

Haverford College

Haverford Scholarship

Faculty Publications

Physics

2012

Computational Complexity in Electronic Structure

James D. Whitefield

Peter John Love
Haverford College

Alan Aspuru-Guzik

Follow this and additional works at: https://scholarship.haverford.edu/physics_facpubs

Repository Citation

Whitfield, James Daniel, Peter John Love, and Alán Aspuru-Guzik. "Computational complexity in electronic structure." *Physical Chemistry Chemical Physics* 15.2 (2013): 397-411.

This Journal Article is brought to you for free and open access by the Physics at Haverford Scholarship. It has been accepted for inclusion in Faculty Publications by an authorized administrator of Haverford Scholarship. For more information, please contact nmedeiro@haverford.edu.

Computational Complexity in Electronic Structure

James Daniel Whitfield,^{a,b,c} Peter John Love^d and Alán Aspuru-Guzik^e

In quantum chemistry, the price paid by all known efficient model chemistries is either the truncation of the Hilbert space or uncontrolled approximations. Theoretical computer science suggests that these restrictions are not mere shortcomings of the algorithm designers and programmers but could stem from the inherent difficulty of simulating quantum systems. Extensions of computer science and information processing exploiting quantum mechanics has led to new ways of understanding the ultimate limitations of computational power. Interestingly, this perspective helps us understand widely used model chemistries in a new light. In this article, the fundamentals of computational complexity will be reviewed and motivated from the vantage point of chemistry. Then recent results from the computational complexity literature regarding common model chemistries including Hartree-Fock and density functional theory are discussed.

1 Introduction

Quantum chemistry is often concerned with solving the Schrödinger equation for chemically-relevant systems such as atoms, molecules, or nanoparticles. By solving a differential and/or eigenvalue equation, the properties of the system and the dynamics of the state are obtained. Examples of properties include: equilibrium geometries, the dissociation energy of molecules, and the vibrational frequencies.

The difficulty stems from the accuracy required and the apparent exponential growth of the computational cost with both the number of electrons and the quality of the description of the system. For practical applications, the accuracy required is typically orders of magnitude smaller than the total energy of the system. As a concrete example, the Carbon atom has total electronic energy of about 37.8 Hartrees while the energy of a Carbon-Hydrogen bond is only 0.16 Hartrees. Solving the full eigenvalue equation takes on the order of n^3 operations for an $n \times n$ matrix. However, when describing interacting many-electron systems, the dimension of the matrix increases exponentially with the number of electrons.

Consequently, the computational methods of electronic structure in chemistry are aimed at circumventing exact diagonalization, in the context of electron structure, called the full configuration method. Avoiding exact diagonalization has led to a wide range of computational methods¹⁻⁴ for computing properties of chemical interest. These methods have recently⁵⁻⁹ begun to include *quantum simulation* following Feynman's suggestion¹⁰ to use quantum computers as sim-

ulators. Subsequent development of these ideas in quantum chemistry has led to new proposed methods utilizing quantum computational techniques^{8,11-14} and proof-of-principle experiments^{15,16}. However, questions about when and where one would expect a quantum computer to be useful^{17,18} have not been fully answered. Ideally, computational complexity can provide some answers about when, where, and why quantum computers would be useful for chemistry. At the same time, it could also help formalize intuitive understanding of when we can expect reliable results from traditional computational methods.

Many results in computational complexity have interchanged classical and quantum computers fluidly leading to new results on the complexity of computing properties of quantum systems¹⁹. We review some recent results appearing in the computational complexity literature that touch on why electronic structure calculations are difficult. Our hope is to encourage future investigations into quantitative understanding of difficult instances of electronic structure calculations.

A similar, but much shorter, discussion of computational complexity in quantum chemistry by Rassolov and Garashchuk²⁰ appeared in 2008 and provided many conjectures that have since been proven or extended.

This perspective assumes exposure to second quantization and mixed states in quantum mechanics. Standard bra-ket notation is used when referring to quantum states. All the necessary concepts from computational complexity and quantum computation are briefly introduced and motivated to make the article as self-contained as possible. A key omission from the present review is a number of computational complexity results^{21,22} concerning quantum-information based wave function ansatzes such matrix product states, density matrix renormalization group and their generalizations²³. These methods are becoming accepted into the mainstream of computational chemistry²⁴⁻²⁶, but do not yet have the widespread availability of the selected methods included in the present work.

^a Vienna Center for Quantum Science and Technology (VCQ), Boltzmanngasse 5, 1090 Vienna, Austria. E-mail: James.Whitfield@univie.ac.at

^b Columbia University, Physics Department, 538 West 120th Street, New York, NY, USA.

^c NEC Laboratories America, 1 Independence Way, Princeton, NJ, USA.

^d Haverford College. 370 Lancaster Avenue, Haverford, PA, USA. Email: plove@haverford.edu

^e Harvard University, Department of Chemistry and Chemical Biology, Cambridge, MA, USA. Email: aspuru@chem.harvard.edu

2 Worst case computational complexity for chemists

The purpose of this section is to provide a quick but precise introduction to computational complexity concepts^{27,28}. The aim is to set the stage for the results subsequently reviewed.

Computational complexity is the study of how resources required to solve a problem change with its size. For instance, *space complexity* is the scaling of memory requirements with the problem size, but, in this article, we focus on *time complexity* which investigates how the running time of the computation changes as the problem size increases.

Computational chemists often informally discuss commonplace concepts in computer science such as the complexity classes of polynomial-time problems (P) and non-deterministic polynomial-time problems (NP). For instance, it is sometimes stated that Hartree-Fock has a runtime which scales as the third power of the number of basis functions. This is scaling disregards difficult instances of the calculation where Hartree-Fock does not converge. Such instances require manual intervention to tweak the algorithm used or adjust the convergence thresholds in a case-by-case fashion.

In computer science, often the focus is on *worst-case complexity* where the most difficult instances of a problem are used to classify the problem's complexity. This has been one of the major areas of theoretical computer science, although work on average case complexity does exist^{29,30}. In this article, the complexity of the problems discussed are characterized by the worst-case scaling.

2.1 Time complexity in equivalent computer models

A proper measure of the *time complexity* of an algorithm is how many basic 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, i.e. one that scales polynomially, $O(n^k)$ with fixed k and input size n .^{*} Otherwise, the problem is considered intractable. This is an asymptotic definition that may not capture the full utility of the algorithm. For example, an asymptotically efficient algorithm may run slower than an asymptotically inefficient for small or fixed size problem instances. Nevertheless, this asymptotic classification of algorithms has proved useful. From a theoretical computer science perspective, the division allows for considerable progress to be made without considering the minutiae of the specific system, implementation, or domain of application.

From a practical standpoint, Moore's³¹ law states the density of transistors in classical computers doubles every two years. Thus, for a fixed computational time, if the computer cannot solve an instance when using a polynomial-time algorithm, one need not wait long as the exponential growth of computing power will reasonably quickly overtake the polynomially large cost. However, if the algorithm runs in exponential-time, one may be forced to wait several lifetimes in order for an instance to become soluble even if Moore's law continues indefinitely. Complicating matters, the exponential growth according to Moore's law is expected to cease sometime this century; hence the recent emphasis on quantum computation.

The time complexity can be characterized using any computationally equivalent model. In the context of computer science, *equivalent* means that one model can simulate the other with an overhead that scales polynomially as a function of system size. These mappings respects the boundary between efficient and inefficient algorithms described earlier. Turing machines and circuits are two typical models discussed in the context of computational complexity.

The Turing computer or *Turing machine* was introduced by Alan Turing³² in 1936 before transistors and electronic circuits and formalizes the idea of a computer as a person to whom computational instruction could be given. Turing³² introduced the concept in order to answer David Hilbert's challenge to decide if a polynomial has roots which are integers using a "finite number of operations." Turing proved that this was not possible using his Turing machine. An illustration depicting the salient features of the Turing machine is given in Fig. 1.

The circuit model of computation has been more widely used when generalizing to the quantum setting although the quantum Turing machine formulation does exist^{33,34}. The time complexity in the circuit model is characterized the number of circuit elements (or gates) used to compute the accept/reject boolean value corresponding to an input. So long as a universal set of gates, e.g. NAND, FAN-OUT, and FAN-IN for classical circuits, are used, the time complexity will only differ by a polynomial factor. The key caveat to consider when using this model is the notion of uniformity. To properly define an algorithm in the circuit model, a way of generating circuits for all input sizes must be given concisely. A circuit family for an algorithm is *uniform* if there is a polynomial-time algorithm that specifies the circuit given the input size.

