Simulating Chemistry Using Quantum Computers

Ivan Kassal,* James D. Whitfield,*
Alejandro Perdomo-Ortiz, Man-Hong Yung,
and Alán Aspuru-Guzik

Department of Chemistry and Chemical Biology, Harvard University, Cambridge, Massachusetts 02138; email: aspuru@chemistry.harvard.edu

Annu. Rev. Phys. Chem. 2011. 62:185–207

First published online as a Review in Advance on December 13, 2010

The *Annual Review of Physical Chemistry* is online at physchem.annualreviews.org

This article's doi:
10.1146/annurev-physchem-032210-103512

0066-426X/11/0505-0185$20.00

*These authors contributed equally to this work.

Keywords

quantum computation, quantum algorithms, chemical dynamics, electronic structure

Abstract

The difficulty of simulating quantum systems, well known to quantum chemists, prompted the idea of quantum computation. One can avoid the steep scaling associated with the exact simulation of increasingly large quantum systems on conventional computers, by mapping the quantum system to another, more controllable one. In this review, we discuss to what extent the ideas in quantum computation, now a well-established field, have been applied to chemical problems. We describe algorithms that achieve significant advantages for the electronic-structure problem, the simulation of chemical dynamics, protein folding, and other tasks. Although theory is still ahead of experiment, we outline recent advances that have led to the first chemical calculations on small quantum information processors.

1. INTRODUCTION

AQC: adiabatic quantum computation

Qubit: quantum bit, generalization of the classical bit

Entangled state: a composite quantum state that is not a product state

One of the greatest challenges in quantum chemistry is to fully understand the complicated electronic structure of atoms and molecules. Over the past century, enormous progress has been made in describing the general behavior of relatively simple systems. In particular, combined with physical insights, elegant computational approaches have been developed, ranging from wave-function methods to quantum Monte Carlo and density functional theory. The challenge is that the Hilbert spaces of quantum systems grow exponentially with system size. Therefore, as these methods are extended to higher accuracy or to larger systems, the computational requirements become unreachable with current computers. This problem is not merely a consequence of technological limitations, but stems from the inherent difficulty of simulating quantum systems with computers based on classical mechanics. It is therefore important to know if the computational bottlenecks of classical computers can be solved by a computing model based on quantum mechanics—quantum computation—whose development has revolutionized our understanding of the connections between computer science and physics.

The idea of mapping the dynamics of a quantum system of interest onto the dynamics of a controllable quantum system was proposed in 1982 by Feynman (1) and developed in 1996 by Lloyd (2). Such a quantum computer would be able to obtain information inaccessible with classical computers. Consequently, quantum simulation promises to be a powerful new tool for quantum chemistry. In this article, we review the recent applications of quantum simulation to chemical problems that have proven difficult on conventional computers. After introducing basic concepts in quantum computation, we describe quantum algorithms for the exact, nonadiabatic simulation of chemical dynamics as well as for the full configuration interaction (FCI) treatment of electronic structure. We also discuss solving chemical optimization problems, such as lattice folding, using adiabatic quantum computation (AQC). Finally, we describe recent experimental implementations of these algorithms, including the first quantum simulations of chemical systems.

2. QUANTUM COMPUTATION

2.1. Differences Between Quantum and Classical Computation

There are fundamental differences between quantum and classical computers. Unlike the classical bit, which is always either a 0 or a 1, the basic unit of quantum information is the qubit (**Figure 1**), which can be in a superposition of $|0\rangle$ and $|1\rangle$: $\alpha|0\rangle + \beta|1\rangle$. States of n qubits are elements of an exponentially large, 2^n-dimensional, Hilbert space, spanned by a basis of the form $|x_1\rangle \cdots |x_n\rangle \equiv |x_1 \ldots x_n\rangle$, where each $|x_i\rangle$ is $|0\rangle$ or $|1\rangle$. This enables entanglement, a feature necessary for the advantage of quantum computers. As an example of an entangled state, the two-qubit Bell state $|\Phi^+\rangle = (|00\rangle + |11\rangle)/\sqrt{2}$ cannot be written as a product state $|\phi_1\rangle|\phi_2\rangle$.

The linearity of quantum theory implies that a quantum computer can execute classical computations in superposition. For example, if the input state contains all possible input values $|x\rangle$ of a function $f(x)$, the function can be computed using a unitary operation U_f as

$$\sum_x a_x|x\rangle|0\rangle \xrightarrow{U_f} \sum_x a_x|x\rangle|f(x)\rangle. \tag{1}$$

With a single call to U_f, the quantum computer produces a state that contains information about all the possible outputs of $f(x)$.

Nevertheless, quantum computation has several limitations. For example, the no-cloning theorem (3, 4) states that an unknown quantum state cannot be copied perfectly. More importantly, the information of a general quantum state cannot be read out with a single projective measurement

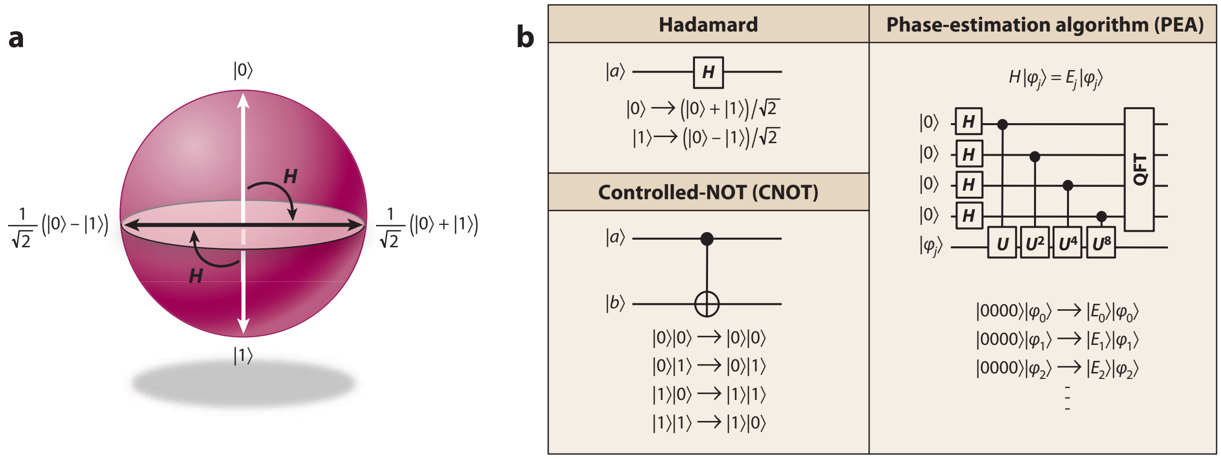

a
$|0\rangle$
$\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$
$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$
H
H
$|1\rangle$
b
Hadamard
$|a\rangle$
H
$|0\rangle \rightarrow (|0\rangle + |1\rangle)/\sqrt{2}$
$|1\rangle \rightarrow (|0\rangle - |1\rangle)/\sqrt{2}$
Controlled-NOT (CNOT)
$|a\rangle$
$|b\rangle$
$|0\rangle|0\rangle \rightarrow |0\rangle|0\rangle$
$|0\rangle|1\rangle \rightarrow |0\rangle|1\rangle$
$|1\rangle|0\rangle \rightarrow |1\rangle|1\rangle$
$|1\rangle|1\rangle \rightarrow |1\rangle|0\rangle$
Phase-estimation algorithm (PEA)
$H|\varphi_j\rangle = E_j|\varphi_j\rangle$
$|0\rangle$
$|0\rangle$
$|0\rangle$
$|0\rangle$
$|\varphi_j\rangle$
H
H
H
H
U
U^2
U^4
U^8
QFT
$|0000\rangle|\varphi_0\rangle \rightarrow |E_0\rangle|\varphi_0\rangle$
$|0000\rangle|\varphi_1\rangle \rightarrow |E_1\rangle|\varphi_1\rangle$
$|0000\rangle|\varphi_2\rangle \rightarrow |E_2\rangle|\varphi_2\rangle$
⋮

Figure 1

Qubit, elementary gates, and phase-estimation algorithm. (*a*) The quantum state of a qubit can be represented on a Bloch sphere. (*b*) The action of the Hadamard gate H on a qubit is shown in panel *a*. The controlled-NOT (CNOT) gate together with single qubit gates form a universal gate set. The quantum circuit for the phase-estimation algorithm (PEA) is shown on the right. Here $U\,|\phi_k\rangle = e^{2\pi i E_k}|\phi_k\rangle$ and QFT is the quantum Fourier transform. The eigenvalues in this case are evaluated to four-digit accuracy.

because that would collapse a superposition into one of its components. Therefore, although the state in Equation 1 contains information about all possible outputs, that information is not immediately accessible. Instead, a quantum algorithm has to be designed in a way that makes it easy to measure a global property of f, without making it necessary to compute all the individual $f(x)$. Algorithms of this kind are discussed in the following sections.

2.2. Approaches to Quantum Computing

There are several models of, or ways of formulating, quantum computation. Most work in quantum simulation has been done in the circuit and adiabatic models. Although the two are known to be computationally equivalent—any computation that can be performed in one model can performed in the other in a comparable amount of time (5–7)—different problems are solved more naturally in different models. We discuss the two models in turn but note that other models hold promise for the development of future simulation algorithms, including topological quantum computing (8, 9), one-way quantum computing (10, 11), and quantum walks (12).

2.2.1. The circuit model. The cornerstone of quantum computation is the generalization of the classical circuit model, composed of classical bits and logical gates, to a quantum circuit model (4, 13, 14). A quantum circuit is a multiqubit unitary transformation U, which maps a set of initial states to some final states. Usually, a unitary gate is decomposed into elementary gates that involve a few (one or two) qubits each.

In classical computing, the NAND gate is universal (15), meaning that any logical circuit can be constructed using NAND gates only. Similarly, in quantum computing, there are sets of unitary operations that form universal quantum gate sets. A quantum computer that can implement such a set is called universal and can perform any unitary transformation U to an arbitrary accuracy. It turns out that the set containing all single-qubit gates in addition to any two-qubit entangling gate, such as CNOT, is universal (3) (**Figure 1**). An entangling gate can be realized by any

Universal quantum gate set: a set of quantum gates that can approximate any quantum circuit physical interaction that can generate entanglement between qubits. Examples of experimental implementations of quantum gates have been reviewed (16), and we cover some of the experiments relevant to quantum simulation in Section 5.

