Exercise 5.3: (Classical fast Fourier transform) Suppose we wish to perform a Fourier transform of a vector containing 2^n complex numbers on a classical computer. Verify that the straightforward method for performing the Fourier transform, based upon direct evaluation of Equation (5.1) requires $\Theta(2^{2n})$ elementary arithmetic operations. Find a method for reducing this to $\Theta(n2^n)$ operations, based upon Equation (5.4).

Exercise 5.4: Give a decomposition of the controlled-R_k gate into single qubit and CNOT gates.

Exercise 5.5: Give a quantum circuit to perform the inverse quantum Fourier transform.

Exercise 5.6: (Approximate quantum Fourier transform) The quantum circuit construction of the quantum Fourier transform apparently requires gates of exponential precision in the number of qubits used. However, such precision is never required in any quantum circuit of polynomial size. For example, let U be the ideal quantum Fourier transform on n qubits, and V be the transform which results if the controlled-R_k gates are performed to a precision $\Delta = 1/p(n)$ for some polynomial $p(n)$. Show that the error $E(U, V) \equiv \max_{|\psi\rangle} \|(U - V)|\psi\rangle\|$ scales as $\Theta(n^2/p(n))$, and thus polynomial precision in each gate is sufficient to guarantee polynomial accuracy in the output state.

5.2 Phase estimation

The Fourier transform is the key to a general procedure known as *phase estimation*, which in turn is the key for many quantum algorithms. Suppose a unitary operator U has an eigenvector $|u\rangle$ with eigenvalue $e^{2\pi i\varphi}$, where the value of φ is unknown. The goal of the phase estimation algorithm is to estimate φ. To perform the estimation we assume that we have available *black boxes* (sometimes known as *oracles*) capable of preparing the state $|u\rangle$ and performing the controlled-U^{2^j} operation, for suitable non-negative integers j. The use of black boxes indicates that the phase estimation procedure is not a complete quantum algorithm in its own right. Rather, you should think of phase estimation as a kind of 'subroutine' or 'module' that, when combined with other subroutines, can be used to perform interesting computational tasks. In specific applications of the phase estimation procedure we shall do exactly this, describing how these black box operations are to be performed, and combining them with the phase estimation procedure to do genuinely useful tasks. For the moment, though, we will continue to imagine them as black boxes.

The quantum phase estimation procedure uses two registers. The first register contains t qubits initially in the state $|0\rangle$. How we choose t depends on two things: the number of digits of accuracy we wish to have in our estimate for φ, and with what probability we wish the phase estimation procedure to be successful. The dependence of t on these quantities emerges naturally from the following analysis.

The second register begins in the state $|u\rangle$, and contains as many qubits as is necessary to store $|u\rangle$. Phase estimation is performed in two stages. First, we apply the circuit shown in Figure 5.2. The circuit begins by applying a Hadamard transform to the first register, followed by application of controlled-U operations on the second register, with U raised

to successive powers of two. The final state of the first register is easily seen to be:

$$\frac{1}{2^{t/2}}\left(|0\rangle + e^{2\pi i 2^{t-1}\varphi}|1\rangle\right)\left(|0\rangle + e^{2\pi i 2^{t-2}\varphi}|1\rangle\right)\ldots\left(|0\rangle + e^{2\pi i 2^{0}\varphi}|1\rangle\right) = \frac{1}{2^{t/2}}\sum_{k=0}^{2^t-1} e^{2\pi i\varphi k}|k\rangle\,. \tag{5.20}$$

We omit the second register from this description, since it stays in the state $|u\rangle$ throughout the computation.

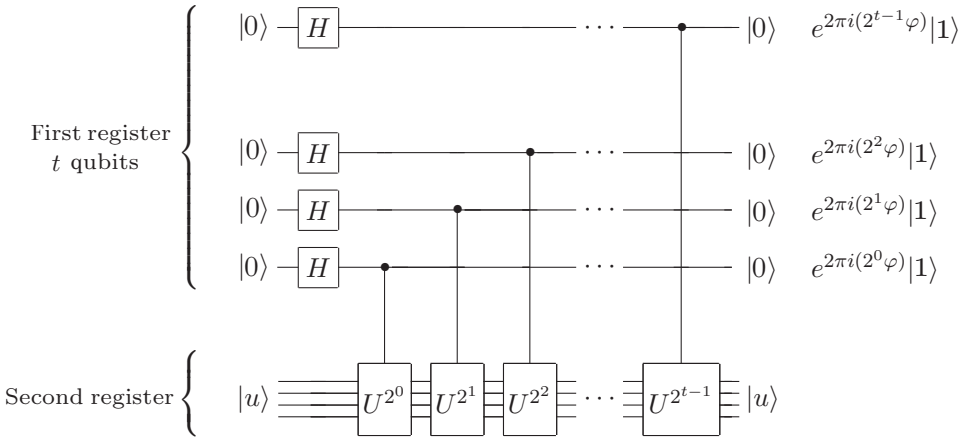

Figure 5.2. The first stage of the phase estimation procedure. Normalization factors of $1/\sqrt{2}$ have been omitted, on the right.