2.2 Computational problems and problem instances

In computer science, computational problem often refers to decision problems which are collections of yes/no questions that an algorithm can decide. Each question is called a *problem instance* and the answer is called the *solution* to that in-

* The notation, $f(x) = O(g(x))$, implies that function $f(x)$ is dominated by $g(x)$ for asymptotically large values of x . To indicate that $f(x)$ is asymptotically larger than $g(x)$, we write $f(x) = \Omega(g(x))$. If $f(x)$ is dominated by and dominates function $g(x)$, then we write $f(x) = \Theta(g(x))$.

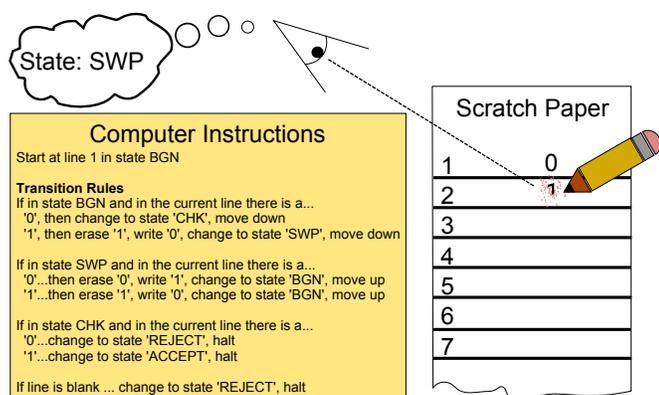


Fig. 1 Illustration of a Turing machine that accepts input strings beginning with '01' or '10.' The computer is specified by the set of states (modes of operation), an alphabet of symbols for the scratch space, and the transition rules. The computer performing an algorithm reads the symbol on the current line, then based on the computer's current state and the given transition rules, the computer changes the symbol of the current line, changes its current state and moves either up or down. The computer halts when it enters the ACCEPT or REJECT states which indicate the output of the computation. In the figure, the possible states are {BGN, CHK, SWP, REJECT, ACCEPT}, the alphabet is {'0','1'}, and the transition rules are listed in the box to the left. Depicted is the second step of a computation that began with input '10'.

stance. Although we focus on decision problem, in many situations the decision problem can be used to extract numerical values by asking sufficiently many yes/no questions.

A collection of decision questions with affirmative answers such as, "Is 149 prime?" or "Is 79 prime?," is called a *language*. Languages define decision problems where only the accept instances are included in the language. If an algorithm *accepts* (returns "True" or outputs 1) on all strings contained in a language then the device is said to *recognize* the language. If the computer is also required to halt on all inputs, then it is said to *decide* the language. The difference being, a Turing machine may recognize a language but fail to halt on some instances, i.e. run forever. The problem of deciding if a Turing machine algorithm halts is called the HALTING problem. This problem has a rich and storied history beginning with the first paper of Turing³². In Fig. 1, the algorithm decides the language of strings beginning with '01' or '10'.

As an example from chemistry, the computational problem of deciding if a molecule has a dipole moment, DIPOLE, is a collection of questions: "Does x have a dipole moment?" Here, x is a string representing the molecule in each instance. The questions 'Does BCl_3 have a dipole moment?', and 'Does NH_3 have a dipole moment?' are instances of this computational problem. The string $x = \text{"BCl}_3\text{"}$ is not in the language

DIPOLE since " BCl_3 " does not have dipole moment while " NH_3 " is in the language since it has a dipole moment. A reasonable modification of the problem would be 'Does x have a dipole moment greater than 0.1 Debyes?' so that small dipole moments may be ignored.

It may also be necessary to promise that x represents a molecule in a specific way such that ill formatted inputs can be ignored. This is accomplished using *promise problems* where the promise would be that x is indeed a string that properly encodes a molecule. Promise problems also arise in order to account for issues of numerical or experimental precision as illustrated in the next paragraph.

Promise problems also play a key role throughout the remainder of this text since we are discussing physical properties where infinite precision is neither required nor expected. As an illustration, we use an example from thermochemistry, where language A is the set of strings corresponding to (ideal) gases with heat capacity at constant pressure, C_p , less than some critical value. Imagine you have an unreasonable lab instructor, who gives you a substance that has heat capacity extremely close to the critical value that decides membership in A . It may take a large number of repetitions of the experimental protocol to be confident that the substance belongs or does not belong to language A . A reasonable lab instructor would announce at the beginning of lab that all the substances he is handing out for the experiment are promised to be at least one Joule per Kelvin away from the critical value. Given this promise, the student will be able to complete their lab in a single lab period. Without such a promise, it may take the student several lab periods to repeat the experiment in order to establish a sufficiently precise value of the heat capacity to decide the instance. Instead when using a promise, if the student has a compound that violates the promise then the instructor would give full credit for either answer.

More formally, in the usual decision problems, the problem is defined using one language, L , specifying the accept instances. The language of reject instances is all strings not in L . Promise problems differ in that they are specified using two languages: one for the accept instances, L_{accept} , and a separate language for the reject instances, L_{reject} . If an instance is not in either of these languages, then the promise is violated and the computation can terminate with any result.

Reusing the DIPOLE example, if we can reformulate the problem to find out about non-zero dipole moments as a promise problem to account for experimental error. Now the computational task of DIPOLE is: 'Given a molecule, x , a trial value for the dipole d_T , and an error tolerance δ , decide if the dipole moment of x is greater than $d_T + \delta$ or less than $d_T - \delta$, promised that the dipole moment is not between $d_T \pm \delta$.' In this case, there are two languages defining the problem $L_{d > d_T + \delta}$ and $L_{d < d_T - \delta}$. If a molecule has a dipole moment of exactly d_T it would violate the promise and the

experimenter or computer does not need to answer or can respond with any answer. The value δ allows us to meaningfully define problems in the presence of errors resulting from imperfect experimental measurements or from the finite precision of a computing device.

2.3 Computation reductions

The idea of computational reduction is at the heart of classifying computational problems. Reducibility is a way of formalizing the relationship between problems. Essentially, it asks, “If I can solve one problem, what other problems can I solve using the same resources?”

There are two main types of reductions: Turing reductions and Karp reductions. For the reduction to be useful, they must be limited to polynomial time. The polynomial-time *Turing reduction*, also called the Cook reduction, of problem A to problem B uses solutions to multiple instances of B to decide the solutions for instances of A . The solutions to instances of B are provided by an *oracle* and each question is called a *query*. Algorithms deciding A which require only polynomial queries of the oracle for B are efficient whenever the oracle for B answers in polynomial time.

The other type of reduction is called a *Karp reduction* or polynomial transformation. If, instead of an oracle, there is an embedding of instances of problem A into instances of problem B , then an efficient solution for all instances of problem B imply efficient solutions for instances of problem A . This transformation is a Karp reduction. When A can be *reduced* to B , under either Karp or Turing reductions, it is denoted $A \leq B$.

To illustrate the difference, we use examples from thermodynamics. Consider trying to determine if the heat capacity at constant pressure, C_p , of an ideal substance is less than some critical value (language A). Now, suppose an oracle, probably in the form of a bomb calorimeter, has been provided for deciding if the heat capacity at constant volume, C_V , of a given substance is above or below a critical value. Call language B the set of strings labeling substances with C_V below this value. By adding the number of moles times the ideal gas constant to the critical value, one can determine membership in language A via the formula $C_p = C_V + nR$. Since each instance of A can be embedded into an instance of B , language A is Karp reducible to B and we can write $A \leq B$.

Suppose instead, an oracle for evaluating if, at fixed pressure, the internal energy, U , of the given substance at a given temperature is less than some critical value (language C) is given. Then by evaluating the internal energy at two different temperatures, the heat capacity at constant pressure can be bounded by numerically estimating the derivative. Because the reduction has to use the oracle for C more than once to decide instances of A , language A is Turing reducible to C and $A \leq C$.

2.4 Basic complexity classes

Equipped with the basic definitions from computer science, we now introduce the concept of computational complexity classes. This will give some insights into why quantum chemistry is difficult. We will introduce six basic complexity classes to classify the time complexity of decision problems. The first three complexity classes are characterized by algorithms that can decide instances in time proportional to a polynomial of the input size; the other classes are characterized by polynomial time verification of problem instances.