QFT: quantum Fourier transform

Efficient algorithm: running in polynomial time

P: polynomial-time complexity class (easy to solve)

NP: nondeterministic polynomial-time complexity class (easy to verify solution)

Easy problem: solvable in polynomial time

Besides the elementary gates, an important quantum transformation is the quantum Fourier transform (QFT). It transforms any quantum state $|\varphi\rangle = \sum_x \varphi(x)|x\rangle$ into its Fourier representation,

$$U_{QFT}|\varphi\rangle = \sum_{k=0}^{N-1} \tilde{\varphi}(k)|k\rangle, \tag{2}$$

where $\tilde{\varphi}(k) = (1/\sqrt{N})\sum_{x=0}^{N-1}\varphi(x)e^{2\pi ikx/N}$ are the discrete Fourier coefficients of $\varphi(x)$. The QFT can be efficiently implemented using a quantum circuit (4): For n qubits, the number of elementary gates required is $O(n^2)$. For comparison, the classical fast Fourier transform requires $O(n2^n)$ gates. We take advantage of the QFT in Section 3.2 for the simulation of quantum dynamics and in Section 3.3 for the measurement of observables.

2.2.2. Adiabatic quantum computation.

An alternative to the gate model is the adiabatic model of quantum computation (17). In this model, the quantum computer remains in its ground state throughout the computation. The Hamiltonian $H(t)$ of the computer is changed slowly from a simple initial Hamiltonian H_i to a final Hamiltonian H_f whose ground state encodes the solution to the computational problem. The adiabatic theorem states that if the variation of the Hamiltonian is sufficiently slow, the easy-to-prepare ground state of H_i will be transformed continuously into the ground state of H_f. It is desirable to complete the evolution as quickly as possible; the maximum rate of change is mostly determined by the energy gap between the ground and first excited states during the evolution (18–21). The applications of AQC to simulation include preparing quantum states of interest and solving optimization problems such as protein folding (22). We discuss the details in Sections 3.4.1 and 4, respectively.

2.3. Quantum Complexity Theory

To understand the computational advantages of quantum algorithms for chemical simulation, we discuss some aspects of computational complexity theory, which defines quantum speedup unambiguously. A proper measure of the complexity of an algorithm is how many operations (or how much time) it takes to solve problems of increasing size. Conventionally, a computational problem is described as easy or tractable if there exists an efficient algorithm for solving it, one that scales polynomially with input size [for an input of size n, as $O(n^k)$ for some k]. Otherwise, the problem is hard. This distinction is admittedly a rough one: For reasonable problem sizes, an inefficient algorithm scaling exponentially as $O(1.0001^n)$ would be faster than an efficient $O(n^{999})$ algorithm. Nevertheless, this convention has proven useful because, in practice, polynomially scaling algorithms generally outperform exponential ones.

The class of all problems that are easy for classical computers (classical Turing machines) is called P (23). (Strictly speaking, these are decision problems, those with a yes-or-no answer. However, other problems can be recast as decision problems; for example, instead of asking "What is the ground-state energy of molecule X?," we might ask "Is the ground-state energy of X less than E_{guess}?") The class of all problems whose answer can be verified in polynomial time is NP. For example, even though we do not know how to factor numbers efficiently, factoring is in NP because we can check the proposed answer efficiently by multiplication. We note that P $\subseteq$ NP because a solution to an easy problem can be verified easily. Whether P $=$ NP is a famously open question;

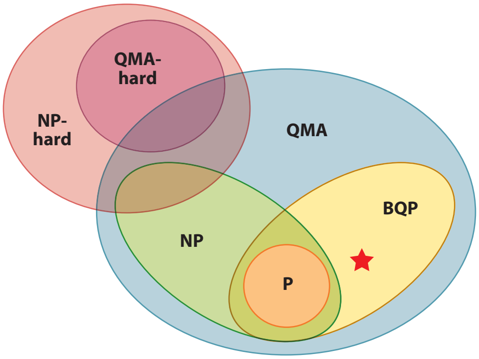


Figure 2

The conjectured relationships between the computational complexity classes discussed in this review. Simulating the time evolution of chemical systems (denoted by the star) is in BQP but widely believed to be outside of P (assuming a constant error and simulated time). That is, it is easy on quantum computers, but probably hard—even in principle—on conventional ones.

however, it is widely believed that they are not equal, that is, that there are problems in NP that cannot be solved easily (24). The hardest among them belong to the class NP-hard: if any NP-hard problem can be solved efficiently, then so can any problem in NP.

The class of problems that are easy for a quantum computer is known as BQP (25). The quantum analog of NP is called QMA and contains those problems easy to check on a quantum computer. In analogy with NP-hard problems, QMA-hard contains the hardest problems in QMA. Shor's (26) factoring algorithm is significant because it provides an example of a problem in BQP that is widely thought (although not proven) to be outside of P, that is, a problem believed to be hard on classical computers that is easy for a quantum computer.

The relationships between the complexity classes mentioned above are illustrated in **Figure 2**. In the remainder of this review, we explore the advantages of quantum simulation over its classical counterpart partly by situating various simulation tasks in the computational classes illustrated in **Figure 2**.

3. QUANTUM SIMULATION

Quantum simulation schemes can be divided into two broad classes. The first is dedicated quantum simulation, in which one quantum system is engineered to simulate another quantum system. For example, quantum gases in optical lattices can be used to simulate superfluidity (27). The other, more general, approach is universal quantum simulation, simulating a quantum system using a universal quantum computer. The terms analog and digital have also been used for dedicated and universal quantum simulation, respectively, to emphasize that quantum error correction is available to the latter (28). Although we focus on universal quantum simulation because most chemical proposals assume a universal quantum computer, we mention dedicated simulators where appropriate.

One of the main goals of quantum simulation is to determine the physical properties of a particular quantum system. This problem can usually be conceptualized as involving three steps: (*a*) Initialize the qubits in a state that can be prepared efficiently, (*b*) apply a unitary evolution to this initial state, and (*c*) read out the desired information from the final state. [Nonunitary open-system dynamics have been studied as well (29).]

We note at the outset that it is not possible to exactly simulate an arbitrary unitary evolution on a quantum computer efficiently. An arbitrary unitary acting on a system of n spins has $2^n \times 2^n$ free

BQP: bounded-error quantum polynomial time complexity class (quantum analog of P)

QMA: quantum Merlin Arthur complexity class (quantum analog of NP)

Dedicated quantum simulation: a quantum system is engineered to simulate another

Universal quantum simulation: a quantum system is simulated by a universal quantum computer parameters and would require an exponential number of elementary quantum gates to implement. However, in quantum chemistry, it usually is not necessary to simulate arbitrary dynamics, as natural systems are not arbitrary (2). Instead, the interactions involved (between molecular orbitals, for example) are local—featuring at most k-body interactions—and this crucial aspect of their structure can be exploited for their efficient simulation. That is, the Hamiltonian generating the unitary evolution is a sum $H = \sum_i H_i$ of polynomially many terms, each of which acts on at most polynomially many degrees of freedom. A local Hamiltonian generates a time evolution that can be decomposed into $t/\delta t$ time steps according to the Lie-Trotter formula,

$$U = e^{-i\sum_i H_i t} \approx \left(\prod_i e^{-iH_i \delta t}\right)^{t/\delta t}. \tag{3}$$

The approximation can be improved by increasing the number of time steps or by using higher-order generalizations of this formula (30, 31). Finally, because each factor $e^{-iH_i\delta t}$ acts on only a subregion of the Hilbert space and therefore can be simulated efficiently, so can a product of polynomially many such factors. Hence, the time it takes to perform the simulation scales polynomially with the simulated time t. Most methods of quantum simulation make use of the Trotter decomposition, and we describe in more detail their applications in chemistry below. We do not discuss all the available methods, for which the reader is directed to comprehensive reviews (32–34).

In the following, we describe two ways in which chemical wave functions can be encoded on a quantum computer, second- and first-quantization approaches (see **Table 1** for a comparison). For each approach, we outline the methods of preparing certain classes of initial states and propagating them in time. Afterward, we discuss the methods of measurement of observables and preparation of ground and thermal states, which do not depend essentially on the way the wave function is encoded.

3.1. Second Quantization

We start by considering the purely electronic molecular problem, in which the BOA has been used to separate the electronic and nuclear motion. The wave function of the electrons can be expanded in an orthonormal basis of M molecular spin-orbitals $\{|\chi_i\rangle\}$. Corresponding to this basis are the fermionic creation and annihilation operators $a_i^\dagger$ and a_i. There is a natural mapping between the electronic Fock space and the state of M qubits: Having qubit i in the state $|0\rangle$ (or $|1\rangle$) indicates that spin-orbital i is unoccupied (or occupied).

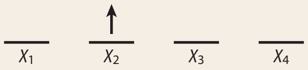
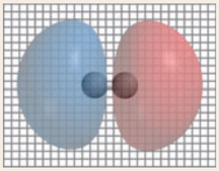
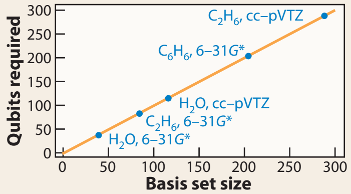
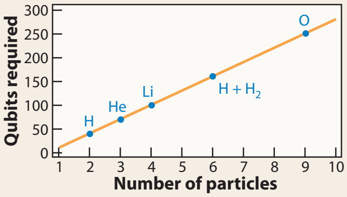
An important subtlety is that electrons in a molecule, unlike the individually addressable qubits, are indistinguishable. In other words, whereas the operators $a_i^\dagger$ and a_i obey the canonical fermionic anticommutation relations, $\{a_i, a_j^\dagger\} = \delta_{ij}$, the qubit operators that change $|0\rangle$ to $|1\rangle$ and vice versa do not. This problem can be solved by using the Jordan-Wigner transformation to enforce the correct commutation relations on the quantum computer (35–38). The Jordan-Wigner transformation for this case results in the following mapping between the fermionic operator algebra and the qubit spin algebra:

$$a_i^\dagger \leftrightarrow 1^{\otimes(i-1)} \otimes \sigma^- \otimes (\sigma^z)^{\otimes(2M-i-1)}, \tag{4a}$$

$$a_i \leftrightarrow 1^{\otimes(i-1)} \otimes \sigma^+ \otimes (\sigma^z)^{\otimes(2M-i-1)}, \tag{4b}$$

where $\sigma^- = |1\rangle\langle 0|$ and $\sigma^+ = |0\rangle\langle 1|$.