Exercise 5.7: Additional insight into the circuit in Figure 5.2 may be obtained by showing, as you should now do, that the effect of the sequence of controlled-U operations like that in Figure 5.2 is to take the state $|j\rangle|u\rangle$ to $|j\rangle U^j|u\rangle$. (Note that this does not depend on $|u\rangle$ being an eigenstate of U.)

The second stage of phase estimation is to apply the *inverse* quantum Fourier transform on the first register. This is obtained by reversing the circuit for the quantum Fourier transform in the previous section (Exercise 5.5), and can be done in $\Theta(t^2)$ steps. The third and final stage of phase estimation is to read out the state of the first register by doing a measurement in the computational basis. We will show that this provides a pretty good estimate of φ. An overall schematic of the algorithm is shown in Figure 5.3.

To sharpen our intuition as to why phase estimation works, suppose φ may be expressed exactly in t bits, as $\varphi = 0.\varphi_1\ldots\varphi_t$. Then the state (5.20) resulting from the first stage of phase estimation may be rewritten

$$\frac{1}{2^{t/2}}\left(|0\rangle + e^{2\pi i 0.\varphi_t}|1\rangle\right)\left(|0\rangle + e^{2\pi i 0.\varphi_{t-1}\varphi_t}|1\rangle\right)\ldots\left(|0\rangle + e^{2\pi i 0.\varphi_1\varphi_2\cdots\varphi_t}|1\rangle\right)\,. \tag{5.21}$$

The second stage of phase estimation is to apply the inverse quantum Fourier transform. But comparing the previous equation with the product form for the Fourier transform, Equation (5.4), we see that the output state from the second stage is the product state $|\varphi_1\ldots\varphi_t\rangle$. A measurement in the computational basis therefore gives us φ exactly!

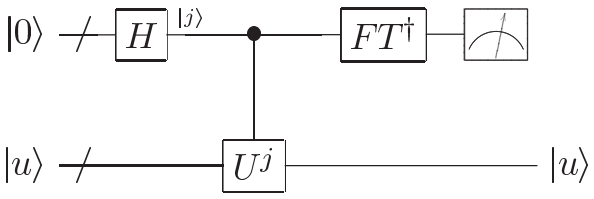

Figure 5.3. Schematic of the overall phase estimation procedure. The top t qubits (the '/' denotes a bundle of wires, as usual) are the first register, and the bottom qubits are the second register, numbering as many as required to perform U. $|u\rangle$ is an eigenstate of U with eigenvalue $e^{2\pi i\varphi}$. The output of the measurement is an approximation to φ accurate to $t - \left\lceil \log\left(2 + \frac{1}{2\epsilon}\right)\right\rceil$ bits, with probability of success at least $1 - \epsilon$.

Summarizing, the phase estimation algorithm allows one to estimate the phase φ of an eigenvalue of a unitary operator U, given the corresponding eigenvector $|u\rangle$. An essential feature at the heart of this procedure is the ability of the inverse Fourier transform to perform the transformation

$$\frac{1}{2^{t/2}} \sum_{j=0}^{2^t-1} e^{2\pi i\varphi j}|j\rangle|u\rangle \rightarrow |\tilde{\varphi}\rangle|u\rangle\,, \tag{5.22}$$

where $|\tilde{\varphi}\rangle$ denotes a state which is a good estimator for φ when measured.

5.2.1 Performance and requirements

The above analysis applies to the ideal case, where φ can be written exactly with a t bit binary expansion. What happens when this is not the case? It turns out that the procedure we have described will produce a pretty good approximation to φ with high probability, as foreshadowed by the notation used in (5.22). Showing this requires some careful manipulations.

Let b be the integer in the range 0 to $2^t - 1$ such that $b/2^t = 0.b_1 \ldots b_t$ is the best t bit approximation to φ which is less than φ. That is, the difference $\delta \equiv \varphi - b/2^t$ between φ and $b/2^t$ satisfies $0 \leq \delta \leq 2^{-t}$. We aim to show that the observation at the end of the phase estimation procedure produces a result which is close to b, and thus enables us to estimate φ accurately, with high probability. Applying the inverse quantum Fourier transform to the state (5.20) produces the state

$$\frac{1}{2^t} \sum_{k,l=0}^{2^t-1} e^{\frac{-2\pi ikl}{2^t}} e^{2\pi i\varphi k}|l\rangle\,. \tag{5.23}$$