In table 1, the three polynomial time complexity classes are listed. First, P , is the class of all decision problems with instances that can be accepted or rejected by a Turing machine in polynomial time. If the Turing machine has access to an unbiased random number generator, then the complexity class of problems that can be decided in polynomial time is called BPP . The term “bounded error” refers to the requirement that the probability of acceptance and of rejection must be bounded away from half so that repetition can be employed to boost the confidence in the answer. An important class of problems falling into this complexity class are efficient Monte Carlo simulations often grouped under the umbrella term quantum Monte Carlo^{35–37} used for computing electronic structure in chemistry. Lastly, the complexity class BQP is characterized by problems soluble in polynomial time with quantum resources. In the quantum computational model³⁸, the quantum algorithm is conceptually simpler to think of as a unitary circuit, U , composed of unitary circuit element that affect, at most, two quantum bits. As mentioned earlier, the number of gates used determines the time complexity. The outcome of the algorithm with input $|Input\rangle$ is “accept” with probability $|\langle Accept|U|Input\rangle|^2$. Similarly, for the “reject” cases.

So far, the discussion has centered on complexity classes containing problems that are solvable in polynomial time given access to various resources: a computer (P), a random number generator (BPP), and a controllable quantum system (BQP). Now, our attention turns to the class of problems that can be computed non-deterministically. The original notion of non-determinism is a Turing machine whose transition rule maps the computer state and the tape symbol to any number of possible outputs. In essence, this is a computer that can clone itself at will to pursue all options at once. NP is the class of problems that could be solved in polynomial time by such a computer. Whether a deterministic polynomial time computer can be used to simulate such a computer is a restatement of the famous open question in computer science: “Does $P = NP$?” This question was selected as a millennium problem by the Clay Mathematics Institute which has offered a one million dollar prize for a correct proof of the answer.

Rather than resorting to imaginary computers, the non-deterministic classes can be defined by considering verifi-

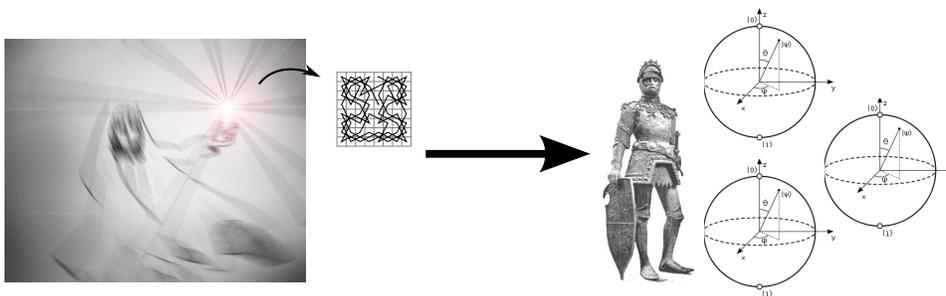


Fig. 2 An example of the non-deterministic complexity class QMA. Non-deterministic problems can be thought of as games. In the Merlin-Arthur games, the proof verifying the validity of input x is magically (hence non-deterministically) given by a wizard Merlin who is prone to deception. The verifier, called Arthur, is trying to decide if he should accept or reject input x . When Merlin gives Arthur a valid proof that verifies x should be accepted, Arthur should accept with high probability (*completeness*) and when Merlin gives him a proof of something incorrect, he should be able to spot the inaccuracy (*soundness*). In the drawing since Arthur has access to quantum bits (qubits), the complexity class depicted is QMA. Figure composed from images found on Wikipedia under the Creative Commons License.

cation of solutions which have been obtained in some non-deterministic way, see Fig. 2. For every “accept” instance, x , there exists at least one proof state y such that the verifier returns “accept” in polynomial time. If x should be rejected, then for every proof state y , the verifier should output “reject”, again, in a polynomial amount of time. Each of the non-deterministic classes listed in table 1, are characterized by the computational power of the verifier. Note that P is a subset of NP because any problem that can be easily solved can be easily verified.

2.5 Completeness and hardness

The goal of classifying computational problems into complexity classes motivates introduction of the terms *hard* and *complete*. The classification of a computational problems as hard for a complexity class means if an algorithm can solve this problem, then all problems in the class can be solved efficiently. In other words, this problem contains all problems in the class either via Karp or Turing reductions. More precisely, a language B is *hard* for a complexity class CC if every language, A , in CC is polynomial-time reducible to B , i.e. $A \leq B$.

Complete problems for a class are, in some sense, the hardest problems in that class. These problems are both a member of the class and are also hard for the class. That is, a language B is *complete* for a complexity class CC if B is in the complexity class CC and for every other language A in CC, $A \leq B$.

The simplest illustration of the difference between hard and complete computational problems is the distinction between optimization problems and decision problems. Optimization problems such as finding saddle points or minima in energy landscapes are frequently encountered in computational chemistry, however these problems are not in the complexity class NP. However, if one can perform optimization

quickly, then responding with the answer to yes/no questions about the solution would be simple. Thus, optimization problems can be classified as NP-hard but not NP-complete since the computational task of optimization is, in a sense, harder than just answering yes or no.

3 Electronic structure and other Hamiltonian problems

Now, armed with the key ideas from computer science, we return to our original inquiry into why quantum chemistry is hard. We answer this questions by exploring the computational complexity of three different widely used methods for computing electronic energy in computational chemistry: Hartree-Fock, two-electron reduced density matrix methods, and density functional theory.

Before delving into the specific complexities of these problems, we first establish notation by reviewing necessary concepts from quantum chemistry, then discussing computational problems concerning classical and quantum spin Hamiltonians. We end the section with mappings between systems of electrons and spin systems.

3.1 Quantum chemistry and second quantization

In quantum chemistry, the annihilation, $\{a_k\}$, and creation operators, $\{a_k^\dagger\}$, correspond respectively to removing and adding an electron into one of M single particle wave functions, $\{\phi_k(\mathbf{x}_1)\}_{k=1}^M$. The single-particle wave functions are called *orbitals* and the set of orbitals is typically called the *basis set*. To include the electron spin, the single particle orbitals ϕ_k are functions of spatial coordinates and a spin variable which are collectively denoted \mathbf{x} .

Table 1 Polynomial time complexity classes are separated by the resources the computer has access to while the non-deterministic polynomial time complexity classes are characterized by the resources of polynomial time computational verifiers

Class	Name	Computer type taking only poly. time
P	Polynomial Time	Turing machine
BPP	Bounded Error Probabilistic Polynomial Time	Turing machine with access to random number generator
BQP	Bounded Error Quantum Polynomial Time	Turing machine with access to quantum resources

Class	Name	Verifier's computer type taking only poly. time
NP	Non-deterministic Polynomial Time	Turing machine
MA	Merlin-Arthur	Turing machine with access to random number generator
QMA	Quantum Merlin-Arthur	Turing machine with access to quantum resources

The electronic spin will play an important role when discussing the connections between systems of electrons and systems of quantum spins. Since electrons are spin- $\frac{1}{2}$ particles, the electron spin is either up or down. Accordingly, there are $M/2$ orbitals with spin up and $M/2$ with spin down. Unless explicitly noted, the summation over orbitals includes a summation over the spatial and spin indices of the orbitals.

Anti-symmetry of the N -electron is enforced by the canonical anti-commutation relations for fermions, i.e. electrons,

$$[a_p, a_q]_+ = a_p a_q + a_q a_p = 0, \quad [a_p, a_q^\dagger]_+ = \delta_{pq} \mathbf{1}. \quad (1)$$

In chemistry, typically the electronic structure is the primary concern, but if interested in vibrational structure, we would have to consider bosonic canonical commutation relations: $[b_p, b_q] = b_p b_q - b_q b_p = 0$ and $[b_p, b_q^\dagger] = \delta_{pq} \mathbf{1}$. The vacuum state, $|vac\rangle$, is the normalized state that the creation/annihilation operators act on and it represents the system with no particles.