Table 1 Comparison of second- and first-quantization approaches to quantum simulation

	Second quantized	First quantized
Wave-function encoding	Fock state in a given basis:  $\lvert\psi\rangle = \lvert 0100\rangle$	On a grid of 2^n sites per dimension: $\lvert\psi\rangle = \sum_x a_x \lvert x\rangle$ 
Qubits required to represent the wave function	One per basis state (spin-orbital):  	3*n* per particle (nuclei and electrons): 
Molecular Hamiltonian	Coefficients precomputed classically: $\sum_{pq} h_{pq} a_p^\dagger a_q + \frac{1}{2} \sum_{pqrs} h_{pqrs} a_p^\dagger a_q^\dagger a_r a_s$	Interaction calculated on the fly: $\sum_i \frac{p_i^2}{2m_i} + \sum_{i<j} \frac{q_i q_j}{r_{ij}}$
Quantum gates required for simulation	$O(M^5)$ with number of basis states	$O(B^2)$ with number of particles
Advantages	• Compact wave-function representation (requires fewer qubits) • Takes advantage of classical electronic-structure theory to improve performance • Already experimentally implemented	• Better asymptotic scaling (requires fewer gates) • Treats dynamics better • Can be used for computing reaction rates or state-to-state transition amplitudes

The electronic Hamiltonian in the second-quantized form is

$$H = \sum_{pq} h_{pq} a_p^\dagger a_q + \frac{1}{2} \sum_{pqrs} h_{pqrs} a_p^\dagger a_q^\dagger a_r a_s, \tag{5}$$

where the spin-orbital indices p, q, r, and s each range from 1 to M. Here, one-electron integrals $h_{pq} \equiv \langle p|(T + V_N)|q\rangle$ involve the electronic kinetic energy T and the nuclear-electron interaction V_N, and the two-electron integrals $h_{pqrs} \equiv \langle pq|V_e|rs\rangle$ contain the electron-electron interaction term V_e. For simulation on a quantum computer, this Hamiltonian is recast into the spin algebra using the Jordan-Wigner transformation (Equation 4), and the time evolution it generates is implemented using the Trotter decomposition (Equation 3). We note that H contains $O(M^4)$ terms, and each of these terms generates a time evolution of the form

$$U_{pq} = e^{-ih_{pq} a_p^\dagger a_q \delta t} \quad \text{or} \quad U_{pqrs} = e^{-ih_{pqrs} a_p^\dagger a_q^\dagger a_r a_s \delta t}. \tag{6}$$

Each of these operators requires $O(M)$ elementary quantum gates to implement because of the Jordan-Wigner transformation. Because there are altogether $O(M^4)$ terms that need to be implemented separately, the total cost of simulating H scales as $O(M^5)$ (39).

Although any basis $\{|\chi_i\rangle\}$ can be chosen to represent H, it is desirable to choose a basis as small as possible that adequately represents the system under study. Electronic-structure experience provides many good starting points, such as the Hartree-Fock (HF) basis or the natural orbitals (40). Regardless of the basis chosen, a lot of the computation can be carried out on classical computers as preprocessing. In particular, the coefficients h_{pq} and h_{pqrs} can be precomputed efficiently on classical computers. That way, only the more computationally demanding tasks are left for the quantum computer.

The HF basis allows the use of the HF reference state as an input to the quantum computation (35). A salient feature is that such states are Fock states, which are easy to prepare on the quantum computer: Some qubits are initialized to $|0\rangle$ and others to $|1\rangle$. In fact, any single-determinant state can be prepared easily in this way. Furthermore, it is possible to prepare superpositions of Fock basis states as inputs for the quantum computation. Although an arbitrary state might be difficult to prepare, many states of interest, including those with only polynomially many determinant contributions, can be prepared efficiently (36–38, 41). The problem of preparing an initial state that is close to the true molecular ground state is addressed in Section 3.4.

The chief advantage of the second-quantization method is that it is frugal with quantum resources: Only one qubit per basis state is required, and the integrals can be precomputed classically. For this reason, the first chemical quantum computation was carried out in second quantization (see Section 5). Nevertheless, there are processes, such as chemical reactions, that are difficult to describe in a small, fixed basis set, and for this we turn to a discussion of first-quantization methods.

3.2. First Quantization

The first-quantization method, due to Zalka (42–44), simulates particles governed by the Schrödinger equation on a grid in real space. (We also note the method of Reference 45, which in our terminology is a hybrid between second- and first-quantization methods. It associates a qubit to the occupation of each lattice site.) For a single particle in one dimension, space is discretized into 2^n points, which, when represented using n qubits, range from $|0\ldots0\rangle$ to $|1\ldots1\rangle$. The particle's wave function can be expanded in this position representation as $|\psi\rangle = \sum_{x=0}^{2^n-1} a_x|x\rangle$. The Hamiltonian to be simulated is

$$H = T + V = \frac{p^2}{2m} + V(x), \tag{7}$$

and the resulting unitary can be implemented using the quantum version of the split-operator method (46, 47):

$$U(\delta t) = e^{-i(T+V)\delta t} \approx U_{\text{QFT}} e^{-iT\delta t} U_{\text{QFT}}^{\dagger} e^{-iV\delta t}. \tag{8}$$

The operators $e^{-iV\delta t}$ and $e^{-iT\delta t}$ are diagonal in the position and momentum representations, respectively. A diagonal operator can be implemented easily because it amounts to adding a phase $e^{-iV(x)\delta t}$ to each basis state $|x\rangle$. Furthermore, it is easy on a quantum computer to switch between the position and momentum representations of a wave function using the efficient QFT. Therefore, simulating a time evolution for time t involves alternately applying $e^{-iV\delta t}$ and $e^{-iT\delta t}$ with the time steps δt chosen to be sufficiently short to secure a desired accuracy. Finally, the scheme can be generalized easily to many particles in three dimensions: A system of B particles requires $3Bn$ qubits, n for each degree of freedom.

The first-quantization method can be applied to many problems. The earliest applications established that as few as 10–15 qubits would be needed for a proof-of-principle demonstration of single-particle dynamics (48) [later improved to 6–10 (49)]. The method could also be used to study faithfully the chaotic dynamics of the kicked-rotor model (50). The first chemical application was the proposal of a method for the calculation of the thermal rate constant (51) (see Section 3.3).

We investigated the applicability of the first-quantization method to the simulation of chemical dynamics (42). The simplest approach is to consider all the nuclei and electrons explicitly, in which case the exact nonrelativistic molecular Hamiltonian reads

$$H = \sum_i \frac{p_i^2}{2m_i} + \sum_{i<j} \frac{q_i q_j}{r_{ij}}, \tag{9}$$

where $r_{ij} \equiv |\mathbf{r}_i - \mathbf{r}_j|$ is the distance between particles i and j, which carry charges q_i and q_j, respectively. As before, the split-operator method can be used to separate the unitaries that are diagonal in the position and momentum bases. We note that a Jordan-Wigner transformation is not required; H preserves permutational symmetry, meaning that if the initial state is properly (anti)symmetrized (see below), it will stay so throughout the simulation.

As the Born-Oppenheimer approximation (BOA) has been used widely in quantum chemistry, it might seem extravagant to simulate explicitly all the nuclei and electrons. Nevertheless, the exact simulation in fact is faster than using the BOA for reactions with more than approximately four atoms (42). The reason for this is the need to evaluate the potential $V(\mathbf{r}_1, \ldots, \mathbf{r}_B)$ on the fly on the quantum computer. In the exact case, the potential is simply the pairwise Coulomb interaction; by contrast, evaluating the complicated, many-body potential energy surfaces that are supplied by the BOA is a much more daunting task, even considering that one can use nuclear time steps that are approximately 1,000 times longer. That is, exact simulation minimizes arithmetic, which is the bottleneck of the quantum computation; by contrast, the bottleneck on classical computers is the prohibitive scaling of the Hilbert space size, which is alleviated by the BOA.

In order to carry out simulations, it is important to prepare suitable initial states. Zalka's (44) original paper contained a general state-preparation scheme, later rediscovered (52–54) and improved (55). The scheme builds the state one qubit at a time by performing a rotation (dependent on the previous qubits) that redistributes the wave-function amplitude as desired. For example, Gaussian wave packets or molecular orbitals can be constructed efficiently. We discussed how to combine such single-particle wave functions into many-particle Slater determinants, superpositions of determinants, and mixed states in Reference 56. In particular, the (anti)symmetrization algorithm of Reference 57 was improved and used to prepare Slater determinants necessary for chemical simulation. Furthermore, we outlined a procedure for translating states that are prepared in second-quantization language into first-quantized wave functions, and vice versa. Techniques for preparing ground and thermal initial states are discussed in Section 3.4.

The first-quantization approach to quantum simulation suffers from the fact that even the simplest simulations might require dozens of qubits and millions of quantum gates (42). Nevertheless, it has advantages that would make it useful if large quantum computers are built. Most importantly, because the Coulomb interaction is pairwise, simulating a system of B particles requires $O(B^2)$ gates, a significant asymptotic improvement over the second-quantized scaling of $O(M^5)$, where M is the size of the basis set.

3.3. Measuring Observables

Above we discuss how to prepare and evolve quantum states on a quantum computer. Information about the resulting state must be extracted in the end; however, full characterization (quantum state tomography) generally requires resources that scale exponentially with the number of qubits. This is because a measurement projects a state into one consistent with the measurement outcome.

Because only a limited amount of information can be extracted efficiently, one needs a specialized measurement scheme to extract the desired observables, such as dipole moments and correlation functions.