Let α_l be the amplitude of $|(b + l)(\mathrm{mod}\ 2^t)\rangle$,

$$\alpha_l \equiv \frac{1}{2^t} \sum_{k=0}^{2^t-1} \left(e^{2\pi i(\varphi - (b+l)/2^t)}\right)^k\,. \tag{5.24}$$

This is the sum of a geometric series, so

$$\alpha_l = \frac{1}{2^t}\left(\frac{1 - e^{2\pi i(2^t\varphi - (b+l))}}{1 - e^{2\pi i(\varphi - (b+l)/2^t)}}\right) \tag{5.25}$$

$$= \frac{1}{2^t}\left(\frac{1-e^{2\pi i(2^t\delta-l)}}{1-e^{2\pi i(\delta-l/2^t)}}\right). \tag{5.26}$$

Suppose the outcome of the final measurement is m. We aim to bound the probability of obtaining a value of m such that $|m-b|>e$, where e is a positive integer characterizing our desired tolerance to error. The probability of observing such an m is given by

$$p(|m-b|>e) = \sum_{-2^{t-1}<l\leq -(e+1)} |\alpha_l|^2 + \sum_{e+1\leq l\leq 2^{t-1}} |\alpha_l|^2 . \tag{5.27}$$

But for any real θ, $|1-\exp(i\theta)|\leq 2$, so

$$|\alpha_l| \leq \frac{2}{2^t\,|1-e^{2\pi i(\delta-l/2^t)}|}. \tag{5.28}$$

By elementary geometry or calculus $|1-\exp(i\theta)|\geq 2|\theta|/\pi$ whenever $-\pi\leq\theta\leq\pi$. But when $-2^{t-1}<l\leq 2^{t-1}$ we have $-\pi\leq 2\pi(\delta-l/2^t)\leq\pi$. Thus

$$|\alpha_l| \leq \frac{1}{2^{t+1}(\delta-l/2^t)}. \tag{5.29}$$

Combining (5.27) and (5.29) gives

$$p(|m-b|>e) \leq \frac{1}{4}\left[\sum_{l=-2^{t-1}+1}^{-(e+1)} \frac{1}{(l-2^t\delta)^2} + \sum_{l=e+1}^{2^{t-1}} \frac{1}{(l-2^t\delta)^2}\right]. \tag{5.30}$$

Recalling that $0\leq 2^t\delta\leq 1$, we obtain

$$p(|m-b|>e) \leq \frac{1}{4}\left[\sum_{l=-2^{t-1}+1}^{-(e+1)} \frac{1}{l^2} + \sum_{l=e+1}^{2^{t-1}} \frac{1}{(l-1)^2}\right] \tag{5.31}$$

$$\leq \frac{1}{2}\sum_{l=e}^{2^{t-1}-1} \frac{1}{l^2} \tag{5.32}$$

$$\leq \frac{1}{2}\int_{e-1}^{2^{t-1}-1} dl\,\frac{1}{l^2} \tag{5.33}$$

$$= \frac{1}{2(e-1)}. \tag{5.34}$$

Suppose we wish to approximate φ to an accuracy 2^{-n}, that is, we choose $e=2^{t-n}-1$. By making use of $t=n+p$ qubits in the phase estimation algorithm we see from (5.34) that the probability of obtaining an approximation correct to this accuracy is at least $1-1/2(2^p-2)$. Thus to successfully obtain φ accurate to n bits with probability of success at least $1-\epsilon$ we choose

$$t = n + \left\lceil \log\left(2+\frac{1}{2\epsilon}\right)\right\rceil . \tag{5.35}$$

In order to make use of the phase estimation algorithm, we need to be able to prepare an eigenstate $|u\rangle$ of U. What if we do not know how to prepare such an eigenstate? Suppose that we prepare some other state $|\psi\rangle$ in place of $|u\rangle$. Expanding this state in terms of eigenstates $|u\rangle$ of U gives $|\psi\rangle=\sum_u c_u|u\rangle$. Suppose the eigenstate $|u\rangle$ has eigenvalue $e^{2\pi i\varphi_u}$. Intuitively, the result of running the phase estimation algorithm will be to give

as output a state close to $\sum_u c_u|\widetilde{\varphi_u}\rangle|u\rangle$, where $\widetilde{\varphi_u}$ is a pretty good approximation to the phase φ_u. Therefore, we expect that reading out the first register will give us a good approximation to φ_u, where u is chosen at random with probability $|c_u|^2$. Making this argument rigorous is left for Exercise 5.8. This procedure allows us to avoid preparing a (possibly unknown) eigenstate, at the cost of introducing some additional randomness into the algorithm.

