dent. Whether physically reasonable models of computation exist which go beyond the quantum circuit model is a fascinating question which we leave open for you.

4.7 Simulation of quantum systems

> *Perhaps [...] we need a mathematical theory of quantum automata. [...] the quantum state space has far greater capacity than the classical one: for a classical system with N states, its quantum version allowing superposition accommodates c^N states. When we join two classical systems, their number of states N_1 and N_2 are multiplied, and in the quantum case we get the exponential growth $c^{N_1 N_2}$. [...] These crude estimates show that the quantum behavior of the system might be much more complex than its classical simulation.*
> – Yu Manin (1980)[Man80], as translated in [Man99]

> *The quantum-mechanical computation of one molecule of methane requires 10^{42} grid points. Assuming that at each point we have to perform only 10 elementary operations, and that the computation is performed at the extremely low temperature $T = 3 \times 10^{-3} K$, we would still have to use all the energy produced on Earth during the last century.*
> – R. P. Poplavskii (1975)[Pop75], as quoted by Manin

> *Can physics be simulated by a universal computer? [...] the physical world is quantum mechanical, and therefore the proper problem is the simulation of quantum physics [...] the full description of quantum mechanics for a large system with R particles [...] has too many variables, it* **cannot be simulated** *with a normal computer with a number of elements proportional to R [... but it can be simulated with] quantum computer elements. [...] Can a quantum system be probabilistically simulated by a classical (probabilistic, I'd assume) universal computer? [...] If you take the computer to be the classical kind I've described so far [..] the answer is certainly, No!*
> – Richard P. Feynman (1982)[Fey82]

Let us close out this chapter by providing an interesting and useful application of the quantum circuit model. One of the most important practical applications of computation is the simulation of physical systems. For example, in the engineering design of a new building, finite element analysis and modeling is used to ensure safety while minimizing cost. Cars are made lightweight, structurally sound, attractive, and inexpensive, by using computer aided design. Modern aeronautical engineering depends heavily on computational fluid dynamics simulations for aircraft designs. Nuclear weapons are no longer exploded (for the most part), but rather, tested by exhaustive computational modeling. Examples abound, because of the tremendous practical applications of predictive simulations. We begin by describing some instances of the simulation problem, then we present a quantum algorithm for simulation and an illustrative example, concluding with some perspective on this application.

4.7.1 Simulation in action

The heart of simulation is the solution of differential equations which capture the physical laws governing the dynamical behavior of a system. Some examples include Newton's

law,

$$\frac{d}{dt}\left(m\frac{dx}{dt}\right) = F\,, \tag{4.88}$$

Poisson's equation,

$$-\vec{\nabla}\cdot(k\,\vec{\nabla}\vec{u}) = \vec{Q}\,, \tag{4.89}$$

the electromagnetic vector wave equation,

$$\vec{\nabla}\cdot\vec{\nabla}\vec{E} = \epsilon_0\mu_0\frac{\partial^2\vec{E}}{\partial t^2}\,, \tag{4.90}$$

and the diffusion equation,

$$\vec{\nabla}^2\psi = \frac{1}{a^2}\frac{\partial\psi}{\partial t}\,, \tag{4.91}$$

just to name a very few. The goal is generally: given an initial state of the system, what is the state at some other time and/or position? Solutions are usually obtained by *approximating* the state with a digital representation, then *discretizing* the differential equation in space and time such that an iterative application of a procedure carries the state from the initial to the final conditions. Importantly, the error in this procedure is bounded, and known not to grow faster than some small power of the number of iterations. Furthermore, *not* all dynamical systems can be simulated *efficiently*: generally, only those systems which can be described efficiently can be simulated efficiently.

Simulation of quantum systems by classical computers is possible, but generally only very inefficiently. The dynamical behavior of many simple quantum systems is governed by Schrödinger's equation,

$$i\hbar\frac{d}{dt}|\psi\rangle = H|\psi\rangle\,. \tag{4.92}$$

We will find it convenient to absorb $\hbar$ into H, and use this convention for the rest of this section. For a typical Hamiltonian of interest to physicists dealing with real particles in space (rather than abstract systems such as qubits, which we have been dealing with!), this reduces to

$$i\frac{\partial}{\partial t}\psi(x) = \left[-\frac{1}{2m}\frac{\partial^2}{\partial x^2} + V(x)\right]\psi(x)\,, \tag{4.93}$$

using a convention known as the position representation $\langle x|\psi\rangle = \psi(x)$. This is an elliptical equation very much like Equation (4.91). So just simulating Schrödinger's equation is not the especial difficulty faced in simulating quantum systems. What is the difficulty?