PEA: phase-estimation algorithm

In principle, an individual measurement can be carried out in any basis. However, as experimental measurement techniques usually address individual qubits, a method is needed to carry out more complicated measurements. In particular, in order to measure an observable A, one would like to carry out a measurement in its eigenbasis $\{|e_k\rangle\}$. This is achieved by the phase-estimation algorithm (PEA) (**Figure 1**) (58, 59, 139):

$$\frac{1}{\sqrt{N}}\sum_t |t\rangle|\psi\rangle \xrightarrow{C-U} \frac{1}{\sqrt{N}}\sum_{k,t} c_k e^{-iA_k t}|t\rangle|e_k\rangle \xrightarrow{QFT} \sum_k c_k|A_k\rangle|e_k\rangle, \tag{10}$$

where $c_k = \langle e_k|\psi\rangle$ and A_k are the eigenvalues of A; $C-U$ is the unitary $U = \exp(-iAt)$ controlled by the ancilla qubits, which are initialized in the state $(1/\sqrt{N})\sum_t |t\rangle$. When measuring the ancilla, the eigenvalue A_k will be measured with probability $|c_k|^2$ and, if the eigenstates are nondegenerate, the wave function will collapse to the eigenvector $|e_k\rangle$. [Other methods for eigenvalue measurement include pairing adiabatic quantum evolution with Kitaev's original scheme (60) and applications of the Hellmann-Feynman theorem (61).] For the PEA to be efficient, it must be possible to simulate the pseudodynamics e^{-iAt} efficiently. In particular, if we are interested in molecular energies, the observable is the Hamiltonian H, and we need to simulate the actual dynamics e^{-iHt} (see Section 3). We note that the PEA is closely related to classical algorithms for preparing eigenstates by Fourier analysis of a propagating system (62, 63). As in classical Fourier analysis, the (pseudo)dynamics must be simulated for longer times to achieve a higher precision in the A_k. More precisely, for a final accuracy of ε, the PEA must run for a time $O(1/\varepsilon)$ (4, 32).

Because quantum measurement is inherently random, repeating a measurement on multiple copies of the same system helps to determine expectation values of observables. The central limit theorem implies that measuring N copies of a state results in a precision that scales as $1/\sqrt{N}$ [the standard quantum limit (SQL)]. For example, repeating the PEA gives an SQL estimate of the coefficients $|c_k|^2$; these can be used to calculate the expectation value $\langle A\rangle = \sum_k |c_k|^2 A_k$, also to the SQL. When entanglement is available, one can achieve precision scaling as $1/N$—this is the Heisenberg or quantum metrology limit (64). An algorithm for the expectation values of observables has been proposed that can get arbitrarily close to the Heisenberg limit (65).

The first algorithm for measuring a chemical observable was Lidar & Wang's (51) calculation of the thermal rate constant by simulating a reaction in first quantization and using the PEA to obtain the energy spectrum and the eigenstates. These values were used to calculate the rate constant on a classical computer by integrating the flux-flux correlation function. This method has been improved with a more direct approach to the rate constant (42) that efficiently obtains the product branching ratios given different reactant states—if the initial state is a thermal state (see Section 3.4.2), this gives the rate constant directly. Furthermore, the method was used to obtain the entire state-to-state scattering matrix. A method for reaction rates using a dedicated quantum simulator in which artificial molecules are manipulated experimentally also has been proposed (66).

More generally, correlation functions provide information about a system's transport and spectroscopic properties. On a quantum computer, the correlation function of any two observables can be estimated efficiently if their pseudodynamics each can be simulated efficiently (36, 38). The method does not suffer from the dynamic sign problem that plagues classical Monte Carlo methods for sampling correlation functions. An alternative approach is the measurement of correlation functions using techniques of linear-response theory (67).

Molecular properties such as the dipole moment or the static polarizability are also of chemical interest. They are derivatives of the molecular energy with respect to an external parameter, such as the electric field. We showed how to calculate them (68) using the PEA and the quantum gradient algorithm (69). The algorithm is insensitive to the dimensionality of the derivatives, an obstacle to classical computers. For example, the molecular gradient and Hessian can be computed—and used to optimize the geometry—with a number of energy evaluations independent of system size.

3.4. Preparing Ground States and Thermal States

In Sections 3.1 and 3.2, we discuss the preparation of various initial states for quantum simulation. We now turn to the discussion of the preparation of ground and thermal states.

3.4.1. Ground-state preparation by phase estimation. A large part of quantum chemistry is concerned with the calculation of ground-state properties of molecules, making it desirable to prepare such states on a quantum computer. In the previous section, we describe how the PEA can be used to measure a quantum state in the eigenbasis of a Hermitian operator. This suggests a method for preparing a ground state: Measuring in the eigenbasis of the Hamiltonian will project a state $|\psi\rangle$ to the ground state $|g\rangle$ with probability $|\langle\psi|g\rangle|^2$.

The problem, therefore, is to prepare a state close to the ground state, from which we can project the ground-state component. Choosing a random state $|\psi_{\text{rand}}\rangle$ is bound to fail, as the overlap is expected to be exponentially small in the number of qubits n: $\langle\psi_{\text{rand}}|g\rangle \sim 2^{-n}$. This means that one would have to repeat the PEA exponentially many times before chancing upon the ground state.

Methods of quantum chemistry can be used to improve the overlap. We studied the ground-state preparation of H_2O and LiH in second quantization, based on the HF approximation (35). The goal was to prepare the ground state of the FCI Hamiltonian, so that its energy could be read out by the PEA, thus solving the electronic-structure problem. Because these molecules were considered at equilibrium geometries, the HF guess was sufficient for the algorithm to estimate the ground-state energies of these molecules with high probability. The overlap can be improved further by choosing a more sophisticated approximation method such as a multiconfiguration self-consistent field wave function (70).

Alternatively, the overlap can be increased using AQC (Section 2.2.2). We applied adiabatic-state preparation (ASP) to the case of the hydrogen molecule H_2 in the STO-3G basis at various bond lengths (35). As the bond length increases, the HF state has decreasing overlap with the exact state, reaching 0.5 at large separations. ASP works by preparing the ground state of the HF Hamiltonian and then slowly changing to the FCI Hamiltonian. The speed of the variation of the Hamiltonian is limited by the energy gap between the ground state and the first excited state. In the case of H_2, this method allowed the preparation of the FCI ground state with a high fidelity.

Procedures similar to ASP have been proposed to study low-energy states of some toy models in physics (61) and superconductivity (71). It is also possible to encode a thermal state into the ground state of a Hamiltonian (72, 73), offering a way to prepare a thermal state, a problem further discussed in the next section.

3.4.2. Thermal-state preparation Although not often a subject of quantum-chemical calculations, the thermal states are significant because they can be used to solve many problems, ranging from statistical mechanics to the calculation of thermal rate constants. Classical algorithms typically rely on Markov chain Monte Carlo methods, which sample from the Gibbs density matrix, $\rho = e^{-\beta H}/Z$, where Z is the partition function. The challenge is that it is generally impossible to sample from the eigenstates $|e_k\rangle$ of a certain Hamiltonian H if they are not predetermined (which is often more challenging).

With a quantum computer, assuming the PEA can be implemented efficiently, we can prepare the thermal state of any classical or quantum Hamiltonian from a Markov chain constructed by repeating a completely positive map (67). A limitation of that approach is that the Metropolis step can make too many transitions between states of very different energies, sometimes leading to a slow convergence rate of the resulting Markov chain. This issue was addressed by building up the Markov chain by applying random local unitary operations (74). The resulting operation is a Metropolis-type sampling for quantum states; although the underlying Markov chain is classical in nature, performing it on a quantum computer provides the benefit of being able to use the PEA without explicitly solving the eigenvalue equations. However, quantum computers can implement Markov chains corresponding to thermal states of classical Hamiltonians with a quadratic speedup (75–80).

Zalka's state preparation algorithm (see Section 3.2) is applicable to preparing the coherent encoding of thermal states (CETS) $|\psi_{\text{th}}\rangle$,

$$|\psi_{\text{th}}\rangle = \sum_k \sqrt{e^{-\beta E_k}/Z}|e_k\rangle|e_k\rangle, \tag{11}$$

which is equivalent to the Gibbs density matrix, $\rho_{\text{th}} = \sum_k e^{-\beta E_k}/Z|e_k\rangle\langle e_k|$, if one register is traced out. If the eigenstates and eigenvalues are known, it is possible to construct the CETS directly (56). When they are unknown, by combining ideas from belief-propagation (81) and quantum amplitude amplification (3), we were able to construct the CETS of classical Hamiltonians with a quadratic quantum speedup (82).

Lastly, a thermal state can be prepared by modeling the physical interaction with a heat bath (44, 67). However, the computational cost of these methods is not well understood.

3.4.3. QMA hardness and future prospects. Unfortunately, the procedures for ground- and thermal-state preparation outlined above are not fully scalable to larger systems. A simple way to see this is to imagine a system composed of N identical, noninteracting molecules. Even if one molecule can be prepared with a ground-state overlap of $1 - \varepsilon$ by any method, the fidelity of the N-molecule state will be exponentially small, $(1 - \varepsilon)^N$ (83). ASP would fail when the energy gap became so small that the Hamiltonian would have to be varied exponentially slowly.

More generally, there are broad classes of Hamiltonians for which finding the ground-state energy (and therefore also a thermal state) is known to be QMA-hard, that is, most likely hard even on a quantum computer (see Section 2.3) (6, 84–87). Nevertheless, the scaling of the ground- and thermal-state energy problems for chemical systems on a quantum computer is an open question. It is possible that algorithms can be found that are not efficient for all QMA-hard Hamiltonians but nevertheless succeed for chemical problems.

4. OPTIMIZATION WITH ADIABATIC QUANTUM SIMULATION

We describe the use of quantum computers to solve classical optimization problems related to chemistry and biology. This class of problems plays an important role in fields such as drug design, molecular recognition, geometry optimization, and protein folding (88, 89).

Of all the models of quantum computation, AQC is perhaps the best suited for dealing with discrete optimization problems. As explained in Section 2.2.2, the essential idea behind AQC is to encode the solution to a computational problem in a (final) Hamiltonian ground state that is prepared adiabatically.

Although final Hamiltonians have been proposed for various problems related to computer science (17, 90–93), only recently were constructions (22) derived for problems of chemical interest such as the lattice heteropolymer problem (94–96), an NP-hard problem (97). It can be used as a model of protein folding (98), one of the cornerstones of biophysics. We note that the quantum-computational implementation of the protein folding problem does not assume that the protein is treated quantum mechanically. Instead, the quantum computer is being used as a tool to solve the classical optimization problem. In the lattice folding problem, the sequence of amino acids is coarse-grained to a sequence of beads (amino acids) connected by strings (peptide bond). This chain of beads occupies points on a two- or three-dimensional lattice; a valid configuration (fold) is a self-avoiding walk on the lattice, and its energy is determined by the interaction energies among amino acids that are nonbonded nearest neighbors in the lattice. The hydrophobic-polar model (99) is the simplest realization of this problem. The amino acids are broken into two groups, hydrophobic (H) and polar (P). Whenever two nonbonded hydrophobic amino acids are nearest neighbors in the lattice, the free energy of the protein is reduced by one unit of energy, $E_{HH} = -1$. The remaining interactions do not contribute to the free energy $E_{HP} = E_{PP} = 0$. The lattice folding problem consists of finding one or more folds that minimize the free energy of the protein. By the thermodynamic hypothesis (100), such fold(s) correspond to the conformation of the native conformation(s) of the protein.