Exercise 5.8: Suppose the phase estimation algorithm takes the state $|0\rangle|u\rangle$ to the state $|\widetilde{\varphi_u}\rangle|u\rangle$, so that given the input $|0\rangle\left(\sum_u c_u|u\rangle\right)$, the algorithm outputs $\sum_u c_u|\widetilde{\varphi_u}\rangle|u\rangle$. Show that if t is chosen according to (5.35), then the probability for measuring φ_u accurate to n bits at the conclusion of the phase estimation algorithm is at least $|c_u|^2(1-\epsilon)$.

Why is phase estimation interesting? For its own sake, phase estimation solves a problem which is both non-trivial and interesting from a physical point of view: how to estimate the eigenvalue associated to a given eigenvector of a unitary operator. Its real use, though, comes from the observation that other interesting problems can be reduced to phase estimation, as will be shown in subsequent sections. The phase estimation algorithm is summarized below.

Algorithm: Quantum phase estimation

Inputs: (1) A black box wich performs a controlled-U^j operation, for integer j, (2) an eigenstate $|u\rangle$ of U with eigenvalue $e^{2\pi i\varphi_u}$, and (3) $t = n + \lceil \log\left(2 + \frac{1}{2\epsilon}\right)\rceil$ qubits initialized to $|0\rangle$.

Outputs: An n-bit approximation $\widetilde{\varphi_u}$ to φ_u.

Runtime: $O(t^2)$ operations and one call to controlled-U^j black box. Succeeds with probability at least $1-\epsilon$.

Procedure:

1.	$\|0\rangle\|u\rangle$	initial state
2.	$\rightarrow \frac{1}{\sqrt{2^t}}\sum_{j=0}^{2^t-1}\|j\rangle\|u\rangle$	create superposition
3.	$\rightarrow \frac{1}{\sqrt{2^t}}\sum_{j=0}^{2^t-1}\|j\rangle U^j\|u\rangle$	apply black box
	$= \frac{1}{\sqrt{2^t}}\sum_{j=0}^{2^t-1} e^{2\pi i j\varphi_u}\|j\rangle\|u\rangle$	result of black box
4.	$\rightarrow \|\widetilde{\varphi_u}\rangle\|u\rangle$	apply inverse Fourier transform
5.	$\rightarrow \widetilde{\varphi_u}$	measure first register

Exercise 5.9: Let U be a unitary transform with eigenvalues ± 1, which acts on a state $|\psi\rangle$. Using the phase estimation procedure, construct a quantum circuit to collapse $|\psi\rangle$ into one or the other of the two eigenspaces of U, giving also a

classical indicator as to which space the final state is in. Compare your result with Exercise 4.34.

5.3 Applications: order-finding and factoring

The phase estimation procedure can be used to solve a variety of interesting problems. We now describe two of the most interesting of these problems: the *order-finding problem*, and the *factoring problem*. These two problems are, in fact, equivalent to one another, so in Section 5.3.1 we explain a quantum algorithm for solving the order-finding problem, and in Section 5.3.2 we explain how the order-finding problem implies the ability to factor as well.

To understand the quantum algorithms for factoring and order-finding requires a little background in number theory. All the required materials are collected together in Appendix 4. The description we give over the next two sections focuses on the quantum aspects of the problem, and requires only a little familiarity with modular arithmetic to be readable. Detailed proofs of the number-theoretic results we quote here may be found in Appendix 4.

The fast quantum algorithms for order-finding and factoring are interesting for at least three reasons. First, and most important in our opinion, they provide evidence for the idea that quantum computers may be inherently more powerful than classical computers, and provide a credible challenge to the strong Church–Turing thesis. Second, both problems are of sufficient intrinsic worth to justify interest in any novel algorithm, be it classical or quantum. Third, and most important from a practical standpoint, efficient algorithms for order-finding and factoring can be used to break the RSA public-key cryptosystem (Appendix 5).

5.3.1 Application: order-finding

For positive integers x and N, $x < N$, with no common factors, the *order* of x modulo N is defined to be the least positive integer, r, such that $x^r = 1 (\text{mod } N)$. The order-finding problem is to determine the order for some specified x and N. Order-finding is believed to be a hard problem on a classical computer, in the sense that no algorithm is known to solve the problem using resources polynomial in the $O(L)$ bits needed to specify the problem, where $L \equiv \lceil \log(N) \rceil$ is the number of bits needed to specify N. In this section we explain how phase estimation may be used to obtain an efficient quantum algorithm for order-finding.

Exercise 5.10: Show that the order of $x = 5$ modulo $N = 21$ is 6.

Exercise 5.11: Show that the order of x satisfies $r \leq N$.

The quantum algorithm for order-finding is just the phase estimation algorithm applied to the unitary operator

$$U|y\rangle \equiv |xy (\text{mod } N)\rangle \,, \tag{5.36}$$

with $y \in \{0,1\}^L$. (Note that here and below, when $N \leq y \leq 2^L - 1$, we use the convention that $xy (\text{mod } N)$ is just y again. That is, U only acts non-trivially when