The key challenge in simulating quantum systems is the *exponential* number of differential equations which must be solved. For one qubit evolving according to the Schrödinger equation, a system of two differential equations must be solved; for two qubits, four equations; and for n qubits, 2^n equations. Sometimes, insightful approximations can be made which reduce the effective number of equations involved, thus making classical simulation of the quantum system feasible. However, there are many physically interesting quantum systems for which no such approximations are known.

Exercise 4.46: (Exponential complexity growth of quantum systems) Let ρ be a density matrix describing the state of n qubits. Show that describing ρ requires $4^n - 1$ independent real numbers.

The reader with a physics background may appreciate that there are many important quantum systems for which classical simulation is intractable. These include the Hubbard model, a model of interacting fermionic particles with the Hamiltonian

$$H = \sum_{k=1}^{n} V_0 n_{k\uparrow} n_{k\downarrow} + \sum_{k,j \text{ neighbors},\sigma} t_0 c^*_{k\sigma} c_{j\sigma} \,, \tag{4.94}$$

which is useful in the study of superconductivity and magnetism, the Ising model,

$$H = \sum_{k=1}^{n} \vec{\sigma}_k \cdot \vec{\sigma}_{k+1} \,, \tag{4.95}$$

and many others. Solutions to such models give many physical properties such as the dielectric constant, conductivity, and magnetic susceptibility of materials. More sophisticated models such as quantum electrodynamics (QED) and quantum chromodynamics (QCD) can be used to compute constants such as the mass of the proton.

Quantum computers can efficiently simulate quantum systems for which there is no known efficient classical simulation. Intuitively, this is possible for much the same reason any quantum circuit can be constructed from a universal set of quantum gates. Moreover, just as there exist unitary operations which cannot be *efficiently* approximated, it is possible in principle to imagine quantum systems with Hamiltonians which cannot be efficiently simulated on a quantum computer. Of course, we believe that such systems aren't actually realized in Nature, otherwise we'd be able to exploit them to do information processing beyond the quantum circuit model.

4.7.2 The quantum simulation algorithm

Classical simulation begins with the realization that in solving a simple differential equation such as $dy/dt = f(y)$, to first order, it is known that $y(t + \Delta t) \approx y(t) + f(y)\Delta t$. Similarly, the quantum case is concerned with the solution of $id|\psi\rangle/dt = H|\psi\rangle$, which, for a time-independent H, is just

$$|\psi(t)\rangle = e^{-iHt}|\psi(0)\rangle \,. \tag{4.96}$$

Since H is usually extremely difficult to exponentiate (it may be sparse, but it is also exponentially large), a good beginning is the first order solution $|\psi(t + \Delta t)\rangle \approx (I - iH\Delta t)|\psi(t)\rangle$. This is tractable, because for many Hamiltonians H it is straightforward to compose quantum gates to efficiently approximate $I - iH\Delta t$. However, such first order solutions are generally not very satisfactory.

Efficient approximation of the solution to Equation (4.96), to high order, is possible for many classes of Hamiltonian. For example, in most physical systems, the Hamiltonian can be written as a sum over many local interactions. Specifically, for a system of n particles,

$$H = \sum_{k=1}^{L} H_k \,, \tag{4.97}$$

where each H_k acts on at most a constant c number of systems, and L is a polynomial in n. For example, the terms H_k are often just two-body interactions such as $X_i X_j$ and one-body Hamiltonians such as X_i. Both the Hubbard and Ising models have Hamiltonians of this form. Such locality is quite physically reasonable, and originates in many systems

from the fact that most interactions fall off with increasing distance or difference in energy. There are sometimes additional global symmetry constraints such as particle statistics; we shall come to those shortly. The important point is that although e^{-iHt} is difficult to compute, $e^{-iH_k t}$ acts on a much smaller subsystem, and is straightforward to approximate using quantum circuits. But because $[H_j, H_k] \neq 0$ in general, $e^{-iHt} \neq \prod_k e^{-iH_k t}$! How, then, can $e^{-iH_k t}$ be useful in constructing e^{-iHt}?