Acting on the vacuum state with a strings of N distinct creation operators yields N -electron wave functions. The most general N -electron state within a basis set is,

$$|\Psi\rangle = \sum_K C_K a_{K_1}^\dagger a_{K_2}^\dagger \cdots a_{K_N}^\dagger |vac\rangle, \quad (2)$$

with $K = (K_1, \dots, K_N)$ and the complex valued C_K are con-

strained such that $\sum_K |C_K|^2 = 1$. To convert between creation and annihilation operators and the coordinate representation, one uses field operators: $\hat{\phi}(\mathbf{x}) = \sum_k \phi_k(\mathbf{x}) a_k$ as in $\Psi(\mathbf{x}_1, \dots, \mathbf{x}_N) = \langle vac | \hat{\phi}(\mathbf{x}_1) \cdots \hat{\phi}(\mathbf{x}_N) | \Psi \rangle / \sqrt{N!}$. The factor of $N!$ accounts for permutation symmetries in the summation. Each state Ψ or $|\Psi\rangle$ is called an N -electron *pure state*. A valid N -electron *mixed state*,

$$\rho^{(N)} = \sum p_i |\Psi_i\rangle \langle \Psi_i| \quad (3)$$

has $\sum p_i = 1$ where each $|\Psi_i\rangle$ is an N -electron wave function. The k -th order reduced density matrix, abbreviated k -RDM, is defined using the field operators:

$$\begin{aligned} \rho^{(k)}(\mathbf{x}_1, \mathbf{x}'_1, \dots, \mathbf{x}_k, \mathbf{x}'_k) \\ = \frac{1}{(N)_k} \langle \Psi | \hat{\phi}^\dagger(\mathbf{x}'_k) \cdots \hat{\phi}^\dagger(\mathbf{x}'_1) \hat{\phi}(\mathbf{x}_1) \cdots \hat{\phi}(\mathbf{x}_k) | \Psi \rangle \end{aligned} \quad (4)$$

$$= \frac{1}{(N)_k} \text{Tr} \left[\hat{\phi}(\mathbf{x}_1) \cdots \hat{\phi}(\mathbf{x}_k) \rho^{(N)} \hat{\phi}^\dagger(\mathbf{x}'_k) \cdots \hat{\phi}^\dagger(\mathbf{x}'_1) \right] \quad (5)$$

With $(N)_k = N!/(N-k)!$, the reduced density matrices are normalized to unity and, in equation (5), the trace sums over the expectation values of states from a complete set of $N-k$ electron wave functions.

Lastly, we define the primary computational chemistry problem encountered in computation of electronic structure.

Computational Problem: ELECTRONIC STRUCTURE. The inputs are the number of electrons, N , a set of M orbitals, a static configuration of nuclei, a trial energy, E_T , an error tolerance δ and Hamiltonian,

$$H_{elec} = T_e + V_{ee} + V_{eN} = \sum_{ij} (T_{ij}^e + V_{ij}^{eN}) a_i^\dagger a_j + V_{ijkl}^{ee} a_i^\dagger a_j^\dagger a_k a_l \quad (6)$$

with T_e the electronic kinetic energy operator, V_{ee} the electron-electron interaction operator and V_{eN} the electron-nuclear interaction. The task is to decide if the ground state energy is less than $E_T - \delta$ or greater than $E_T + \delta$ promised that the energy is not between $E_T \pm \delta$.

Methods related to getting approximate solutions to ELECTRONIC STRUCTURE have complexities ranging from NP-complete to QMA-complete as shown in sections 4 and 5. The complexity of ELECTRONIC STRUCTURE itself is not yet clear. In the literature, only Hamiltonians with more flexibility, namely a local magnetic field, have been demonstrated to be QMA-complete as shown see section 6.1.

3.2 Classical and quantum spin Hamiltonians

The problem of estimating the ground-state energy of Hamiltonians of different forms lies at the intersection of computer science, physics and chemistry. In this section, we define

computational problems related to both classical and quantum spin Hamiltonians. Spin systems have long been known to provide fertile ground for complexity theory^{19,39–41}. Given a Hamiltonian, deciding if there exists a spin configuration, e.g. $(\uparrow\downarrow\cdots)$, that satisfies a certain property, e.g. has energy below a certain threshold, can be difficult even when checking the property for each configuration is easy. Computations designed to decide properties of this sort are often in the complexity class NP. If, additionally, that particular property can embed instances of any other NP properties, then that property corresponds to an NP-complete computational problem.

Computational Problem: ISING. The inputs are N classical spins with possible values ± 1 , trial energy, E_T , and Ising Hamiltonian

$$H_{\text{ising}} = - \sum_{\langle ij \rangle}^{neighbors} J_{ij} S_i S_j \quad (7)$$

having J_{ij} either 0 or ± 1 . The task is to decide if there is a configuration of spins $c = (s_1, s_2, \dots, s_N)$ such that the energy of the Ising Hamiltonian is less than E_T or if all configurations have energy above E_T .

Unlike the other Hamiltonian problems discussed in this review, ISING does not require a promise on the precision because the energy values of the Hamiltonian are integer-valued instead of continuous.

Although some instances of the Ising lattice can be solved analytically, e.g. one-dimensional Ising lattices or the solution by Onsager⁴² for two-dimensional Ising lattices with uniform couplings, Barahona³⁹ showed that problem ISING is NP-hard when interactions are restricted to nearest neighbors on a $2 \times L \times L$ lattice. Since this model is only of tangential interest to electronic structure, details of this proof are omitted. Since the “accept” instances can be verified in polynomial time given a configuration c with energy less than E_T , this problem is in the complexity class NP and hence, is NP-complete. Further results show that all non-planar lattices, even in two-dimensions, are NP-complete⁴⁰.

To provide examples of Hamiltonian problems that are complete for the quantum analogue of NP, we turn to the quantum analogues of the Ising Hamiltonian: quantum spin Hamiltonians. To define similar quantum Hamiltonians with QMA-complete properties, it turns out that one only needs spins with angular momentum of $\frac{1}{2}$ which are often called *qubits*. The interactions are now expressed using tensor products of the Pauli sigma matrices with matrix representations

$$\sigma^x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \sigma^y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad \sigma^z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \quad (8)$$

Additionally, σ^0 is the identity matrix.

A widely studied class of qubit Hamiltonian problems whose computational complexity was first investigated by Kitaev *et al.*⁴¹ is the problem of computing, to polynomial accuracy, the ground state energy of k -spin local Hamiltonians. This set of problems is referred to as k -LOCAL HAMILTONIAN in the computational complexity literature following⁴¹, but we use the name k -LOCAL-SPIN HAMILTONIAN to emphasize the nature of the Hamiltonian.

Computational Problem: k-LOCAL-SPIN HAMILTONIAN. Given k -spin-local Hamiltonian acting on N spins,

$$H_{kQMA} = - \sum_{\substack{D=(d_1, \dots, d_k) \\ C=(c_1, \dots, c_k)}}^m J_{C,D} \sigma_{c_1}^{d_1} \otimes \sigma_{c_2}^{d_2} \otimes \cdots \otimes \sigma_{c_k}^{d_k} \quad (9)$$

where $d_i \in \{x, y, z, 0\}$, c_i labels a spin, there are $m = \text{poly}(N)$ terms, and $|J_{C,D}| \leq 1$, decide if the ground state energy is less than $E_0 - \delta$ or if the ground state energy is greater than $E_0 + \delta$ with $\delta < 1/\text{poly}(N)$ promised that it is not between $E_0 \pm \delta$.

Note that the problem is defined so that the error scales relative to the number of non-zero terms m in equation (9). To understand this, consider the following set of instances with error tolerance $\delta = 0.5$. Consider Hamiltonian, H , with ground state energy 17.7 and Hamiltonian, H' , with energy 17.6. As it stands, the two Hamiltonians cannot be distinguished. However, by including each term in H and H' ten additional times, the ground state energies are now 177 and 176, which can be resolved at error tolerance δ . By considering m as a polynomial in N , one cannot indefinitely rescale the energy to effectively shrink the error tolerance.

The first demonstration of a QMA-complete problem⁴¹ required $k = 5$. Subsequently k was reduced⁴³ to 2. Finally, the problem was shown⁴⁴ to remain QMA-complete even when $J_{C,D}$ is non-zero only if two spins are spatially adjacent on a 2D lattice. The problem can be further restricted by including only a limited set of two-spin interactions and still remain QMA-complete⁴⁵. Complexity results concerning qubit Hamiltonians as well as other variants with higher dimensional spins and more restrictive lattices were recently reviewed by Osborne¹⁹.

3.3 Relationships between spin systems and electronic systems

Since chemists are not necessarily interested in qubit or quantum spin systems, we now discuss connections to systems indistinguishable electrons. In this subsection, we will examine how to embed a system of spins into an electronic Hamiltonian and how to embed electronic systems into spin Hamiltonians. Lastly, we use these connections to give an example of a fermionic system (although not ELECTRONIC STRUCTURE) that is QMA-complete.

