

$$\alpha_k(x) = x_{k+1} 2^{-1} + x_{k+2} 2^{-2} \ldots + x_n 2^{k-n} \tag{36}$$

is $x_2 2^{-1}$ and the last one is $x_n 2^{1-n}$. This implies that

$$\alpha_1(x) = x_2 2^{-1} + x_3 2^{-2} + \ldots + x_n 2^{1-n} \tag{37}$$

CASE 3. If $k = 2$, the first term in

$$\alpha_k(x) = x_{k+1} 2^{-1} + x_{k+2} 2^{-2} \ldots + x_n 2^{k-n} \tag{38}$$

is $x_3 2^{-1}$ and the last one is $x_n 2^{2-n}$. As a result, α_2 will have the form

$$\alpha_2(x) = x_3 2^{-1} + x_4 2^{-2} + \ldots + x_n 2^{2-n} \tag{39}$$

We will now look at two examples which illustrate how $\{x_1, x_2, \ldots, x_n\}$ can be extracted from function

$$F(\omega) = \varphi_{n-1}(\omega) \otimes \varphi_{n-2}(\omega) \otimes \ldots \otimes \varphi_1(\omega) \otimes \varphi_0(\omega) \tag{40}$$

Example 1. Let us consider the simplest possible scenario, in which ω has only 1 bit in its binary expansion, and we have a single qubit at our disposal. This implies that $n = 1$, and that ω has the form

$$\omega = x_1 2^{-1} \tag{41}$$

If we substitute $n = 1$ into (40), function

$$F(\omega) = \varphi_{n-1}(\omega) \otimes \varphi_{n-2}(\omega) \otimes \ldots \otimes \varphi_1(\omega) \otimes \varphi_0(\omega) \tag{42}$$

reduces to

$$F(\omega) = \varphi_0(\omega) \tag{43}$$

Given that

$$\varphi_0(\omega) = \frac{1}{\sqrt{2}} \left[\psi_0 + e^{2\pi i \alpha_0(x)} \psi_1\right] \tag{44}$$

it now follows that we only need to determine $\alpha_0(x)$.

When $n = 1$, the last term in

$$\alpha_0(x) = x_1 2^{-1} + x_2 2^{-2} + \ldots + x_n 2^{-n} \tag{45}$$

is $x_1 2^{-1}$. Since the first and last terms are identical in this case, $\alpha_0(x)$ becomes

$$\alpha_0(x) = x_1 2^{-1} \tag{46}$$

and $F(\omega)$ can be represented as

$$\begin{aligned} F(\omega) &= \frac{1}{\sqrt{2}}\left[\psi_0 + e^{2\pi i \alpha_0(x)}\psi_1\right] = \frac{1}{\sqrt{2}}\left[\psi_0 + e^{2\pi i x_1 2^{-1}}\psi_1\right] = \\ &= \frac{1}{\sqrt{2}}\left[\psi_0 + e^{\pi i x_1}\psi_1\right] \end{aligned} \tag{47}$$

Observing that

$$e^{\pi i x_1} = (e^{i\pi})^{x_1} = (-1)^{x_1} \tag{48}$$

(47) can be rewritten as

$$F(\omega) = \frac{1}{\sqrt{2}}\left[\psi_0 + (-1)^{x_1}\psi_1\right] \tag{49}$$

Since this expression matches the output of a Hadamard gate with input ψ_{x_1} (see lecture notes for week 4), it follows that $F(\omega) = \hat{H}\psi_{x_1}$, and that

$$\hat{H}[F(\omega)] = \hat{H}^2\psi_{x_1} = \psi_{x_1} \tag{50}$$

We can therefore conclude that x_1 can be recovered by simply passing $F(\omega)$ through a Hadamard gate.