Exercise 4.47: For $H = \sum_k^L H_k$, prove that $e^{-iHt} = e^{-iH_1 t} e^{-iH_2 t} \ldots e^{-iH_L t}$ for all t if $[H_j, H_k] = 0$, for all j, k.

Exercise 4.48: Show that the restriction of H_k to involve at most c particles *implies* that in the sum (4.97), L is upper bounded by a polynomial in n.

The heart of quantum simulation algorithms is the following asymptotic approximation theorem:

Theorem 4.3: (**Trotter formula**) Let A and B be Hermitian operators. Then for any real t,

$$\lim_{n\to\infty} (e^{iAt/n} e^{iBt/n})^n = e^{i(A+B)t} . \tag{4.98}$$

Note that (4.98) is true even if A and B do not commute. Even more interestingly, perhaps, it can be generalized to hold for A and B which are generators of certain kinds of semigroups, which correspond to general quantum operations; we shall describe such generators (the 'Lindblad form') in Section 8.4.1 of Chapter 8. For now, we only consider the case of A and B being Hermitian matrices.

Proof
By definition,

$$e^{iAt/n} = I + \frac{1}{n} iAt + O\left(\frac{1}{n^2}\right) , \tag{4.99}$$

and thus

$$e^{iAt/n} e^{iBt/n} = I + \frac{1}{n} i(A+B)t + O\left(\frac{1}{n^2}\right) . \tag{4.100}$$

Taking products of these gives us

$$(e^{iAt/n} e^{iBt/n})^n = I + \sum_{k=1}^{n} \left(\begin{array}{c} n \\ k \end{array} \right) \frac{1}{n^k} \left[i(A+B)t \right]^k + O\left(\frac{1}{n}\right) , \tag{4.101}$$

and since $\left(\begin{array}{c} n \\ k \end{array} \right) \frac{1}{n^k} = \left(1 + O\left(\frac{1}{n}\right)\right)/k!$, this gives

$$\lim_{n\to\infty} (e^{iAt/n} e^{iBt/n})^n = \lim_{n\to\infty} \sum_{k=0}^{n} \frac{(i(A+B)t)^k}{k!} \left(1 + O\left(\frac{1}{n}\right)\right) + O\left(\frac{1}{n}\right) = e^{i(A+B)t} . \tag{4.102}$$

□

Modifications of the Trotter formula provide the methods by which higher order

approximations can be derived for performing quantum simulations. For example, using similar reasoning to the proof above, it can be shown that

$$e^{i(A+B)\Delta t} = e^{iA\Delta t}e^{iB\Delta t} + O(\Delta t^2)\,. \tag{4.103}$$

Similarly,

$$e^{i(A+B)\Delta t} = e^{iA\Delta t/2}e^{iB\Delta t}e^{iA\Delta t/2} + O(\Delta t^3)\,. \tag{4.104}$$

An overview of the quantum simulation algorithm is given below, and an explicit example of simulating the one-dimensional non-relativistic Schrödinger equation is shown in Box 4.2.

Algorithm: Quantum simulation

Inputs: (1) A Hamiltonian $H = \sum_k H_k$ acting on an N-dimensional system, where each H_k acts on a small subsystem of size independent of N, (2) an initial state $|\psi_0\rangle$, of the system at $t = 0$, (3) a positive, non-zero accuracy δ, and (3) a time t_f at which the evolved state is desired.

Outputs: A state $|\tilde{\psi}(t_f)\rangle$ such that $|\langle\tilde{\psi}(t_f)|e^{-iHt_f}|\psi_0\rangle|^2 \geq 1 - \delta$.

Runtime: $O(\text{poly}(1/\delta))$ operations.