Given an electronic Hamiltonian, there is a *orbital pair pseudo-spin* mapping^{46,47} that is used to embed qubit models to the ground state of the electronic systems. To embed N spins, there must be $M = 2N$ orbitals and N electrons, i.e. half-filling, and each quantum spin is identified with a pair of fermionic modes. The embedding requires translating each spin $|q_i\rangle = \alpha|\uparrow_i\rangle + \beta|\downarrow_i\rangle$ to fermionic operators: $\alpha a_{i\uparrow}^\dagger + \beta a_{i\downarrow}^\dagger$. The Pauli operators appearing in the spin Hamiltonian, then become single fermion terms, e.g. $h_{ij}a_i^\dagger a_j$. As important examples consider,

$$\sigma_i^x = |\downarrow_i\rangle\langle\uparrow_i| + |\uparrow_i\rangle\langle\downarrow_i| \leftrightarrow a_{i\downarrow}^\dagger a_{i\uparrow} + a_{i\uparrow}^\dagger a_{i\downarrow} \quad (10a)$$

$$\sigma_i^y = i(|\downarrow_i\rangle\langle\uparrow_i| - |\uparrow_i\rangle\langle\downarrow_i|) \leftrightarrow i(a_{i\downarrow}^\dagger a_{i\uparrow} - a_{i\uparrow}^\dagger a_{i\downarrow}) \quad (10b)$$

$$\sigma_i^z = |\uparrow_i\rangle\langle\uparrow_i| - |\downarrow_i\rangle\langle\downarrow_i| \leftrightarrow a_{i\uparrow}^\dagger a_{i\uparrow} - a_{i\downarrow}^\dagger a_{i\downarrow} \quad (10c)$$

Taking tensor products of the Pauli matrices leads to two-fermion terms, e.g. $h_{ijkl}a_i^\dagger a_j^\dagger a_k a_l$. The final concern is preventing double occupancy within a pair of sites which would invalidate the pseudo-spin interpretation. This is handled⁴⁷⁻⁴⁹ by including an additional term which penalizes invalid electronic configurations. For each pair of electronic modes, the two-fermion penalty

$$P_i = C a_{i\uparrow}^\dagger a_{i\uparrow} a_{i\downarrow}^\dagger a_{i\downarrow} \quad (11)$$

is added to the electronic Hamiltonian. Because penalty $P = \sum_i P_i$ commutes with the fermionic Pauli matrices, the ground state still corresponds to the solution of the spin Hamiltonian. The constant C can be selected as a low order polynomial in the system size to ensure that the ground state remains in a valid pseudo-spin state[†]. Note that we have not explicitly relied on the anti-commutation properties and a nearly identical construction exists for bosonic systems⁵⁰.

A second connection is given using well established techniques developed to translate certain spin systems to simpler non-interacting fermionic systems that can be exactly solved⁵¹⁻⁵³. The Jordan-Wigner transform⁵¹ provides this connection by mapping fermions to spin operators such that the canonical anti-commutation relations are preserved. The Jordan-Wigner transform between N fermionic creation and annihilation operators and the Pauli matrices acting on N spins is given by

$$a_j \Leftrightarrow \mathbf{1}^{\otimes j-1} \otimes \sigma^+ \otimes \sigma^{z \otimes N-j} \quad (12a)$$

$$a_j^\dagger \Leftrightarrow \mathbf{1}^{\otimes j-1} \otimes \sigma^- \otimes \sigma^{z \otimes N-j} \quad (12b)$$

where $\sigma^+ = \frac{\sigma^x + i\sigma^y}{2} = |\downarrow\rangle\langle\uparrow|$ and $\sigma^- = \frac{\sigma^x - i\sigma^y}{2} = |\uparrow\rangle\langle\downarrow|$. The qubit state $|\uparrow \dots \uparrow\rangle$ corresponds to the vacuum state and the

string of σ^z operators, preserve the commutation relations in equation (1) since σ^z and σ^\pm anti-commute. More sophisticated generalizations of the Jordan-Wigner transform, reduce the number and the spatial extent of the spin-spin interactions⁵⁴⁻⁵⁶.

The orbital pair pseudo-spin mapping and the Jordan-Wigner transformation complement each other and will be used repeatedly throughout the remainder of the article. The orbital pair pseudo-spin mapping requires carefully engineered penalties and fixes the total number of orbitals. Thus, it is primarily useful when translating spin systems to electronic systems. By contrast, the Jordan-Wigner transformation is primarily useful in the other direction, that is, when translating an arbitrary electronic system to a quantum spin system. The next subsection gives a concrete example of how these connections are employed when studying the computational complexity of electronic systems.

3.4 Generic local fermionic problems are QMA-complete

The orbital pair construction allows one to immediately ascertain that the ground state energy decision problem for Hamiltonians containing two-fermion interactions,

$$H_{2f} = \sum h_{ij}a_i^\dagger a_j + h_{ijkl}a_i^\dagger a_j^\dagger a_k a_l, \quad (13)$$

is QMA-hard⁴⁸. This follows via a Karp reduction using the orbital pair pseudo-spins as in equation (10). Since only two-spin interactions in equation (9) are required for QMA-completeness⁴³, each spin-spin interaction term translates to two-fermion terms under the pseudo-spin mapping. Since the pseudo-spin construction can be extended to bosonic systems, the two-boson ground state energy decision problem is also QMA-hard⁵⁰.

The ground state energy of the electronic state can be verified to be above or below $E_T \pm \delta$ by a BQP quantum computer given the proof state. This implies that the problem is QMA-complete. We sketch the idea relying on well known results about BQP quantum simulation of chemical systems^{6,7,9}. The Jordan-Wigner transformation is used to translate the two-electron Hamiltonian into a qubit Hamiltonian. If Merlin provides the fermionic ground state[‡], the energy of corresponding qubit state can be determined through simulation^{8,57,58} of the evolution under the qubit version of the two-electron Hamiltonian. This quantum evolution can be simulated efficiently on a quantum computer^{59,60}. The resulting evolution in the time-domain is Fourier transformed to extract the energy of the ground state^{5,8,11,61}, allowing the verifier to determine whether to accept or reject the instance. This argument carries through for the bosonic case as well using the bosonic equivalent of the Jordan-Wigner transform⁵⁰.

‡ The verifier requires multiple copies of the state to ensure that the state has exactly N electrons.

† More precisely, the norm of the Hamiltonian $\|H\|$ is upper bounded by the sum of the individual terms. Since there are, at most, $O(N)^4$ terms the norm of the total Hamiltonian must scale less than a fourth order polynomial in the system size.

4 Hartree-Fock

Hartree-Fock is one of the most important quantum chemistry techniques as it typically recovers about 99% of the total electronic energy. Hartree-Fock is known to be a weak approximation in many instances, but it is the basis for more sophisticated (post-Hartree-Fock) methods which improve upon the Hartree-Fock wave function. Furthermore, Hartree-Fock provides the mathematical framework for the widely adopted notion of molecular orbitals used throughout chemistry.

The implementation of the Hartree-Fock algorithm requires evaluating and manipulating $O(M^4)$ two-electron integrals. When this computation dominates the runtime, the algorithm scales as somewhere between $O(M^2)$ and $O(M^3)$. However, since Hartree-Fock solves a nonlinear eigenvalue equation through an iterative method¹, the convergence of self consistent implementations is the key obstacle that prevents the worst case scaling from being polynomial. The computational complexity result proving that the worst case scaling cannot be polynomial unless P=NP was provided in an unpublished appendix of Schuch and Verstraete⁴⁹ available on the arXiv preprint server.

For the purposes of this article, the Hartree-Fock procedure can be succinctly explained as the minimization of the energy of an N -electron system given M basis functions with the restriction that in the expansion found in equation (2), all C_K are zero except one. Explicitly,

$$E_{HF} = \min_{\Psi = \prod_i (b_i^\dagger)^{k_i} |vac\rangle, \sum k_i = N} \langle \Psi | H_{elec} | \Psi \rangle \quad (14)$$

Here, the single Fock state corresponding to the minimal value of E_{HF} is called the Hartree-Fock state: Ψ_{HF} . The optimized set of creation operators,

$$b_i^\dagger = \sum_j C_{ij} a_j^\dagger \quad (15)$$

place and remove electrons from the set of *molecular orbitals*, $\Psi_j(\mathbf{x}) = \sum_j^M C_{ij} \phi_i(\mathbf{x})$. The formal computational problem HARTREE-FOCK will be defined analogous to the other Hamiltonian problems with a promise given to account for precision.