Example 2. Suppose ω has 2 bits in its binary expansion, and that we have 2 qubits at our disposal. In that case $n = 2$, and can express ω as

$$\omega = x_1 2^{-1} + x_2 2^{-2} \tag{51}$$

Setting $n = 2$ in expression

$$F(\omega) = \varphi_{n-1}(\omega) \otimes \varphi_{n-2}(\omega) \otimes \ldots \otimes \varphi_1(\omega) \otimes \varphi_0(\omega) \tag{52}$$

reduces $F(\omega)$ to

$$F(\omega) = \varphi_1(\omega) \otimes \varphi_0(\omega) = \frac{1}{\sqrt{2}}\left[\psi_0 + e^{2\pi i \alpha_1(x)}\psi_1\right] \otimes \frac{1}{\sqrt{2}}\left[\psi_0 + e^{2\pi i \alpha_0(x)}\psi_1\right] \tag{53}$$

which means that we need to compute $\alpha_0(x)$ and $\alpha_1(x)$.

When $n = 2$, the last term in

$$\alpha_0(x) = x_1 2^{-1} + x_2 2^{-2} + \ldots + x_n 2^{-n} \tag{54}$$

is $x_2 2^{-2}$, so $\alpha_0(x)$ becomes

$$\alpha_0(x) = x_1 2^{-1} + x_2 2^{-2} \tag{55}$$

If we repeat this analysis for α_1, we will see that the last term in

$$\alpha_1(x) = x_2 2^{-1} + x_3 2^{-2} + \ldots + x_n 2^{1-n} \tag{56}$$

is now $x_2 2^{-1}$, which matches the first term. As a result, $\alpha_1(x)$ will have the form

$$\alpha_1(x) = x_2 2^{-1} \tag{57}$$

Using expressions (55) and (57), we can rewrite $\varphi_1(x)$ and $\varphi_0(x)$ as

$$\varphi_1(x) = \frac{1}{\sqrt{2}}\left[\psi_0 + e^{2\pi i\alpha_1(x)}\psi_1\right] = \frac{1}{\sqrt{2}}\left[\psi_0 + e^{\pi i x_2}\psi_1\right] = \frac{1}{\sqrt{2}}\left[\psi_0 + (-1)^{x_2}\psi_1\right] = \hat{H}\psi_{x_2} \tag{58}$$

and

$$\varphi_0(x) = \frac{1}{\sqrt{2}}\left[\psi_0 + e^{2\pi i\alpha_0(x)}\psi_1\right] = \frac{1}{\sqrt{2}}\left[\psi_0 + e^{2\pi i(x_1 2^{-1} + x_2 2^{-2})}\psi_1\right] \tag{59}$$

respectively. We can now determine x_1 and x_2 in two steps.

STEP 1. Our first step will be to recover x_2 from $\varphi_1(x)$. This is actually quite straightforward, because it only requires passing this function through a Hadamard gate (given that $\hat{H}\varphi_1 = \psi_{x_2}$).

STEP 2. Recovering x_1 from $\varphi_0(x)$ is considerably more complicated, since the expression for $\varphi_0(x)$ includes both x_1 and x_2. In order to address this problem, we should first observe that when $x_2 = 0$, function

$$\varphi_0(x) = \frac{1}{\sqrt{2}}\left[\psi_0 + e^{2\pi i(x_1 2^{-1} + x_2 2^{-2})}\psi_1\right] \tag{60}$$

simplifies to

$$\begin{aligned}\varphi_0(x) &= \frac{1}{\sqrt{2}}\left[\psi_0 + e^{2\pi i x_1 2^{-1}}\psi_1\right] = \frac{1}{\sqrt{2}}\left[\psi_0 + e^{\pi i x_1}\psi_1\right] = \\ &= \frac{1}{\sqrt{2}}\left[\psi_0 + (-1)^{x_1}\psi_1\right] = \hat{H}\psi_{x_1}\end{aligned} \tag{61}$$

Since x_1 can be easily determined in this case (using a Hadamard gate), it follows that we will have to perform additional operations only when $x_2 = 1$.