Procedure: Choose a representation such that the state $|\tilde{\psi}\rangle$ of $n = \text{poly}(\log N)$ qubits approximates the system and the operators $e^{-iH_k\Delta t}$ have efficient quantum circuit approximations. Select an approximation method (see for example Equations (4.103)–(4.105)) and Δt such that the expected error is acceptable (and $j\Delta t = t_f$ for an integer j), construct the corresponding quantum circuit $U_{\Delta t}$ for the iterative step, and do:

1.	$\|\tilde{\psi}_0\rangle \leftarrow \|\psi_0\rangle$; $j = 0$	initialize state
2.	$\rightarrow \|\tilde{\psi}_{j+1}\rangle = U_{\Delta t}\|\tilde{\psi}_j\rangle$	iterative update
3.	$\rightarrow j = j + 1$; goto **2** until $j\Delta t \geq t_f$	loop
4.	$\rightarrow \|\tilde{\psi}(t_f)\rangle = \|\tilde{\psi}_j\rangle$	final result

Exercise 4.49: (Baker–Campbell–Hausdorf formula) Prove that

$$e^{(A+B)\Delta t} = e^{A\Delta t}e^{B\Delta t}e^{-\frac{1}{2}[A,B]\Delta t^2} + O(\Delta t^3)\,, \tag{4.105}$$

and also prove Equations (4.103) and (4.104).

Exercise 4.50: Let $H = \sum_k^L H_k$, and define

$$U_{\Delta t} = \left[e^{-iH_1\Delta t}e^{-iH_2\Delta t}\ldots e^{-iH_L\Delta t}\right]\left[e^{-iH_L\Delta t}e^{-iH_{L-1}\Delta t}\ldots e^{-iH_1\Delta t}\right]\,. \tag{4.106}$$

(a) Prove that $U_{\Delta t} = e^{-2iH\Delta t} + O(\Delta t^3)$.

(b) Use the results in Box 4.1 to prove that for a positive integer m,

$$E(U_{\Delta t}^m, e^{-2miH\Delta t}) \leq m\alpha\Delta t^3\,, \tag{4.107}$$

for some constant α.

Box 4.2: Quantum simulation of Schrödinger's equation

The methods and limitations of quantum simulation may be illustrated by the following example, drawn from the conventional models studied by physicists, rather than the abstract qubit model. Consider a single particle living on a line, in a one-dimensional potential $V(x)$, governed by the Hamiltonian

$$H = \frac{p^2}{2m} + V(x)\,, \tag{4.108}$$

where p is the momentum operator and x is the position operator. The eigenvalues of x are continuous, and the system state $|\psi\rangle$ resides in an infinite dimensional Hilbert space; in the x basis, it can be written as

$$|\psi\rangle = \int_{-\infty}^{\infty} |x\rangle\langle x|\psi\rangle\, dx\,. \tag{4.109}$$

In practice, only some finite region is of interest, which we may take to be the range $-d \leq x \leq d$. Furthermore, it is possible to choose a differential step size Δx sufficiently small compared to the shortest wavelength in the system such that

$$|\tilde{\psi}\rangle = \sum_{k=-d/\Delta x}^{d/\Delta x} a_k |k\Delta x\rangle \tag{4.110}$$

provides a good physical approximation of $|\psi\rangle$. This state can be represented using $n = \lceil \log(2d/\Delta x + 1) \rceil$ qubits; we simply replace the basis $|k\Delta x\rangle$ (an eigenstate of the x operator) with $|k\rangle$, a computational basis state of n qubits. Note that only n qubits are required for this simulation, whereas classically 2^n complex numbers would have to be kept track of, thus leading to an exponential resource saving when performing the simulation on a quantum computer.