The theory behind the quantum-computational implementation of lattice folding is guided by the proposed quantum adiabatic platform on superconducting qubits (101). This scheme is designed to find solutions to the problem

$$E(\mathrm{s}) = -\sum_{i} h_i s_i + \sum_{j>i} J_{ij} s_i s_j, \tag{12}$$

where $|h_i| \leq 1$, $|J_{ij}| \leq 1$, and $s_i = \pm 1$. Given a set of $\{h_{|i}\}$ and the interaction matrix $\{J_{ij}\}$, the goal is to find the assignment $\mathbf{s}^* = s_1^* s_2^* \cdots s_N^*$ that minimizes $E(\mathbf{s})$.

The time-dependent Hamiltonian is chosen to be

$$H(\tau) = A(\tau) H_i + B(\tau) H_f, \quad \tau = t/t_{\mathrm{run}}, \tag{13}$$

where H_i has a simple-to-prepare ground state and $H_f = -\sum_i h_i \sigma_z^i + \sum_{j>i} J_{ij} \sigma_z^i \sigma_z^j$, where σ_z^i denotes the Pauli matrix acting on the i-th qubit, and t_{run} is the running time. The time-dependent functions $A(\tau)$ and $B(\tau)$ are such that $A(0) \gg B(0)$ and $A(1) \ll B(1)$. Therefore, at the beginning (end) of the simulation, the ground state corresponds to the ground state of H_i (H_f). As desired, $|\mathbf{s}^*\rangle \equiv |s_1^*, s_2^*, \ldots, s_N^*\rangle$ is the ground state of H_f. Measurement of this final state provides the solution to our problem.

The theoretical challenge is to map the lattice folding free-energy function into the form of Equation 12 (22, 102). In two dimensions, we use two binary variables determining the direction of each bond between two amino acids (beads). If a particular bond points upwards, we write 11; if it points downwards, leftwards, or rightwards, we write 00, 10, or 01, respectively. For an N–amino acid protein, we need two binary variables for each of the $N-1$ bonds. Fixing the direction of the first bond reduces the number of variables to $\ell = 2(N-2)$ binary variables. Any possible N-bead fold can be represented by the string of binary variables of the form $01 q_1 q_2 \cdots q_{\ell-1} q_\ell$, where we set the direction of the first bond to be right (01).

As an example, the free-energy function (102) associated with the folding of a four–amino acid peptide assisted by a chaperone protein (see **Figure 3**) is

$$\begin{aligned} E(q) = 4 &- 3q_1 + 4q_2 - 4q_1 q_2 - q_3 + q_1 q_3 - 2q_2 q_3 + 4q_4 - 2q_1 q_4 \\ &- 8q_2 q_4 + 5q_1 q_2 q_4 - 2q_3 q_4 + 5q_2 q_3 q_4 - q_1 q_2 q_3 q_4. \end{aligned} \tag{14}$$

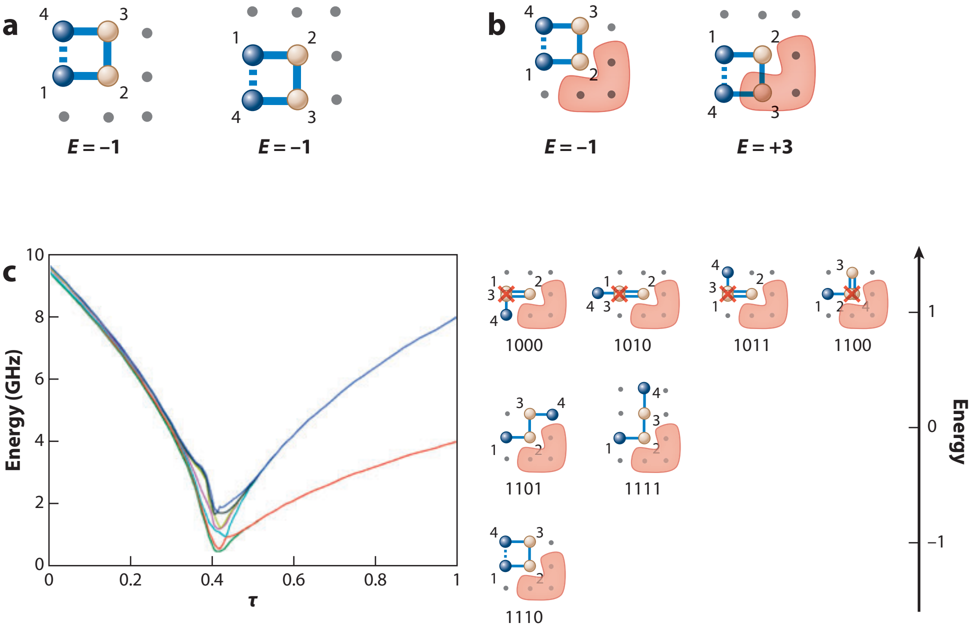


Figure 3

Lattice folding in the hydrophobic-polar model using quantum adiabatic hardware. (*a*) In vacuum, the four–amino acid peptide HPPH could fold clockwise or counterclockwise, depending on whether the third amino acid moves downward or upward, respectively. The presence of two hydrophobic amino acids (1 and 4) on two adjacent sites lowers the energy by one unit. (*b*) In vivo folding of proteins involves the assistance of molecular chaperone proteins whose main function is to assist the folding of the newly synthesized polypeptide. The chaperone molecule (*pink region*) obstructs the third amino acid from moving downward, making the counterclockwise fold the only global minimum in the energy landscape. An overlap of an amino acid with the chaperone increases the energy by four units. (*c*, *left panel*) Time-dependent spectrum showing the eight lowest-energy eigenvalues for the Hamiltonian in Equation 13, as implemented using eight-qubit hardware in Reference 103. (*Right panel*) Corresponding energy landscape (Equation 14) up to $E = 1$. Each overlap of two amino acids (indicated by a *red x*) raises the free energy by two units. The four binary variables encode the direction of the third and fourth bonds of the peptide. The quantum adiabatic hardware found the correct solution 80.3% of the time (103).

By substituting values for the four binary variables defining the directions of the second (q_1q_2) and third (q_3q_4) bonds, we can verify that the 16 assignments provide the desired energy spectrum (**Figure 3**). Equation 14 is not in the form of Equation 12. We converted this energy function from its quartic form to a quadratic form, using two extra ancilla binary variables (102). After the substitution $q_i \equiv \frac{1}{2}(1 - s_i)$, the free-energy function now resembles that of Equation 12. An early experimental realization is described in Section 5.

Solving the hydrophobic-polar model is NP-hard (97, 104, 105). AQC is equivalent to the circuit model, so it is likely unable to solve NP-hard problems in polynomial time (see Section 2.3). Real-world problems (and the instances defining biologically relevant proteins) are not necessarily structureless. Taking advantage of the structure of or information about a particular problem instance is one of the ideas behind new algorithmic strategies (106–108). An example is to introduce heuristic strategies for AQC by initializing the calculation with an educated guess (108).

5. EXPERIMENTAL PROGRESS

Experimental quantum simulation has progressed rapidly (28, 32) since the early simulation of quantum oscillators using nuclear magnetic resonance (NMR) (109). Here we review the proof-of-principle chemical applications of available quantum-computational devices.

5.1. Quantum Optics

On an optical quantum computer, various degrees of freedom of single photons, such as polarization or path, are used to encode quantum information (110, 111). This architecture was used for the first quantum simulation of a molecular system, a minimal-basis model of the hydrogen molecule H_2 (112). Qubits were encoded in photon polarization, whereas two-qubit gates were implemented probabilistically using linear-optical elements and projective measurement. The minimal-basis description of H_2 used two spin-orbitals per atom. Because the FCI Hamiltonian is block-diagonal with 2 × 2 blocks, two qubits sufficed for the experiment: one for storing the system wave function and one for the readout of the PEA. The PEA was implemented iteratively, extracting one bit of the value of the energy at a time. Twenty bits of the energy were obtained, and the answer was exact within the basis set. **Figure 4** describes the experiment and the potential energy surfaces that were obtained.

5.2. Nuclear Magnetic Resonance

Nuclear spins can serve as qubits, being addressed and read out using an NMR spectrometer (113). The first experimental quantum simulation, of a harmonic oscillator, was performed using NMR (109). The platform has since been used to simulate a number of model systems (114–117), leading up to the recent simulation of H_2 (118). The H_2 experiment used ^{13}C-labeled chloroform, in which the carbon and hydrogen nuclear spins form two qubits. The experiment achieved 45 bits of precision (15 iterations of PEA, 3 bits per iteration) in the ground-state energy. ASP (Section 3.4) was implemented for various bond distances.