Computational Problem: HARTREE-FOCK. The inputs are the number of electrons, N , a set of M orbitals, a trial energy, E_T , an error tolerance, $\delta < 1/\text{poly}(N)$, and a two-electron Hamiltonian, c.f. equation (13). The sum of the absolute values of the coefficients is required to scale less than a polynomial in N . The task is to decide if the Hartree-Fock energy $E_{HF} = \langle \Psi_{HF} | H | \Psi_{HF} \rangle$ is less than $E_T - \delta$ or greater than $E_T + \delta$ promised that E_{HF} is not between $E_T \pm \delta$.

4.1 HARTREE-FOCK is NP-complete

To prove that any other NP problem can be mapped to the HF problem (NP-hardness), we can use a Karp reduction to embed instances of ISING into instances of HF⁴⁹. Using the fermionic version of σ^z given in equation (10), H_{ising} only requires two-electron interactions,

$$\begin{aligned} S_i S_j &= \sigma_i^z \sigma_j^z \\ &= (a_{i\uparrow}^\dagger a_{i\uparrow} - a_{i\downarrow}^\dagger a_{i\downarrow})(a_{j\uparrow}^\dagger a_{j\uparrow} - a_{j\downarrow}^\dagger a_{j\downarrow}) \end{aligned} \quad (16)$$

The satisfying assignment of the ISING instance is some spin configuration $|s_1 s_2 \dots s_N\rangle$ for $N = 2L^2$ with s_i as either \uparrow or \downarrow . The correct and exact Hartree-Fock solution for the N -electron wave function should assign $b_i^\dagger = a_{i\uparrow}^\dagger$ when $s_i = \uparrow$ and $b_i^\dagger = a_{i\downarrow}^\dagger$ when $s_i = \downarrow$. Thus, $\text{ISING} \leq \text{HARTREE-FOCK}$ under a Karp reduction.

The promise on the error tolerance, δ , given in the specification of the problem is necessary because of the pseudo-spin representation used in equation (16). Corrections to the energy due to the pseudo-spin representation can be computed using perturbation theory with the unperturbed Hamiltonian given by equation (11) and the converted Ising Hamiltonian, c.f. equation (16), as the perturbation. The first order corrections in small parameter C^{-1} are the Ising energies and the errors due to the pseudo-spin mapping arise at second order in C^{-1} . A coarse estimate for C is obtained by multiplying the number of non-zero terms by the maximum absolute value of a coefficient occurring in equation (13). Since we map the problem to ISING, there are $O(N^2)$ terms in the summation and the maximum absolute value of each term is unity. Hence, C is estimated as $O(N^2)$. So long as $\delta < O(N^{-2})$, the first order corrections occurring at order C^{-1} can be distinguished and the ground state Ising energies can be recovered.

To prove inclusion of HF in the complexity class NP, an algorithm for verifying the energy in polynomial time must be given. If the coefficient matrix C describing the correct orbital rotation is given, then the energy is calculated easily using Slater-Condon rules⁶². Thus, given the answer, the accept/reject conditions are quickly verified. Since the problem can also be quickly verified, HARTREE-FOCK is NP-complete.

5 2-RDM methods

Many computational chemistry algorithms seek to minimize the energy or other properties by manipulating the wave function but to quote Coulson⁶³, “wave functions tell us more than we need to know... All the necessary information required for energy and calculating properties of molecules is embodied in the first and second order density matrices.” Extensive work

has been done to transform this remark into a host of computational methods in quantum chemistry^{64,65}. In this section and the following, we review the prominent computational complexity results related to this body of work.

With respect to the 2-RDMs, it is easy to evaluate properties of the electronic system since the Hamiltonian only contains two-electron terms. However, difficulties arise when determining if the 2-RDM is valid or invalid. Unconstrained optimization of the energy using 2-RDM can lead to ground state energies of negative infinity if the validity of each 2-RDM cannot be determined. While the criteria for validity have recently been developed^{66,67}, deciding the validity of each 2-RDM using such criteria is known as the *N-representability* problem. Mazziotti⁶⁶ also provides a proof that the following problem, *N-REP*, is at least NP-hard. In the next subsection, we follow Liu *et al.*⁴⁸ to demonstrate the stronger conclusion that *N-REP* is QMA-complete.

Computational Problem: N-REP. The inputs are the number of electrons, N , an error tolerance $\beta \geq 1/\text{poly}(N)$, and a 2-RDM, $\mu^{(2)}$. The task is to decide (i) if $\mu^{(2)}$ is consistent with some N -electron mixed state, $\rho^{(N)}$, or (ii) if $\mu^{(2)}$ is bounded away from all 2-RDMs that are consistent with an N -electron state by at least β § promised that either (i) or (ii) is true.

5.1 *N-REP* is QMA-complete

To show QMA-hardness, a Turing reduction is used to show $2\text{-LOCAL-SPIN HAMILTONIAN} \leq N\text{-REP}$. Before proceeding to the Turing reduction, note, since *N-REP* only considers a fixed number of electrons, the one-electron operators are not needed as $a_i^\dagger a_j = a_i^\dagger \left(\sum_k^M a_k^\dagger a_k \right) a_j / (N - 1)$. Each valid 2-RDM can be represented by an $\binom{M}{2} \times \binom{M}{2}$ dimensional matrix with $\binom{M}{2}^2 - 1$ independent parameters¶. A complete set of observables, such as the projection onto each matrix element¶, is then used to characterize the space of 2-RDMs.

The space of valid 2-RDMs is convex; that is if $\mu_1^{(2)}, \mu_2^{(2)}, \dots, \mu_L^{(2)}$ are valid 2-RDMs, then the convex sum $\sum_j n_j \mu_j^{(2)}$ with $\sum_j n_j = 1$ is also a valid 2-RDM. This follows directly from the convexity properties of sets of density matrices and probability distributions.

With access to an oracle for *N-REP*, the boundaries of the convex set of valid 2-RDMs can be characterized. Because the

oracle is used multiple times throughout the verification procedure, this is a Turing reduction. Since convex minimization problems can be solved efficiently and reliably, see e.g.⁶⁸, the QMA-hardness is nearly demonstrated. The last point of concern addressed by Liu *et al.*⁴⁸, are the errors introduced by the promise given on the oracle. To demonstrate that the algorithm remains robust in spite of such errors, the authors use a tailored version of the shallow-cut ellipsoid convex optimization technique⁶⁹.

Let us remark that the required error tolerance follows since the reduction relies on the QMA-completeness of $2\text{-LOCAL-SPIN HAMILTONIAN}$ which includes a promise on allowed the error tolerance.

To demonstrate that *N-REP* is QMA-complete, what remains is demonstrating that *N-REP* is in the complexity class QMA. If $\mu^{(2)}$ is a valid 2-RDM, then, relying on the Jordan-Wigner transform introduced earlier, Merlin can send polynomial copies of the correct N -electron state, $\mu^{(N)}$. The verifier, Arthur, first checks that the number of electrons in the given state is N by evaluating $\text{Tr}[\mu^{(N)} \sum_k a_k^\dagger a_k]$. Then Arthur randomly picks observables from the complete set described earlier and tests that the expectation value of the state $\mu^{(N)}$ and $\mu^{(2)}$ match until convinced. Merlin might try to cheat by sending entangled copies of $\mu^{(N)}$ but it can be proven that Arthur cannot be fooled based on the Markov inequality**. See Aharonov and Regev⁷⁰ for a proof of this fact.

5.2 Restriction to pure states

The pure state restriction of *N-REP* has interesting consequences. Consider, *PURE N-REP*, where the question is now: “Did the 2-RDM come from a N -electron pure state (up to error β)?” This contrasts with the original problem where the consistency questions refers to an N -electron mixed state.

This problem is no longer in QMA since the verifier, Arthur, cannot easily check that the state is pure if the prover, Merlin, cheats by sending entangled copies of $|\Psi^{(N)}\rangle$. If instead, two independent unentangled provers send the proof state, the state from the first prover can be used to verify the purity. The purity is checked using the pairwise swap test⁷¹, a quantum algorithmic version of the Hong-Ou-Mandel effect in quantum optics⁷², on each of the supposedly unentangled states. When the given state is not a product state (or nearly so), Arthur will detect it. If the set of states from the first prover pass the test, the second set can be used to verify that randomly selected expectation values of $|\Psi^{(N)}\rangle$ and $\mu^{(2)}$ match. The complexity class where two verifiers are used instead of one verifier is QMA(2).