Computation of $|\tilde{\psi}(t)\rangle = e^{-iHt}|\tilde{\psi}(0)\rangle$ must utilize one of the approximations of Equations (4.103)–(4.105) because in general $H_1 = V(x)$ does not commute with $H_0 = p^2/2m$. Thus, we must be able to compute $e^{-iH_1\Delta t}$ and $e^{-iH_0\Delta t}$. Because $|\tilde{\psi}\rangle$ is expressed in the eigenbasis of H_1, $e^{-iH_1\Delta t}$ is a diagonal transformation of the form

$$|k\rangle \rightarrow e^{-iV(k\Delta x)\Delta t}|k\rangle\,. \tag{4.111}$$

It is straightforward to compute this, since we can compute $V(k\Delta x)\Delta t$. (See also Problem 4.1.) The second term is also simple, because x and p are conjugate variables related by a quantum Fourier transform $U_{\text{FFT}} x U_{\text{FFT}}^\dagger = p$, and thus $e^{-iH_0\Delta t} = U_{\text{FFT}} e^{-ix^2\Delta t/2m} U_{\text{FFT}}^\dagger$; to compute $e^{-iH_0\Delta t}$, do

$$|k\rangle \rightarrow U_{\text{FFT}} e^{-ix^2/2m} U_{\text{FFT}}^\dagger |k\rangle\,. \tag{4.112}$$

The construction of U_{FFT} is discussed in Chapter 5.

4.7.3 An illustrative example

The procedure we have described for quantum simulations has concentrated on simulating Hamiltonians which are sums of local interations. However, this is not a fundamental

requirement! As the following example illustrates, efficient quantum simulations are possible even for Hamiltonians which act non-trivially on all or nearly all parts of a large system.

Suppose we have the Hamiltonian

$$H = Z_1 \otimes Z_2 \otimes \cdots \otimes Z_n \,, \tag{4.113}$$

which acts on an n qubit system. Despite this being an interaction involving all of the system, indeed, it can be simulated efficiently. What we desire is a simple quantum circuit which implements $e^{-iH\Delta t}$, for arbitrary values of Δt. A circuit doing precisely this, for $n = 3$, is shown in Figure 4.19. The main insight is that although the Hamiltonian involves all the qubits in the system, it does so in a *classical* manner: the phase shift applied to the system is $e^{-i\Delta t}$ if the *parity* of the n qubits in the computational basis is even; otherwise, the phase shift should be $e^{i\Delta t}$. Thus, simple simulation of H is possible by first classically computing the parity (storing the result in an ancilla qubit), then applying the appropriate phase shift conditioned on the parity, then uncomputing the parity (to erase the ancilla). This strategy clearly works not only for $n = 3$, but also for arbitrary values of n.

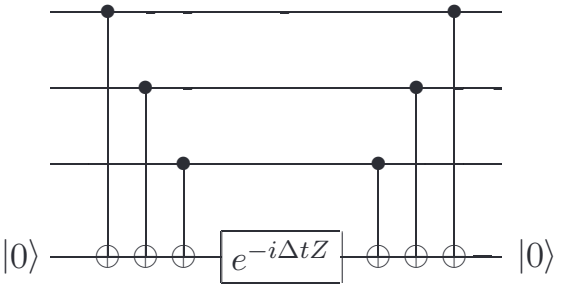

Figure 4.19. Quantum circuit for simulating the Hamiltonian $H = Z_1 \otimes Z_2 \otimes Z_3$ for time Δt.

Furthermore, extending the same procedure allows us to simulate more complicated extended Hamiltonians. Specifically, we can efficiently simulate any Hamiltonian of the form

$$H = \bigotimes_{k=1}^{n} \sigma_{c(k)}^{k} \,, \tag{4.114}$$

where $\sigma_{c(k)}^{k}$ is a Pauli matrix (or the identity) acting on the kth qubit, with $c(k) \in \{0, 1, 2, 3\}$ specifying one of $\{I, X, Y, Z\}$. The qubits upon which the identity operation is performed can be disregarded, and X or Y terms can be transformed by single qubit gates to Z operations. This leaves us with a Hamiltonian of the form of (4.113), which is simulated as described above.

Exercise 4.51: Construct a quantum circuit to simulate the Hamiltonian

$$H = X_1 \otimes Y_2 \otimes Z_3 \,, \tag{4.115}$$

performing the unitary transform $e^{-i\Delta t H}$ for any Δt.