5.3. Superconducting Systems

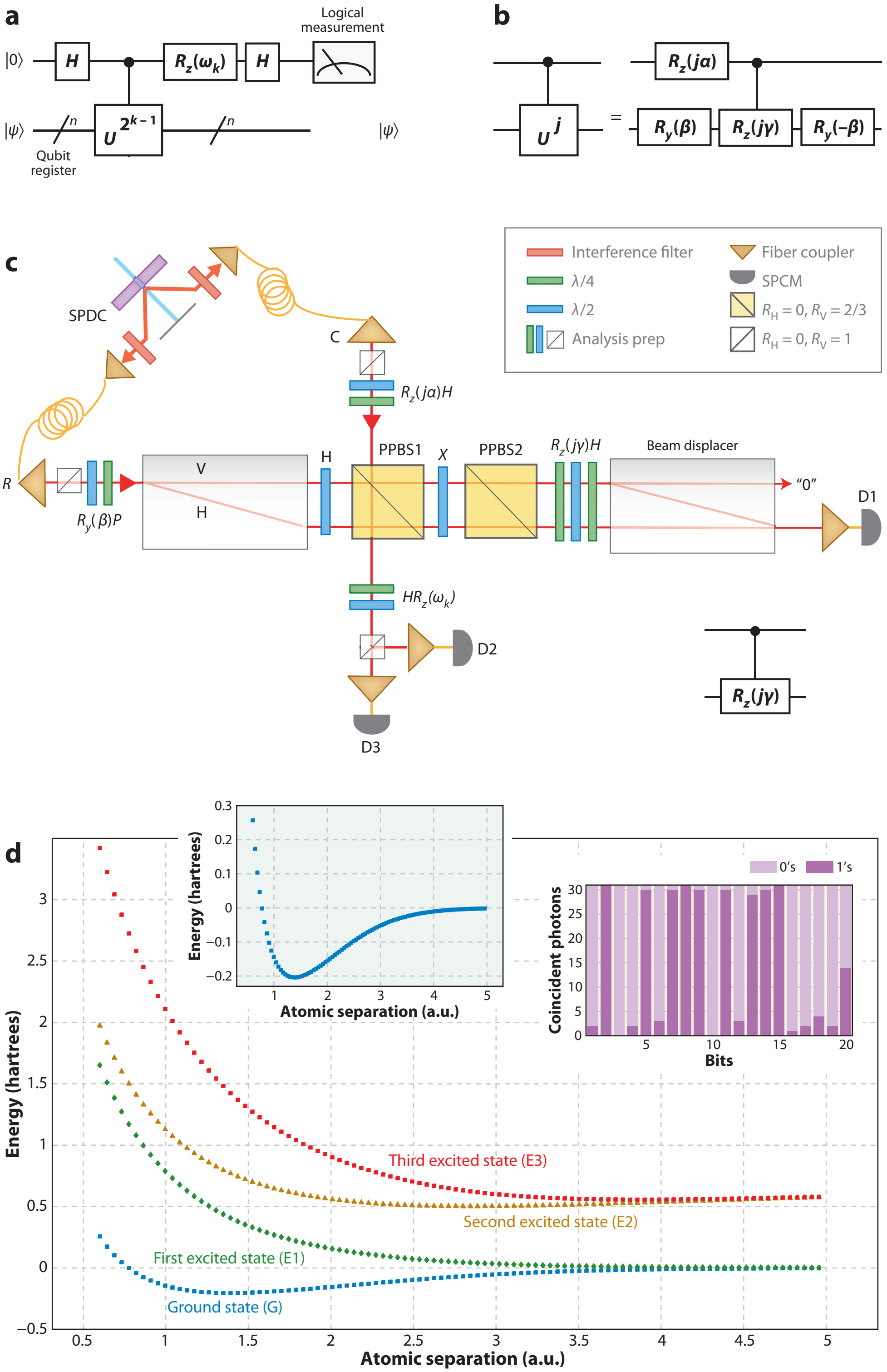
The circulating current (clockwise or counterclockwise) flowing in a micrometer-sized loop of a superconductor can be used as a qubit (119, 120). Examples of applications based on superconducting qubits include the tailor-made generation of bosonic harmonic oscillator states (121) and the implementation of the Deutsch-Jozsa and Grover quantum search algorithms (122). Recently, the free-energy function discussed in Section 4 for the four–amino acid peptide assisted by a chaperone protein (see **Figure 3**) has been realized experimentally (103). A microprocessor consisting of an array of coupled superconductor qubits has been used to implement the time-dependent Hamiltonian in Equation 13, with $H_i \propto \sum_i \sigma_x^i$ as the initial Hamiltonian (101, 123, 124). The quantum hardware operating at a temperature of 20 mK found the correct solution with a probability of 80.3%. Characterization of this device is currently underway (125–128).

5.4. Trapped Ions

Qubits can also be encoded in the electronic states of cold trapped ions, offering one of the most controllable systems available today (129–131). This platform has already produced sophisticated simulations of physical systems (132–134), but chemical applications are still to come.

Figure 4

Experimental simulation of the H_2 molecule on a linear-optical quantum computer (112). (*a*) Two-qubit iterative version of the phase-estimation algorithm for evaluating molecular energies. (*b*) Decomposition of the algorithm into gates. (*c*) The layout of the optical elements used to implement the quantum gates on photonic polarization qubits. (*d*) The computed potential energy surfaces of the H_2 molecule in a minimal-basis set. The results are the exact (in the basis) full-configuration-interaction energies, to 20 bits of precision. Figure adapted from Reference 112.



6. CONCLUSIONS

Above we outline how a quantum computer could be employed for the simulation of chemical systems and their properties, including correlation functions and reaction rates. A method for lattice protein folding also is discussed. Although we focus on the adiabatic and circuit models, these are not the only universal models of quantum computation, and it may be possible to make further algorithmic progress with models such as topological quantum computing (8,9), one-way quantum computing (10, 11), and quantum walks (12, 135, 136).

We also report on the first experiments relevant to chemistry, and we expect more to come in the near future. With recent technological advances, there are many prospects for the future of quantum simulation. However, as more qubits are added to experiments, more effort will be needed to control decoherence, as error correction procedures (4) might not be sufficient in practice because of the spatial and temporal overheads required (137). Instead, it may be possible to build resilient quantum simulators or to incorporate the noise into the simulation.

Although practical quantum computers are not available yet, quantum information theory has already influenced the development of new methods for quantum chemistry. For instance, density matrix renormalization group theory has been extended using quantum information, and its applications to chemistry have been pursued vigorously (138). By studying the simulation of chemical systems on quantum computers, we can also expect new insights into the complexity of computing their properties classically.

Quantum computers are still in an incipient stage, much like classical computers of the vacuum-tube era. Nevertheless, the rapid progress in experimental quantum information science shows no signs of abating, giving many researchers cause for optimism about the technology's future. Larger quantum computers would revolutionize chemistry, by allowing for routine execution of exact, nonadiabatic dynamics simulations, as well as FCI calculations of molecular systems that are intractable with current classical computing technology.

SUMMARY POINTS

1. A universal quantum computer can simulate chemical systems more efficiently (in some cases exponentially so) than a classical computer.
2. Preparing the ground state of an arbitrary Hamiltonian is a QMA-hard problem. Nevertheless, a quantum computer may be able to prepare ground states of certain practically important chemical Hamiltonians, especially when the systems are not strongly correlated.
3. Simulation of quantum dynamics of physical systems is in general efficient with a quantum computer.
4. Properties of quantum states can be obtained by various measurement methods.
5. Classical optimization problems, such as lattice protein folding, can be studied by means of the AQC model.
6. Quantum simulation for chemistry has been realized experimentally in quantum optics, NMR, and superconducting devices.

FUTURE ISSUES

1. Developing quantum simulation methods based on alternative models of quantum computation is an open research direction.

2. Dedicated quantum simulators built so far are mostly for simulating condensed matter systems. It is desirable to make experimental progress on simulating chemical systems.
3. Decoherence is currently the major obstacle for scaling up the current experimental setups. Progress in theoretical and experimental work is needed to overcome decoherence.
4. Methods of quantum error correction, not covered in this review, will be important for large-scale simulations.

DISCLOSURE STATEMENT

The authors are not aware of any affiliations, memberships, funding, or financial holdings that might be perceived as affecting the objectivity of this review.

ACKNOWLEDGEMENTS

We would like to thank DARPA under the Young Faculty Award N66001-09-1-2101-DOD35CAP, the Army Research Office under contract W911-NF-07-0304, the Camille and Henry Dreyfus Foundation, the Sloan Foundation, and the Croucher Foundation (M.-H.J.) for funding. A.A.-G. would also like to thank Peter J. Love for many discussions about quantum computing for quantum chemistry over the years.

LITERATURE CITED

1. **Feynman R. 1982. Simulating physics with computers. *Int. J. Theor. Phys.* 21:467–88**
2. **Lloyd S. 1996. Universal quantum simulators. *Science* 273:1073–78**
3. Kaye P, Laflamme R, Mosca M. 2007. *An Introduction to Quantum Computing*. New York: Oxford Univ. Press
4. **Nielsen MA, Chuang IL. 2000. *Quantum Computation and Quantum Information*. Cambridge, UK: Cambridge Univ. Press**
5. Aharonov D, van Dam W, Kempe J, Landau Z, Lloyd S, Regev O. 2007. Adiabatic quantum computation is equivalent to standard quantum computation. *SIAM J. Comput.* 37:166–94
6. Kempe J, Kitaev A, Regev O. 2006. The complexity of the local Hamiltonian problem. *SIAM J. Comput.* 35:1070–97
7. Mizel A, Lidar DA, Mitchell M. 2007. Simple proof of equivalence between adiabatic quantum computation and the circuit model. *Phys. Rev. Lett.* 99:070502
8. Kitaev A. 2003. Fault-tolerant quantum computation by anyons. *Ann. Phys.* 303:2–30
9. Nayak C, Simon SH, Stern A, Freedman M, Sarma SD. 2008. Non-Abelian anyons and topological quantum computation. *Rev. Mod. Phys.* 80:1083–159
10. Raussendorf R, Briegel HJ. 2001. A one-way quantum computer. *Phys. Rev. Lett.* 86:5188–91
11. Raussendorf R, Browne DE, Briegel HJ. 2003. Measurement-based quantum computation on cluster states. *Phys. Rev. A* 68:022312
12. Kempe J. 2003. Quantum random walks: an introductory overview. *Contemp. Phys.* 44:307
13. Barenco A, Bennett CH, Cleve R, DiVincenzo DP, Margolus N, et al. 1995. Elementary gates for quantum computation. *Phys. Rev. A* 52:3457–67
14. Deutsch D. 1989. Quantum computational networks. *Proc. R. Soc. Lond. A* 425:73–90
15. Null L, Lobur J. 2003. *The Essentials of Computer Organization and Architecture*. Sudbury, MA: Jones & Bartlett
16. **Ladd TD, Jelezko F, Laflamme R, Nakamura Y, Monroe C, O'Brien JL. 2010. Quantum computers. *Nature* 464:45–53**

1. Proposed quantum computers as a means for solving quantum simulation problems.

2. Showed that any local Hamiltonian can be simulated efficiently on a universal quantum computer.

4. The standard and accessible introduction to quantum computation.

16. An accessible review of current progress in experimental quantum computing.

17. **Farhi E, Goldstone J, Gutmann S, Sipser M. 2000. Quantum computation by adiabatic evolution. arXiv:quant-ph/0001106**

17. Introduced the adiabatic model of quantum computation by using the quantum adiabatic theorem to solve optimization problems.