§ The appropriate metric is the trace distance, $d_{tr}(A, B) = \|A - B\|_{tr} = \frac{1}{2} \text{Tr}[\sqrt{(A - B)^\dagger (A - B)}]$, which generalizes the distance metric from standard probability theory.

¶ Since each 2-RDM is hermitian, the complex off-diagonal elements occur in pairs and the diagonal elements must be real. Since the trace is normalized to unity, there a reduction of one degree of freedom, leaving $\binom{M}{2}^2 - 1$ independent parameters.

|| Liu *et al.*⁴⁸ used a different set observables inspired by the Pauli matrices with more convenient mathematical properties.

** Given any random variable X , the Markov inequality states $\langle |X| \rangle \geq a \Pr(|X| \geq a)$ with $\langle X \rangle$ defining the expectation value of random variable X and $|X|$ defining the absolute value of X .

6 Density functional theory

In this section, a further reduced description is examined: density functional theory. Density functional theory (DFT) methods are of profound importance in computational chemistry due to their speed and reasonable accuracy^{3,65,73}. In this section, the complexity of the difficult aspects of DFT is shown to be QMA-complete. This was first conjectured by Rassolov and Garashchuk²⁰ and rigorously proven by Schuch and Verstraete⁴⁹.

In DFT, the wave function is replaced by the one-electron probability density, $n(\mathbf{x}) = \rho^{(1)}(\mathbf{x}, \mathbf{x})$. The use of an object in three spatial dimensions to replace an object in $3N$ spatial dimensions without losing any information seems almost absurd, but the theoretical foundations of DFT are well established^{65,74}. The Hohenberg-Kohn theorem^{73,74} proves that the probability density obtained from the ground state wave function of electrons in a molecular system is in one-to-one correspondence with the external potential usually arising from the static nuclear charges. Therefore, all properties of the system are determined by the one-electron probability density.

In the proof of the Hohenberg-Kohn theorem, one of the most important, and, elusive, functionals of the one-electron density is encountered: the universal functional. In all electronic Hamiltonians, the electrons possess kinetic energy and the electrons interact via the Coulomb interaction; only V_{eN} due to the nuclear configuration changes from system to system. Separating these parts allows one to define the universal functional of DFT as

$$F[n(\mathbf{x})] = \min_{\rho^{(N)} \rightarrow n(\mathbf{x})} \text{Tr}[(T_e + V_{ee})\rho^{(N)}], \quad (17)$$

which takes as input the probability density and returns the lowest possible energy of $T_e + V_{ee}$ consistent with the probability density. The nuclear potential energy can be directly determined efficiently using $\int d\mathbf{x} n(\mathbf{x})V_{eN}(\mathbf{x})$ using Gaussian quadratures or Monte Carlo sampling. Regardless of the method used to evaluate the integral, the domain does not (explicitly) increase with the number of electrons. As a decision problem, we have the following:

Computational Problem: UNIVERSAL FUNCTIONAL. The inputs are the number of electrons, N , a probability density $n(\mathbf{x})$, a trial energy E_T and an error tolerance $\delta < 1/\text{poly}(N)$. The task is to decide if $F[n(\mathbf{x})]$ is greater than $E_T + \delta$ or less than $E_T - \delta$ promised that $F[n(\mathbf{x})]$ is not between $E_T \pm \delta$. It is required that the summation over the Hamiltonian coefficients $\sum |V_{ijkl}^{ee}| + |T_{ij}^e| + |V_{ij}^{eN}| + |V_{ij}^{mag}|$ scale less than $\text{poly}(N)$.

6.1 UNIVERSAL FUNCTIONAL IS QMA-COMPLETE

The demonstration that UNIVERSAL FUNCTIONAL is QMA-hard proceeds via a series of reductions. Specifically, Schuch

and Verstraete⁴⁹ show that $H_{2QMA} \leq H_{heisenberg} \leq H_{hubbard} \leq H_{elec}$ where $H_1 \leq H_2$ means that instances of the ground state decision problem for Hamiltonian H_1 can be embedded (Karp reduced) into ground state problem instances of Hamiltonian H_2 . In their proof, the authors utilize the magnetic field to encode the problem instances causing H_{elec} as defined in equation (6) to include an additional local magnetic field, V^{mag} . Note that the local field only affects the spin of the electron and does not require spin-dependent density functionals since the charge and spin do not couple. The Hamiltonian, H_{2QMA} , was listed earlier in equation (9). Again since the reduction relies on the QMA-completeness of the 2-LOCAL-SPIN HAMILTONIAN problem, the promised error tolerance and the upper-bound of the coefficients of the Hamiltonian are required for QMA-completeness.

The $H_{hubbard}$ and $H_{heisenberg}$ Hamiltonians are commonly encountered models in condensed matter physics and are of the form

$$H_{hubbard} = \sum_{\langle ij \rangle} \sum_{s \in \{\uparrow, \downarrow\}}^{M/2} t a_{i,s}^\dagger a_{j,s} + \sum_i^{M/2} U a_{i\uparrow}^\dagger a_{i\downarrow}^\dagger a_{i\downarrow} a_{i\uparrow} - \sum_i^{M/2} \sum_{d \in \{x,y,z\}} B_i^d \sigma_i^d \quad (18)$$

$$H_{heisenberg} = \sum_{d \in \{x,y,z\}} \sum_{ij}^M J \sigma_i^d \sigma_j^d + \sum_i^M B_i^d \sigma_i^d \quad (19)$$

The first Hamiltonian describes an electronic system where the Pauli matrices, σ^d , are expressed using orbital pairs as in equation (10), while the second Hamiltonian describes a system of quantum spin. In both Hamiltonians, different problem instances are embedded using the local magnetic field. The embedding of $H_{heisenberg}$ instances into $H_{hubbard}$ follows directly from the orbital pair mapping described earlier in section 3.3. The remaining two Karp reductions are more involved.

First, let us consider the reduction $H_{2QMA} \leq H_{heisenberg}$. This Karp reduction follows along the same lines used in reducing $kQMA$ from $k = 5$ to $k = 2$ based on *perturbative gadgets*^{43,44} widely used in quantum complexity proofs. A mediator spin splits the state space of the system into a low energy and a high energy sectors. In the low energy sector, the perturbative coupling to the high energy states ‘‘mediates’’ new interactions in the low energy sector. For example, with $H_m = B_m |\varphi_m\rangle\langle\varphi_m|$ acting on the mediator spin and a perturbation $V = \sigma_i^X \sigma_m^X \sigma_j^0 + \sigma_i^0 \sigma_m^Y \sigma_j^Y$, in the low energy space of spin m , there is, at second order, an effective interaction: $\sigma_i^X \sigma_j^Y$. Since the Hamiltonian problems refer to the ground state energy, the high energy sector is not important.

The reduction $H_{2QMA} \leq H_{heisenberg}$ requires:

1. converting arbitrary strength couplings to constant

strength couplings: $J_{12}\sigma_1^A\sigma_2^B \mapsto J(\sigma_1^A\sigma_m^M + \sigma_m^N\sigma_2^B)$;

2. converting inequivalent couplings to equivalent couplings: $\sigma_1^A\sigma_2^B \mapsto \sigma_1^A\sigma_m^A + \sigma_m^B\sigma_2^B$;
3. and finally converting equivalent couplings to Heisenberg interactions: $\sigma_1^A\sigma_2^A \mapsto \sum_d \sigma_1^d\sigma_m^d + \sum_d \sigma_m^d\sigma_1^d$.

The full transformation requires 15 mediator spins where the local field on each of the mediator spins splits the system into low and high energy sectors. The specific local field depends on the interaction desired. The strength of the local field applied to the mediator spin $|B_m|$ is usually very large to ensure that perturbation theory applies. So far this has limited the practical relevance of these constructions; for instance, field strengths required on the final set of mediator spins scales at nearly the 100th power of the system size.