Using this procedure allows us to simulate a wide class of Hamiltonians containing terms which are not local. In particular, it is possible to simulate a Hamiltonian of the form

$H = \sum_{k=1}^{L} H_k$ where the only restriction is that the individual H_k have a tensor product structure, and that L is polynomial in the total number of particles n. More generally, all that is required is that there be an efficient circuit to simulate each H_k separately. As an example, the Hamiltonian $H = \sum_{k=1}^{n} X_k + Z^{\otimes n}$ can easily be simulated using the above techniques. Such Hamiltonians typically do not arise in Nature. However, they provide a new and possibly valuable vista on the world of quantum algorithms.

4.7.4 Perspectives on quantum simulation

The quantum simulation algorithm is very similar to classical methods, but also differs in a fundamental way. Each iteration of the quantum algorithm must completely replace the old state with the new one; there is no way to obtain (non-trivial) information from an intermediate step without significantly changing the algorithm, because the state is a quantum one. Furthermore, the final measurement must be chosen cleverly to provide the desired result, because it disturbs the quantum state. Of course, the quantum simulation can be repeated to obtain statistics, but it is desirable to repeat the algorithm only at most a polynomial number of times. It may be that even though the simulation can be performed efficiently, there is no way to efficiently perform a desired measurement.

Also, there are Hamiltonians which simply can't be simulated efficiently. In Section 4.5.4, we saw that there exist unitary transformations which quantum computers cannot efficiently approximate. As a corollary, not all Hamiltonian evolutions can be efficiently simulated on a quantum computer, for if this were possible, then all unitary transformations could be efficiently approximated!

Another difficult problem – one which is very interesting – is the simulation of equilibration processes. A system with Hamiltonian H in contact with an environment at temperature T will generally come to thermal equilibrium in a state known as the *Gibbs* state, $\rho_{\text{therm}} = e^{-H/k_B T}/\mathcal{Z}$, where k_B is Boltzmann's constant, and $\mathcal{Z} = \text{tr}\, e^{-H/k_B T}$ is the usual partition function normalization, which ensures that $\text{tr}(\rho) = 1$. The process by which this equilibration occurs is not very well understood, although certain requirements are known: the environment must be large, it must have non-zero population in states with energies matching the eigenstates of H, and its coupling with the system should be weak. Obtaining ρ_{therm} for arbitrary H and T is generally an exponentially difficult problem for a classical computer. Might a quantum computer be able to solve this efficiently? We do not yet know.

On the other hand, as we discussed above many interesting quantum problems can indeed be simulated efficiently with a quantum computer, even when they have extra constraints beyond the simple algorithms presented here. A particular class of these involve global symmetries originating from particle statistics. In the everyday world, we are used to being able to identify different particles; tennis balls can be followed around a tennis court, keeping track of which is which. This ability to keep track of which object is which is a general feature of classical objects – by continuously measuring the position of a classical particle it can be tracked at all times, and thus uniquely distinguished from other particles. However, this breaks down in quantum mechanics, which prevents us from following the motion of individual particles exactly. If the two particles are inherently different, say a proton and an electron, then we can distinguish them by measuring the sign of the charge to tell which particle is which. But in the case of identical particles, like two electrons, it is found that they are truly indistinguishable.

Indistinguishability of particles places a constraint on the state vector of a system which

manifests itself in two ways. Experimentally, particles in Nature are found to come in two distinct flavors, known as bosons and fermions. The state vector of a system of bosons remains unchanged under permutation of any two constituents, reflecting their fundamental indistinguishability. Systems of fermions, in contrast, experience a sign change in their state vector under interchange of any two constituents. Both kinds of systems can be simulated efficiently on a quantum computer. The detailed description of how this is done is outside the scope of this book; suffice it to say the procedure is fairly straightforward. Given an initial state of the wrong symmetry, it can be properly symmetrized before the simulation begins. And the operators used in the simulation can be constructed to respect the desired symmetry, even allowing for the effects of higher order error terms. The reader who is interested in pursuing this and other topics further will find pointers to the literature in 'History and further reading,' at the end of the chapter.