18. Amin MHS. 2009. Consistency of the adiabatic theorem. *Phys. Rev. Lett.* 102:220401
19. Messiah A. 1999. *Quantum Mechanics*. New York: Dover
20. Tong DM. 2010. Quantitative condition is necessary in guaranteeing the validity of the adiabatic approximation. *Phys. Rev. Lett.* 104:120401
21. Tong DM, Singh K, Kwek LC, Oh CH. 2007. Sufficiency criterion for the validity of the adiabatic approximation. *Phys. Rev. Lett.* 98:150402
22. Perdomo A, Truncik C, Tubert-Brohman I, Rose G, Aspuru-Guzik A. 2008. Construction of model Hamiltonians for adiabatic quantum computation and its application to finding low-energy conformations of lattice protein models. *Phys. Rev. A* 78:012320
23. Arora S, Barak B. 2009. *Computational Complexity: A Modern Approach*. Cambridge, UK: Cambridge Univ. Press
24. Aaronson S. 2008. The limits of quantum computers. *Sci. Am.* 298(3):62–69
25. Watrous J. 2009. Quantum computational complexity. In *Encyclopedia of Complexity and System Science*, ed. RA Meyer, pp. 7174–201. Berlin: Springer
26. Shor PW. 1997. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.* 26:1484–509
27. Bakr WS, Peng A, Tai ME, Ma R, Simon J, et al. 2010. Probing the superfluid-to-Mott-insulator transition at the single-atom level. *Science* 329:547–50
28. Buluta I, Nori F. 2009. Quantum simulators. *Science* 326:108–11
29. Bacon D, Childs AM, Chuang IL, Kempe J, Leung DW, Zhou X. 2001. Universal simulation of Markovian quantum dynamics. *Phys. Rev. A* 64:062302
30. Berry D, Ahokas G, Cleve R, Sanders B. 2007. Efficient quantum algorithms for simulating sparse Hamiltonians. *Commun. Math. Phys.* 270:359–71
31. Hatano N, Suzuki M. 2005. Finding exponential product formulas of higher orders. In *Lecture Notes in Physics*, Vol. 679, ed. A Das, BK Chakrabarti, pp. 37–68. Berlin: Springer
32. Brown KL, Munro WJ, Kendon VM. 2010. Using quantum computers for quantum simulation. arXiv:1004.5528
33. Childs AM. 2004. *Quantum information processing in continuous time*. PhD thesis. Mass. Inst. Technol., Cambridge
34. Schack R. 2006. Simulation on a quantum computer. *Inf. Forsch. Entwickl.* 21:21–27
35. **Aspuru-Guzik A, Dutoi AD, Love PJ, Head-Gordon M. 2005. Simulated quantum computation of molecular energies. *Science* 309:1704–7**

35. Proposed a method for solving the electronic-structure problem on a quantum computer.

36. **Ortiz G, Gubernatis JE, Knill E, Laflamme R. 2001. Quantum algorithms for fermionic simulations. *Phys. Rev. A* 64:022319**

36. Introduced the Jordan-Wigner transformation for the quantum simulation of Fermionic systems.

37. Somma R, Ortiz G, Gubernatis JE, Knill E, Laflamme R. 2002. Simulating physical phenomena by quantum networks. *Phys. Rev. A* 65:042323
38. Somma R, Ortiz G, Knill E, Gubernatis J. 2003. Quantum simulations of physics problems. *Int. J. Quantum Inf.* 2:189–206
39. Whitfield J, Biamonte J, Aspuru-Guzik A. 2010. Quantum computing resource estimate of molecular energy simulation. arXiv:1001.3855
40. Davidson E. 1972. Properties and uses of natural orbitals. *Rev. Mod. Phys.* 44:451–64
41. Wang H, Ashhab S, Nori F. 2009. Efficient quantum algorithm for preparing molecular-system-like states on a quantum computer. *Phys. Rev. A* 79:042335
42. **Kassal I, Jordan SP, Love PJ, Mohseni M, Aspuru-Guzik A. 2008. Polynomial-time quantum algorithm for the simulation of chemical dynamics. *Proc. Natl. Acad. Sci. USA* 105:18681–86**

42. Proposed a first-quantization method for the simulation of chemical dynamics on a quantum computer.

43. Wiesner S. 1996. Simulations of many-body quantum systems by a quantum computer. arXiv:quant-ph/9603028
44. **Zalka C. 1998. Simulating quantum systems on a quantum computer. *Proc. R. Soc. A* 454:313–22**

44. Provided a detailed and comprehensive introduction to simulation in first quantization.

45. Boghosian BM, Taylor W. 1998. Simulating quantum mechanics on a quantum computer. *Physica D* 120:30–42

46. Feit M, Fleck J Jr, Steiger A. 1982. Solution of the Schrödinger equation by a spectral method. *J. Comput. Phys.* 47:412–33

47. Kosloff D, Kosloff R. 1983. A Fourier method solution for the time dependent Schrödinger equation as a tool in molecular dynamics. *J. Comput. Phys.* 51:35–53

48. Strini G. 2002. Error sensitivity of a quantum simulator. I: A first example. *Fortschr. Phys.* 50:171–83

49. Benenti G, Strini G. 2008. Quantum simulation of the single-particle Schrödinger equation. *Am. J. Phys.* 76:657–62

50. Lévi B, Georgeot B, Shepelyansky DL. 2003. Quantum computing of quantum chaos in the kicked rotator model. *Phys. Rev. E* 67:046220

51. Lidar DA, Wang H. 1999. Calculating the thermal rate constant with exponential speedup on a quantum computer. *Phys. Rev. E* 59:2429–38

52. Grover L, Rudolph T. 2002. Creating superpositions that correspond to efficiently integrable probability distributions. arXiv:quant-ph/0208112

53. Kaye P, Mosca M. 2004. Quantum networks for generating arbitrary quantum states. arXiv:quant-ph/0407102

54. Kitaev A, Webb WA. 2008. Wavefunction preparation and resampling using a quantum computer. arXiv:0801.0342

55. Soklakov AN, Schack R. 2006. Efficient state preparation for a register of quantum bits. *Phys. Rev. A* 73:012307–13

56. Ward NJ, Kassal I, Aspuru-Guzik A. 2009. Preparation of many-body states for quantum simulation. *J. Chem. Phys.* 130:194105

57. Abrams DS, Lloyd S. 1997. Simulation of many-body Fermi systems on a universal quantum computer. *Phys. Rev. Lett.* 79:2586–89

58. Abrams DS, Lloyd S. 1999. Quantum algorithm providing exponential speed increase for finding eigenvalues and eigenvectors. *Phys. Rev. Lett.* 83:5162–65

59. Kitaev A. 1995. Quantum measurements and the Abelian stabilizer problem. arXiv:quant-ph/9511026

60. Biamonte JD, Bergholm V, Whitfield JD, Fitzsimons J, Aspuru-Guzik A. 2010. Adiabatic quantum simulators. arXiv:1002.0368

61. Oh S. 2008. Quantum computational method of finding the ground-state energy and expectation values. *Phys. Rev. A* 77:012326

62. Davis MJ, Heller EJ. 1981. Multidimensional wave functions from classical trajectories. *J. Chem. Phys.* 75:3916–24

63. Tannor DJ. 2006. *Introduction to Quantum Mechanics: A Time-Dependent Perspective*. Mill Valley, CA: Univ. Sci. Books

64. Giovannetti V, Lloyd S, Maccone L. 2004. Quantum-enhanced measurements: beating the standard quantum limit. *Science* 306:1330–36

65. Knill E, Ortiz G, Somma R. 2007. Optimal quantum measurements of expectation values of observables. *Phys. Rev. A* 75:012328

66. Smirnov A, Savel'ev S, Mourokh L, Nori F. 2007. Modelling chemical reactions using semiconductor quantum dots. *Europhys. Lett.* 80:67008

67. Terhal B, DiVincenzo D. 2000. Problem of equilibration and the computation of correlation functions on a quantum computer. *Phys. Rev. A* 61:022301

68. Kassal I, Aspuru-Guzik A. 2009. Quantum algorithm for molecular properties and geometry optimization. *J. Chem. Phys.* 131:224102

69. Jordan SP. 2005. Fast quantum algorithm for numerical gradient estimation. *Phys. Rev. Lett.* 95:050501

70. Wang H, Kais S, Aspuru-Guzik A, Hoffmann MR. 2008. Quantum algorithm for obtaining the energy spectrum of molecular systems. *Phys. Chem. Chem. Phys.* 10:5388–93

71. Wu LA, Byrd MS, Lidar DA. 2002. Polynomial-time simulation of pairing models on a quantum computer. *Phys. Rev. Lett.* 89:057904