To see what that entails, let us introduce an operator $\hat{R}_k$ which transforms basis functions ψ_0 and ψ_1 as

$$\begin{aligned}\hat{R}_k\psi_0 &= \psi_0 \\ \hat{R}_k\psi_1 &= e^{2\pi i 2^{-k}}\psi_1\end{aligned} \tag{62}$$

It is not difficult to see that the inverse of this operator satisfies

$$\begin{aligned}\hat{R}_k^{-1}\psi_0 &= \psi_0 \\ \hat{R}_k^{-1}\psi_1 &= e^{-2\pi i 2^{-k}}\psi_1\end{aligned} \tag{63}$$

which implies that

$$\begin{aligned}\hat{R}_2^{-1}\varphi_0(x) &= \frac{1}{\sqrt{2}}\left[\hat{R}_2^{-1}\psi_0 + e^{2\pi i(x_1 2^{-1} + x_2 2^{-2})}\hat{R}_2^{-1}\psi_1\right] = \\ &= \frac{1}{\sqrt{2}}\left[\psi_0 + e^{2\pi i(x_1 2^{-1} + x_2 2^{-2})}\cdot e^{-2\pi i 2^{-2}}\psi_1\right]\end{aligned} \tag{64}$$

From expression (64), it is obvious that applying operator $\hat{R}_2^{-1}$ to φ_0 allows us to eliminate

the term $x_2 \cdot 2^{-2}$ when $x_2 = 1$, since

$$\begin{aligned}
&\hat{R}_2^{-1}\varphi_0 = \frac{1}{\sqrt{2}}\left[\psi_0 + e^{2\pi i(x_1 2^{-1}+2^{-2})} \cdot e^{-2\pi i 2^{-2}}\psi_1\right] = \\
&= \frac{1}{\sqrt{2}}\left[\psi_0 + e^{2\pi i(x_1 2^{-1}+2^{-2}-2^{-2})} \cdot \psi_1\right] = \frac{1}{\sqrt{2}}\left[\psi_0 + e^{2\pi i x_1 2^{-1}}\psi_1\right] = \\
&= \frac{1}{\sqrt{2}}\left[\psi_0 + e^{\pi i x_1}\psi_1\right] = \frac{1}{\sqrt{2}}\left[\psi_0 + (-1)^{x_1}\psi_1\right]
\end{aligned} \tag{65}$$

As a result, we can extract x_1 by passing function $\hat{R}_2^{-1}\varphi_0$ through a Hadamard gate.

The analysis that we just performed indicates that we need to apply $\hat{R}_2^{-1}$ to φ_0 when $x_2 = 1$, but not when $x_2 = 0$. The quantum circuit in Fig. 3 shows how this can be done. The black node in this diagram indicates that the quantum gate associated with operator $\hat{R}_2^{-1}$ acts *selectively*, and *activates only when* $x_2 = 1$.

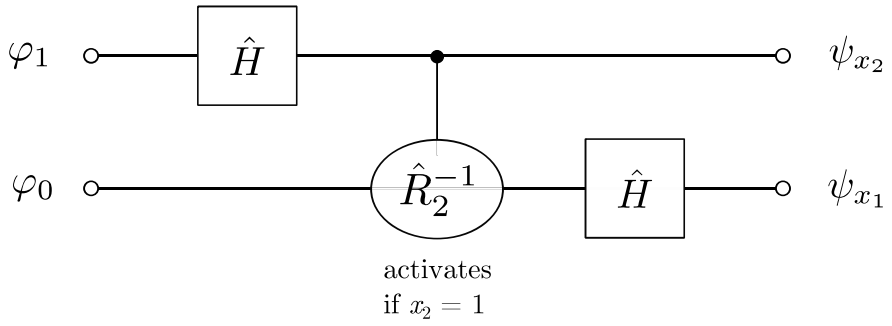

Figure 3: A circuit that transforms $\varphi_1 \otimes \varphi_0$ into $\psi_{x_1} \otimes \psi_{x_2}$.

Remark 4. The textbook describes an operator that corresponds to such a "conditional" transformation. We won't get into the details here, but it suffices to say that it can be constructed using operators $\hat{A}_0$, $\hat{A}_1$ and $\hat{R}_2^{-1}$.

Remark 5. The circuit in Fig. 3 allows us to map function $F(\omega) = \varphi_1(x) \otimes \varphi_0(x)$ into $\Psi_x = \psi_{x_1} \otimes \psi_{x_2}$. Recalling that $F(\omega)$ represents the quantum Fourier transform of Ψ_x, it follows that *this circuit performs the inverse quantum Fourier transform.*

The quantum circuit described in Fig. 3 can be easily generalized to the case when we have n qubits. The diagram in Fig. 4 shows what it looks like when $n = 3$ (the textbook analyzes this scenario in much more detail). This diagram indicates that the bits of ω are once again recovered in stages, starting from x_3 and ending with x_1.