Problem 4.1: (Computable phase shifts) Let m and n be positive integers. Suppose $f : \{0, \ldots, 2^m - 1\} \rightarrow \{0, \ldots, 2^n - 1\}$ is a classical function from m to n bits which may be computed reversibly using T Toffoli gates, as described in Section 3.2.5. That is, the function $(x, y) \rightarrow (x, y \oplus f(x))$ may be implemented using T Toffoli gates. Give a quantum circuit using $2T + n$ (or fewer) one, two, and three qubit gates to implement the unitary operation defined by

$$|x\rangle \rightarrow \exp\left(\frac{-2i\pi f(x)}{2^n}\right)|x\rangle . \tag{4.116}$$

Problem 4.2: Find a depth $O(\log n)$ construction for the $C^n(X)$ gate. (*Comment:* The depth of a circuit is the number of distinct timesteps at which gates are applied; the point of this problem is that it is possible to parallelize the $C^n(X)$ construction by applying many gates in parallel during the same timestep.)

Problem 4.3: (Alternate universality construction) Suppose U is a unitary matrix on n qubits. Define $H \equiv i \ln(U)$. Show that

(1) H is Hermitian, with eigenvalues in the range 0 to 2π.

(2) H can be written

$$H = \sum_g h_g g , \tag{4.117}$$

where h_g are real numbers and the sum is over all n-fold tensor products g of the Pauli matrices $\{I, X, Y, Z\}$.

(3) Let $\Delta = 1/k$, for some positive integer k. Explain how the unitary operation $\exp(-ih_g g\Delta)$ may be implemented using $O(n)$ one and two qubit operations.

(4) Show that

$$\exp(-iH\Delta) = \prod_g \exp(-ih_g g\Delta) + O(4^n\Delta^2) , \tag{4.118}$$

where the product is taken with respect to any fixed ordering of the n-fold tensor products of Pauli matrices, g.

(5) Show that

$$U = \left[\prod_g \exp(-ih_g g\Delta)\right]^k + O(4^n\Delta). \tag{4.119}$$

(6) Explain how to approximate U to within a distance $\epsilon > 0$ using $O(n16^n/\epsilon)$ one and two qubit unitary operations.

Problem 4.4: (Minimal Toffoli construction) (Research)

(1) What is the smallest number of two qubit gates that can be used to implement the Toffoli gate?
(2) What is the smallest number of one qubit gates and CNOT gates that can be used to implement the Toffoli gate?
(3) What is the smallest number of one qubit gates and controlled-Z gates that can be used to implement the Toffoli gate?

Problem 4.5: (Research) Construct a family of Hamiltonians, $\{H_n\}$, on n qubits, such that simulating H_n requires a number of operations super-polynomial in n. (*Comment:* This problem seems to be quite difficult.)

Problem 4.6: (Universality with prior entanglement) Controlled-NOT gates and single qubit gates form a universal set of quantum logic gates. Show that an alternative universal set of resources is comprised of single qubit unitaries, the ability to perform measurements of pairs of qubits in the Bell basis, and the ability to prepare arbitrary four qubit entangled states.

Summary of Chapter 4: Quantum circuits

- **Universality:** Any unitary operation on n qubits may be implemented exactly by composing single qubit and controlled-NOT gates.
- **Universality with a discrete set:** The Hadamard gate, phase gate, controlled-NOT gate, and $\pi/8$ gate are *universal* for quantum computation, in the sense that an arbitrary unitary operation on n qubits can be approximated to an arbitrary accuracy $\epsilon > 0$ using a circuit composed of only these gates. Replacing the $\pi/8$ gate in this list with the Toffoli gate also gives a universal family.
- **Not all unitary operations can be efficiently implemented:** There are unitary operations on n qubits which require $\Omega(2^n \log(1/\epsilon)/\log(n))$ gates to approximate to within a distance ϵ using any finite set of gates.
- **Simulation:** For a Hamiltonian $H = \sum_k H_k$ which is a sum of polynomially many terms H_k such that efficient quantum circuits for H_k can be constructed, a quantum computer can efficiently simulate the evolution e^{-iHt} and approximate $|\psi(t)\rangle = e^{-iHt}|\psi(0)\rangle$, given $|\psi(0)\rangle$.