72. Aharonov D, Ta-Shma A. 2008. Adiabatic quantum state generation. *SIAM J. Comput.* 37:47–82

73. Somma R, Batista C, Ortiz G. 2007. Quantum approach to classical statistical mechanics. *Phys. Rev. Lett.* 99:030603

74. Temme K, Osborne T, Vollbrecht K, Poulin D, Verstraete F. 2009. Quantum Metropolis sampling. arXiv:0911.3635
75. Chiang C, Wocjan P. 2010. Quantum algorithm for preparing thermal Gibbs states-detailed analysis. arXiv:1001.1130
76. Poulin D, Wocjan P. 2009. Preparing ground states of quantum many-body systems on a quantum computer. *Phys. Rev. Lett.* 102:130503
77. Poulin D, Wocjan P. 2009. Sampling from the thermal quantum Gibbs state and evaluating partition functions with a quantum computer. *Phys. Rev. Lett.* 103:220502
78. Somma R, Boixo S, Barnum H, Knill E. 2008. Quantum simulations of classical annealing processes. *Phys. Rev. Lett.* 101:130504
79. Szegedy M. 2004. Quantum speed-up of Markov chain based algorithms. In *Proc. 45th Annu. IEEE Symp. Found. Comput. Sci.*, pp. 32–41. Washington, DC: IEEE Comput. Soc.
80. Wocjan P, Abeyesinghe A. 2008. Speedup via quantum sampling. *Phys. Rev. A* 78:042336
81. Mezard M, Montanari A. 2009. *Information, Physics, and Computation*. New York: Oxford Univ. Press
82. Yung MH, Nagaj D, Whitfield JD, Aspuru-Guzik A. 2010. Simulation of classical thermal states on a quantum computer: a transfer matrix approach. arXiv:1005.0020
83. Kohn W. 1999. Nobel lecture: Electronic structure of matter: wave functions and density functionals. *Rev. Mod. Phys.* 71:1253–66
84. Aharonov D, Naveh T. 2002. Quantum NP: a survey. arXiv:quant-ph/0210077
85. Kitaev A, Shen AH, Vyalyi MN. 2002. *Classical and Quantum Computation*. Providence, RI: Am. Math. Soc.
86. Oliveira R, Terhal BM. 2008. The complexity of quantum spin systems on a two-dimensional square lattice. *Quantum Inf. Comput.* 8:900–24
87. Schuch N, Verstraete F. 2009. Computational complexity of interacting electrons and fundamental limitations of density functional theory. *Nat. Phys.* 5:732–35
88. Floudas CA, Pardalos PM. 2000. *Optimization in Computational Chemistry and Molecular Biology: Local and Global Approaches*. New York: Springer. 1st ed.
89. Hartmann AK, Rieger H. 2004. *New Optimization Algorithms in Physics*. New York: Wiley-VCH
90. Choi V. 2010. Adiabatic quantum algorithms for the NP-complete maximum-weight independent set, exact cover and 3-SAT problems. arXiv:1004.2226
91. Hogg T. 2003. Adiabatic quantum computing for random satisfiability problems. *Phys. Rev. A* 67:022314
92. Neven H, Denchev VS, Rose G, Macready WG. 2009. Training a large scale classifier with the quantum adiabatic algorithm. arXiv:0912.0779
93. Neven H, Rose G, Macready WG. 2008. Image recognition with an adiabatic quantum computer. I: Mapping to quadratic unconstrained binary optimization. arXiv:0804.4457
94. Kolinski A, Skolnick J. 1996. *Lattice Models of Protein Folding, Dynamics and Thermodynamics*. London: Chapman & Hall
95. Mirny L, Shakhnovich E. 2001. Protein folding theory: from lattice to all-atom models. *Annu. Rev. Biophys. Biomol. Struct.* 30:361–96
96. Pande VS, Grosberg AY, Tanaka T. 2000. Heteropolymer freezing and design: towards physical models of protein folding. *Rev. Mod. Phys.* 72:259–86
97. Hart WE, Istrail S. 1997. Robust proofs of NP-Hardness for protein folding: general lattices and energy potentials. *J. Comput. Biol.* 4:1–22
98. Dill KA, Ozkan SB, Shell MS, Weikl TR. 2008. The protein folding problem. *Annu. Rev. Biophys.* 37:289–316
99. Lau KF, Dill KA. 1989. A lattice statistical mechanics model of the conformational and sequence spaces of proteins. *Macromolecules* 22:3986–97
100. Epstein CJ, Goldberger RF, Anfinsen CB. 1963. Genetic control of tertiary protein structure: studies with model systems. *Cold Spring Harb. Symp.* 28:439–49
101. Kaminsky WM, Lloyd S, Orlando TP. 2004. Scalable superconducting architecture for adiabatic quantum computation. arXiv:quant-ph/0403090

102. Perdomo-Ortiz A, O'Gorman B, Aspuru-Guzik A. 2010. Construction of energy functions for self-avoiding walks and the lattice heteropolymer model: resource efficient encoding for quantum optimization. Manuscript in preparation

103. Perdomo-Ortiz A, Drew-Brook M, Dickson N, Rose G, Aspuru-Guzik A. 2010. Experimental realization of a 8-qubit quantum-adiabatic algorithm for a lattice protein model: towards optimization on a quantum computer. Manuscript in preparation

104. Berger B, Leighton T. 1998. Protein folding in the hydrophobic-hydrophilic (HP) model is NP-complete. *J. Comput. Biol.* 5:27–40

105. Crescenzi P, Goldman D, Papadimitriou C, Piccolboni A, Yannakakis M. 1998. On the complexity of protein folding. *J. Comput. Biol.* 5:597–603

106. Amin MHS, Choi V. 2009. First-order quantum phase transition in adiabatic quantum computation. *Phys. Rev. A* 80:062326

107. Farhi E, Goldstone J, Gosset D, Gutmann S, Meyer H, Shor P. 2009. Quantum adiabatic algorithms, small gaps, and different paths. arXiv:0909.4766v2

108. Perdomo-Ortiz A, Venegas-Andraca S, Aspuru-Guzik A. 2010. A study of heuristic guesses for adiabatic quantum computation. *Quantum Inf. Process.* In press; doi: 10.1007/s11128-010-0168-z

109. Somaroo S, Tseng CH, Havel TF, Laflamme R, Cory DG. 1999. Quantum simulations on a quantum computer. *Phys. Rev. Lett.* 82:5381–84

110. Kok P, Munro WJ, Nemoto K, Ralph TC, Dowling JP, Milburn GJ. 2007. Linear optical quantum computing with photonic qubits. *Rev. Mod. Phys.* 79:135–40

111. O'Brien JL. 2007. Optical quantum computing. *Science* 318:1567–70

112. Lanyon BP, Whitfield JD, Gillett GG, Goggin ME, Almeida MP, et al. 2010. Towards quantum chemistry on a quantum computer. *Nat. Chem.* 2:106–11

112. The first experimental implementation of a quantum-chemical algorithm on a quantum computer.

113. Baugh J, Chamilliard J, Chandrashekar CM, Ditty M, Hubbard A, et al. 2007. Quantum information processing using nuclear and electron magnetic resonance: review and prospects. *Phys. Canada* 63:197–206

114. Brown KR, Clark RJ, Chuang IL. 2006. Limitations of quantum simulation examined by simulating a pairing Hamiltonian using nuclear magnetic resonance. *Phys. Rev. Lett.* 97:050504

115. Negrevergne C, Somma R, Ortiz G, Knill E, Laflamme R. 2005. Liquid-state NMR simulations of quantum many-body problems. *Phys. Rev. A* 71:032344

116. Peng X, Du J, Suter D. 2005. Quantum phase transition of ground-state entanglement in a Heisenberg spin chain simulated in an NMR quantum computer. *Phys. Rev. A* 71:012307

117. Yang X, Wang A, Xu F, Du J. 2006. Experimental simulation of a pairing Hamiltonian on an NMR quantum computer. *Chem. Phys. Lett.* 422:20–24

118. Du J, Xu N, Wang P, Wu S, Lu D. 2010. NMR implementation of a molecular hydrogen quantum simulation with adiabatic state preparation. *Phys. Rev. Lett.* 104:030502

119. Levi BG. 2009. Superconducting qubit systems come of age. *Phys. Today* 62:114–16

120. You JQ, Nori F. 2005. Superconducting circuits and quantum information. *Phys. Today* 58:42–47

121. Hofheinz M, Wang H, Ansmann M, Bialczak RC, Lucero E, et al. 2009. Synthesizing arbitrary quantum states in a superconducting resonator. *Nature* 459:546–49

122. DiCarlo L, Chow JM, Gambetta JM, Bishop LS, Johnson BR, et al. 2009. Demonstration of two-qubit algorithms with a superconducting quantum processor. *Nature* 460:240–44

123. Harris R, Johansson J, Berkley AJ, Johnson MW, Lanting T, et al. 2010. Experimental demonstration of a robust and scalable flux qubit. *Phys. Rev. B* 81:134510

124. Johnson MW, Bunyk P, Maibaum F, Tolkacheva E, Berkley AJ, et al. 2010. A scalable control system for a superconducting adiabatic quantum optimization processor. *Supercond. Sci. Technol.* 23:065004

125. Berkley AJ, Johnson MW, Bunyk P, Harris R, Johansson J, et al. 2009. A scalable readout system for a superconducting adiabatic quantum optimization system. arXiv:0905.0891

126. Harris R, Johnson MW, Lanting T, Berkley AJ, Johansson J, et al. 2010. Experimental investigation of an eight qubit unit cell in a superconducting optimization processor. *Phys. Rev. B* 82:024511

127. Harris R, Lanting T, Berkley AJ, Johansson J, Johnson MW, et al. 2009. Compound Josephson-junction coupler for flux qubits with minimal crosstalk. *Phys. Rev. B* 80:052506

128. Lanting T, Harris R, Johansson J, Amin MHS, Berkley AJ, et al. 2010. Observation of co-tunneling in pairs of coupled flux qubits. arXiv:1006.0028

129. Blatt R, Wineland D. 2008. Entangled states of trapped atomic ions. *Nature* 453:1008–15

130. Cirac JI, Zoller P. 1995. Quantum computations with cold trapped ions. *Phys. Rev. Lett.* 74:4091–94

131. Johanning M, Varón AF, Wunderlich C. 2009. Quantum simulations with cold trapped ions. *J. Phys. B* 42:154009

132. Friedenauer A, Schmitz H, Glueckert JT, Porras D, Schaetz T. 2008. Simulating a quantum magnet with trapped ions. *Nat. Phys.* 4:757–61

133. Gerritsma R, Kirchmair G, Zähringer F, Solano E, Blatt R, Roos CF. 2010. Quantum simulation of the Dirac equation. *Nature* 463:68–71

134. Kim K, Chang MS, Korenblit S, Islam R, Edwards EE, et al. 2010. Quantum simulation of frustrated Ising spins with trapped ions. *Nature* 465:590–93

135. Childs AM. 2009. Universal computation by quantum walk. *Phys. Rev. Lett.* 102:180501

136. Lovett NB, Cooper S, Everitt M, Trevers M, Kendon V. 2010. Universal quantum computation using discrete time quantum walk. *Phys. Rev. A* 81:042330

137. Clark CR, Metodi TS, Gasster SD, Brown KR. 2009. Resource requirements for fault-tolerant quantum simulation: the ground state of the transverse Ising model. *Phys. Rev. A* 79:062314

138. Chan GKL, Dorando JJ, Ghosh D, Hachmann J, Neuscamman E, et al. 2008. An introduction to the density matrix renormalization group ansatz in quantum chemistry. *Prog. Theor. Chem. Phys.* 18:49–65

139. Jacksch P, Papageorgiou A. 2003. Eigenvector approximation leading to exponential speedup of quantum eigenvalue calculation. *Phys. Rev. Lett.* 91:257902

Annual Review of
Physical Chemistry

Volume 62, 2011

Contents

Indexes

Errata

An online log of corrections to *Annual Review of Physical Chemistry* articles may be found at http://physchem.annualreviews.org/errata.shtml