What can we conclude from all this? Examples 1 and 2 (as well as the circuit shown in Fig. 4) indicate that if ω has n bits in its binary expansion, we can compute it precisely using a pair of n qubit quantum circuits. The question that we will adress next is what to do when the number of available qubits is *smaller* than the number of bits in ω. This can occur if ω is a real number (and therefore has infinitely many bits), or if it is a rational number whose binary expansion contains more than n bits.

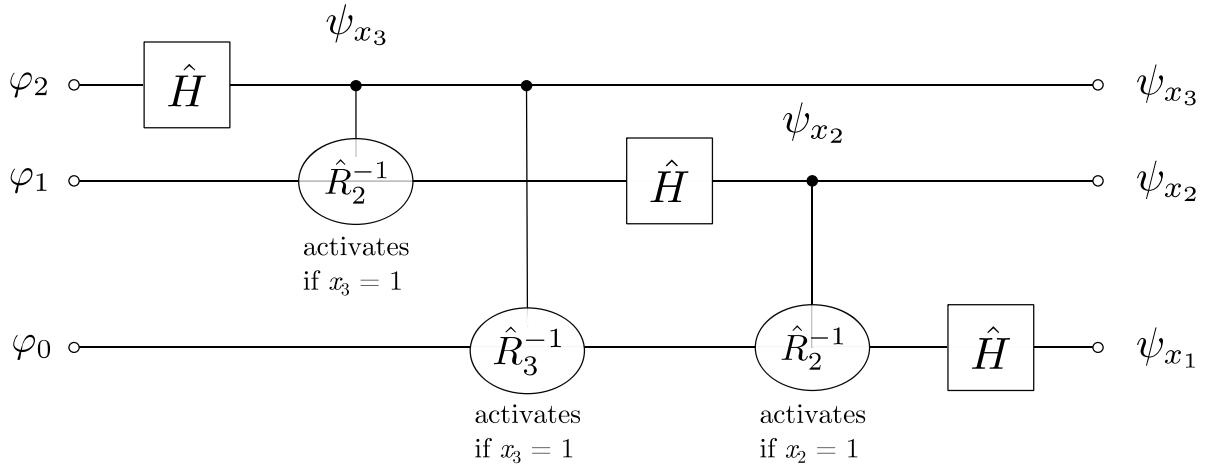

Figure 4: A quantum circuit that produces $\psi_{x_1} \otimes \psi_{x_2} \otimes \psi_{x_3}$.

Scenario 2: Approximating ω with a Preassigned Precision

Suppose that we have n qubits at our disposal, and that the number of bits in the binary expansion of ω is *larger* than n. In that case, ω will have the general form

$$\omega = q + x_1 2^{-1} + x_2 2^{-2} + \ldots + x_n 2^{-n} + \delta \tag{66}$$

where

$$\delta = \sum_{j=n+1}^{\infty} x_j 2^{-j} \tag{67}$$

and at least one of the terms in the infinite sum is nonzero.

Since our quantum computer is limited to n qubits by assumption, in this case we will *not* be able to determine ω precisely, and our goal will be to produce an approximation that is as accurate as possible. To see how this can be done, we should first recall that the integer part of ω has no effect on eigenvalue $\lambda = e^{2\pi i\omega}$. This allows us to disregard q in (66), and assume that

$$\omega = x_1 2^{-1} + x_2 2^{-2} + \ldots + x_n 2^{-n} + \sum_{j=n+1}^{\infty} x_j 2^{-j} \tag{68}$$

with no loss of generality. If we now define integer x as

$$x = x_1 2^{n-1} + x_2 2^{n-2} + \ldots + x_n 2^0 \tag{69}$$

it is obvious that

$$\frac{x}{2^n} = x_1 2^{-1} + x_2 2^{-2} + \ldots + x_n 2^{-n} \tag{70}$$

represents the *best possible* n bit approximation of ω (because it matches the first n bits of ω *exactly*).