History and further reading

The gate constructions in this chapter are drawn from a wide variety of sources. The paper by Barenco, Bennett, Cleve, DiVincenzo, Margolus, Shor, Sleator, Smolin, and Weinfurter[BBC+95] was the source of many of the circuit constructions in this chapter, and for the universality proof for single qubit and controlled-NOT gates. Another useful source of insights about quantum circuits is the paper by Beckman, Chari, Devabhaktuni, and Preskill[BCDP96]. A gentle and accessible introduction has been provided by DiVincenzo[DiV98]. The fact that measurements commute with control qubit terminals was pointed out by Griffiths and Niu[GN96].

The universality proof for two-level unitaries is due to Reck, Zeilinger, Bernstein, and Bertani[RZBB94]. The universality of the controlled-NOT and single qubit gates was proved by DiVincenzo[DiV95b]. The universal gate G in Exercise 4.44 is sometimes known as the Deutsch gate[Deu89]. Deutsch, Barenco, and Ekert[DBE95] and Lloyd[Llo95] independently proved that almost any two qubit quantum logic gate is universal. That errors caused by sequences of gates is at most the sum of the errors of the individual gates was proven by Bernstein and Vazirani [BV97]. The specific universal set of gates we have focused on – the Hadamard, phase, controlled-NOT and $\pi/8$ gates, was proved universal in Boykin, Mor, Pulver, Roychowdhury, and Vatan[BMP+99], which also contains a proof that θ defined by $\cos(\theta/2) \equiv \cos^2(\pi/8)$ is an irrational multiple of π. The bound in Section 4.5.4 is based on a paper by Knill[Kni95], which does a much more detailed investigation of the hardness of approximating arbitrary unitary operations using quantum circuits. In particular, Knill obtains tighter and more general bounds than we do, and his analysis applies also to cases where the universal set is a continuum of gates, not just a finite set, as we have considered.

The quantum circuit model of computation is due to Deutsch[Deu89], and was further developed by Yao[Yao93]. The latter paper showed that the quantum circuit model of computation is equivalent to the quantum Turing machine model. Quantum Turing machines were introduced in 1980 by Benioff[Ben80], further developed by Deutsch[Deu85] and Yao[Yao93], and their modern definition given by Bernstein and Vazirani[BV97]. The latter two papers also take first steps towards setting up a theory of quantum computational complexity, analogous to classical computational complexity theory. In particular, the inclusion **BQP** $\subseteq$ **PSPACE** and some slightly stronger results was proved by Bernstein and Vazirani. Knill and Laflamme[KL99] develop some fascinating connections between quantum and classical computational complexity. Other interesting work on quantum computational complexity includes the paper by Adleman, Demarrais and Huang[ADH97], and the paper by Watrous[Wat99]. The latter paper gives intriguing evidence to suggest that quantum computers are more powerful than classical computers in the setting of 'interactive proof systems'.

The suggestion that non-computational basis starting states may be used to obtain computational power beyond the quantum circuits model was made by Daniel Gottesman and Michael Nielson.

That quantum computers might simulate quantum systems more efficiently than classical computers was intimated by Manin[Man80] in 1980, and independently developed in more detail by Feynman[Fey82] in 1982. Much more detailed investigations were subsequently carried out by Abrams and Lloyd[AL97], Boghosian and Taylor[BT97], Sornborger and Stewart[SS99], Wiesner[Wie96], and Zalka[Zal98]. The Trotter formula is attributed to Trotter[Tro59], and was also proven by Chernoff[Che68], although the simpler form for

unitary operators is much older, and goes back to the time of Sophus Lie. The third order version of the Baker–Campbell–Hausdorff formula, Equation (4.104), was given by Sornborger and Stewart[SS99]. Abrams and Lloyd[AL97] give a procedure for simulating many-body Fermi systems on a quantum computer. Terhal and DiVincenzo address the problem of simulating the equilibration of quantum systems to the Gibbs state[TD98]. The method used to simulate the Schrödinger equation in Box 4.2 is due to Zalka[Zal98] and Wiesner[Wie96].

Exercise 4.25 is due to Vandersypen, and is related to work by Chau and Wilczek[CW95]. Exercise 4.45 is due to Boykin, Mor, Pulver, Roychowdhury, and Vatan[BMP+99]. Problem 4.2 is due to Gottesman. Problem 4.6 is due to Gottesman and Chuang[GC99].