Following the ideas that we developed previously, we will apply the same quantum circuit as before to input function

$$\Psi_{\text{in}} = \psi_0 \otimes \psi_0 \otimes \ldots \otimes \psi_0 \otimes \xi_j \tag{71}$$

When we do so, we will once again obtain

$$\Psi_{\text{out}} = F(\omega) \otimes \xi_j \tag{72}$$

where

$$F(\omega) = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{2\pi i \omega y} \Psi_y \tag{73}$$

What is different in this case is that $F(\omega)$ is *not* the quantum Fourier transform of state Ψ_x any more, because

$$\hat{Q}(\Psi_x) = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{2\pi i \frac{x}{2^n} y} \Psi_y \tag{74}$$

and $\omega \neq x/2^n$. As a result, we will not be able to recover x by simply passing $F(\omega)$ through the second circuit (which would be equivalent to applying the inverse Fourirer transform to $F(\omega)$).

What will we obtain if we do so anyway? To see this, we should first note that applying operator $\hat{Q}^{-1}$ to $F(\omega)$ produces

$$\Psi_{\text{out}} = \sum_{k=0}^{2^n-1} a_k(\omega) \Psi_k \tag{75}$$

where coefficients a_k are defined as

$$a_k(\omega) = \frac{1}{2^n} \sum_{m=0}^{2^n-1} e^{2\pi i m(\omega - \frac{k}{2^n})} \tag{76}$$

(a derivation of this expression is provided in the textbook). Note that these coefficients depend on ω, which means that the output function will depend on ω as well.

If we now perform a measurement on all n particles in the standard basis, we will obtain one of the functions from set $\{\Psi_0, \Psi_1, \ldots, \Psi_{2^n-1}\}$ (in the following, we will denote this function by Ψ_r). Ideally, we would like r to equal x, since $x/2^n$ represents the best possible n bit approximation of ω. This is much too restrictive, however, because requiring $r = x$ allows for only *one* acceptable state (out of 2^n possibilities). Since the probability of such an outcome is very low, we need to come up with an alternative strategy.

One possibility would be to expand the set of acceptable functions, and include any Ψ_r that satisfies

$$\left| \frac{x}{2^n} - \frac{r}{2^n} \right| < 2^{-\tau} \tag{77}$$

where τ is a positive integer. Doing so would obviously make a "favorable" outcome more likely, and would also ensure a certain level of similarity between functions Ψ_x and Ψ_r, since inequality (77) implies that $x/2^n$ and $r/2^n$ have the first τ bits in common. What this means is that we will obtain an approximation

$$\bar{\omega} = \frac{r}{2^n} = 0.r_1 r_2 \ldots r_n \tag{78}$$

which matches the first τ bits of ω.

There is no doubt, of course, that we would lose some accuracy by making such a compromise, since τ must be *smaller* than n (given that we have only n qubits at our disposal). We will see, however, that the price tag is not too high if n is sufficiently large.

To explain why this is so, we first need to evaluate the likelihood that we will actually register a state Ψ_r that satisfies condition (77). From expression (75), we know that the probability of observing any given state Ψ_k is

$$P(\Psi_k) = |a_k(\omega)|^2 \tag{79}$$

We will therefore need to add $|a_r(\omega)|^2$ over all r for which inequality (77) holds. This is not a trivial thing to do, but the final answer turns out to be quite simple.

To represent this answer in a form that is convenient from a computational standpoint, let us introduce a constant e which is defined as

$$e = 2^{n-\tau} - 1 \tag{80}$$

Condition (77) can then be rewritten as

$$|x - r| < 2^{n-\tau} = e + 1 \tag{81}$$

which is equivalent to

$$|x - r| \leq e \tag{82}$$

(since x, r and e are integers). Inequality (82) is useful because the probability that it will be satisfied has a lower bound that is easy to evaluate. This lower bound can be expressed as

$$\text{Prob}\left(|x - r| \leq e\right) > 1 - \frac{1}{2(e-1)} \tag{83}$$

and we can use it to determine the number of qubits that are needed to achieve a desired accuracy.

To see how this can be done, we should first recognize that expression (83) tells us that the probability of an "acceptable" outcome will exceed $1 - \varepsilon$ if ε and e are related as

$$\frac{1}{2(e-1)} \leq \varepsilon \tag{84}$$

Recalling that $e = 2^{n-\tau} - 1$ and setting $s = n - \tau$, condition (84) becomes

$$\frac{1}{2(2^s - 2)} \leq \varepsilon \tag{85}$$

which can be equivalently expressed as

$$\frac{1}{2\varepsilon} \leq 2^s - 2 \tag{86}$$

This inequality indicates that s should satisfy

$$s \geq \log_2(2 + \frac{1}{2\varepsilon}) \tag{87}$$

and that we need

$$n = \tau + s \geq \tau + \log_2(2 + \frac{1}{2\varepsilon}) \tag{88}$$

qubits in order to appoximate the first τ bits of ω with the desired probability.

The following example illustrates the practical value of this result.

Example 3. Suppose that we want our estimate to match the first 20 bits of ω, and that we would like the probability of obtaining such an approximation to exceed 99.9% when we perform a measurement. In that case, we should choose $\varepsilon = 0.001$ and n should satisfy

$$n \geq \tau + \log_2(2 + \frac{1}{2\varepsilon}) = 20 + 8.97 \tag{89}$$

Rounding off to the next integer, we can conclude that approximating ω with this precision requires at least 29 qubits.

Procedure when ξ_j is Unkown

We will close this section by pointing out a problem that could potentially limit the effectiveness of the algorithm that we just described. This problem has to do with the assumption that eigenfunction ξ_j (which corresponds to $\lambda_j = e^{2\pi i\omega}$) is known in advance, and that we can place a particle into this state whenever we need to. Based on these assumptions, we established that the circuits shown in Figs. 2 and 4 will produce function

$$\Psi_{\text{out}} = \Psi_r \otimes \xi_j = (\psi_{r_1} \otimes \psi_{r_2} \otimes \ldots \otimes \psi_{r_n}) \otimes \xi_j \tag{90}$$

which approximates ω as

$$\bar{\omega} = \frac{r}{2^n} = 0.r_1 r_2 \ldots r_n \tag{91}$$

We also showed that there is a high probability that this number will match the first τ bits of ω if the number of available qubits is sufficiently large.

The question that we now have to address is what happens when eigenfunction ξ_j is *not* available. Under such circumstances, it becomes necessary to determine whether ξ_j can be adequately replaced by some other function ξ that is easy to produce. To see if something like that is possible, we should first recall that $\hat{U}$ is a quantum operator, whose eigenfunctions $\{\xi_i\}$ constitute an orthonormal basis by definition (this is one of the postulates of quantum mechanics). As a result, we can express any function ξ in this space as

$$\xi = \sum_i \alpha_i \xi_i \tag{92}$$

If we choose one of these functions as our input (instead of ξ_j), it can be shown that the output state will have the form

$$\Psi_{\text{out}} = \sum_i \alpha_i [\Psi_{r_1^{(i)} r_2^{(i)} \ldots r_n^{(i)}} \otimes \xi_i] \tag{93}$$

The possible outcomes have probabilities $|\alpha_i|^2$ $(i = 1, 2, \ldots)$, and we know that one of them will be observed when we make a measurement. We don't know which one it will be, however - all that we can say is that if we register state

$$\Psi_{r_1^{(l)} r_2^{(l)} \ldots r_n^{(l)}} \otimes \xi_l \tag{94}$$

the resulting approximation

$$\bar{\omega}_l = 0.r_1^{(l)} r_2^{(l)} \dots r_n^{(l)} \tag{95}$$

is very likely to match the first τ bits of the eigenvalue that corresponds to ξ_l.

Such a scenario is obviously different from the one that we examined previously, but the algorithm that we described is still useful, because it guarantees that we will find *one* eigenvalue of $\hat{U}$ with the assigned precision. In subsequent lectures, we will see that this is sufficient for solving certain difficult problems (such as order finding and prime factorization, for example).