The remaining reduction, $H_{\text{hubbard}} \leq H_{\text{elec}}$, was heuristically known since the Hubbard model phenomenologically describes electrons in solid state systems. Schuch and Verstraete⁴⁹ rigorously demonstrate this reduction by careful accounting for error terms. They begin with a simple model used for studying solids where non-interacting (spin-less) electrons are subjected to a periodic delta function potential with $M/2$ such sites. The orbitals of this system can be solved exactly. The resulting second quantized Hamiltonian has uniform kinetic hopping terms: $\sum_{(ij)} ta_{ij}^\dagger a_j$. After incorporating the electron spin, electron-electron interactions are introduced. The strength of this interaction is rescaled by changing the spatial distance between neighboring sites until only the electrons at the same site can interact. Since each of the $M/2$ identical sites can only support one bound state, the exchange integral must vanish and the Coulomb integral is the same for each interaction yielding $\sum_i U a_{i\uparrow}^\dagger a_{i\downarrow}^\dagger a_{i\downarrow} a_{i\uparrow}$.

Since the Hubbard model and the Heisenberg model require magnetic fields to define problem instances, the electronic Hamiltonian which they consider is not precisely the same as equation (6); this also requires that the functional takes into account the separate spin components of the probability density when performing the minimization in equation (17).

Regardless, a polynomial time solution of UNIVERSAL FUNCTIONAL would also solve the QMA-hard electronic Hamiltonian with local magnetic fields. This follows as the functional is convex

$$\begin{aligned} F[\sum p_j n_j] &= \min_{\rho^{(N)} \rightarrow \sum p_j n_j} \text{Tr} \left[(T_e + V_{ee}) \rho^{(N)} \right] \\ &\leq \sum_j p_j \min_{\rho_j^{(N)} \rightarrow n_j} \text{Tr} \left[(T_e + V_{ee}) \rho_j^{(N)} \right] \quad (20) \end{aligned}$$

and the one-electron probability densities are also convex allowing one to perform convex optimization to find the minimum energy of the QMA-hard electronic Hamiltonian with

the local magnetic fields. Since the conditions for consistency are simple to check for one-electron probability densities⁶⁵, the complications encountered in 2-RDM convex optimization from the approximate consistency conditions are not present.

The inclusion of UNIVERSAL FUNCTIONAL in complexity class QMA follows along the same lines of the generic two-fermion problem discussed in section 3.4. However, in this case, Arthur must subtract the energy of the local field from the total energy to decide if the energy of F is above or below $E_T \pm \delta$.

6.2 Restriction to pure states

The pure state restriction of DFT affects the complexity of computing F . Consider evaluating $F[n]$ where the input density arises from optimization over pure states $|\Psi\rangle$ instead of mixed states as in equation (17). In this case, the universal functional is no longer convex.

$$\sum_i p_i \min_{\Psi_i \rightarrow n_i} \langle \Psi_i | (T_e + V_{ee}) | \Psi_i \rangle \neq \min_{\Psi \rightarrow \sum p_i n_i} \langle \Psi | (T_e + V_{ee}) | \Psi \rangle.$$

The optimization problem would then be NP and an oracle for PURE UNIVERSAL FUNCTIONAL would allow a Turing reduction from QMA to NP. Stated differently, an NP machine with access to an oracle for PURE UNIVERSAL FUNCTIONAL, could solve any problem in QMA. Similar to the 2-RDM case, the restriction to pure states complicates the verification such that two Merlins are required to show that the state is pure and that the proposed solution is correct.

7 Other topics

In this article, we have focused on three pillars of quantum chemistry but there are several other results worth mentioning. Brown *et al.*⁷⁵ showed that calculating the density of states for quantum systems is in the same complexity class as computing the classical partition function. The complexity class of these problems is #P where the problem instances request the number of solutions to problem in complexity class P. Rassolov and Garashchuk²⁰ briefly examined the complexity of the de Broglie-Bohm formulation of quantum mechanics finding that Monte Carlo techniques can be employed to efficiently sample the propagated Hamiltonian-Jacobi equations without the quantum potential. They then suggest that the singularities of the quantum potential will ultimately be the source of difficulties in the quantum propagation. Troyer and Wiese⁷⁶ showed that the sign problem, properly defined, arising when evaluating quantum partition functions with Quantum Monte Carlo is NP-hard. Extension of DFT can be used in the time dependent domain in what is termed time-dependent density functional theory (TDDFT). Tempel and Aspuru-Guzik⁷⁷ recently

demonstrated that one can construct TDDFT functionals for use in quantum computation.

8 Concluding remarks

To close the paper, we make a few remarks on scaling of the error tolerances. In the computational problems investigated, the error tolerance was upper bounded by an inverse polynomial $1/q(N)$ in the system size. In HARTREE-FOCK, it was the pseudo-spin mapping that necessitated the promise, but, in UNIVERSAL FUNCTIONAL and N -REP, the promise is inherited from the 2-LOCAL-SPIN HAMILTONIAN problem. At first glance, it would appear that the error tolerance becomes more restrictive as the system size increases, however this is not necessarily the case.

We focus on the problem k -LOCAL-SPIN HAMILTONIAN since it is at the heart of the reductions for both UNIVERSAL FUNCTIONAL and N -REP. In this problem, the error tolerance, δ , is normalized by the number of terms in the Hamiltonian^{††}, m . Therefore, the error tolerance only shrinks with system size if the number of non-zero terms in the Hamiltonian is constant or grows slower than $q(N)$. In many cases, the number of terms in the Hamiltonian increase with system size. Thus, even if asking for fixed error, e.g. 1 kcal/mol, as the system size increases the promise is still fulfilled and the problem of deciding instances of k -LOCAL-SPIN HAMILTONIAN is QMA-complete.

When the error tolerance is independent of the system size or grows slowly, it is not clear if the problem remains QMA-complete. This is related to the on-going research into possible quantum generalizations of the PCP theorem^{‡‡}. The PCP theorem shows that the task of approximating some NP-complete problems is also NP-complete^{78,79}. A quantum generalization, if it exists, would allow one to show that k -LOCAL-SPIN HAMILTONIAN remains QMA-complete even when the normalized error tolerance is bounded by a constant instead of an inverse polynomial of N . This remains a prominent open question in modern computer science and is discussed more thoroughly by Osborne¹⁹.

With respect to complexity in quantum chemistry, we have examined three computational complexity of different quantum chemistry algorithms: HARTREE-FOCK, N -REP, and UNIVERSAL FUNCTIONAL. The difficulty of these computational chemistry problems relative to problems found in other

fields gives insights into why computational chemistry is difficult. The difficulty of these problems imply that even a quantum computer will be unable to solve all instances since it is believed that BQP does not contain NP. However, even if quantum computers are never constructed, the study of quantum computational complexity gives new insight into the relative difficulty of problems. For instance, characterizing valid two-electron reduced density matrices or evaluating the universal functional of DFT is arguably more difficult than evaluating the Hartree-Fock energy based on the probable separation of complexity classes QMA and NP. The examination of other properties of molecular systems and other problems encountered in quantum chemistry will provide fertile ground for future research into the difficulty of quantum chemistry.

Acknowledgements

JDW and PJJ acknowledge support from the NSF under award numbers 1017244 and PHY-0955518, respectively. PJJ and AAG acknowledge support from the NSF CCI, “Quantum Information for Quantum Chemistry,” (CHE-1037992). AAG additionally thanks HRL Laboratories, The Dreyfus Foundation and The Sloan Foundation for their support. JDW would like to thank The Max-Planck Institute PKS, Dresden and ISI, Torino where parts of this work were completed. The authors wish to thank Z. Zamboris, S. Aaronson, J. Biamonte, D. Gavinsky, and M.-H. Yung for helpful comments on the manuscript. JDW also acknowledges helpful discussions with T. Ito about the PCP theorem.

References

- 1 A. Szabo and N. Ostlund, *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*, Dover Publications, 1996.
- 2 T. Helgaker, P. Jorgensen and J. Olsen, *Molecular Electronic-Structure Theory*, John Wiley and Sons, 2000.
- 3 W. Koch and M. C. Holthausen, *A chemist's guide to density functional theory*, Wiley-VCH, 2001.
- 4 C. J. Cramer, *Essentials of Computational Chemistry: Theories and models*, Wiley, 2004.
- 5 A. Aspuru-Guzik, A. D. Dutoi, P. Love and M. Head-Gordon, *Science*, 2005, **309**, 1704.
- 6 K. L. Brown, W. J. Munro and V. M. Kendon, *Entropy*, 2010, **12**, 2268.
- 7 I. Kassal*, J. Whitfield*, A. Perdomo-Ortiz, M.-H. Yung and A. Aspuru-Guzik, *Ann. Rev. Phys. Chem.*, 2011, **62**, 185–207.
- 8 J. D. Whitfield, J. D. Biamonte and A. Aspuru-Guzik, *J. Mol. Phys.*, 2011, **109**, 735.
- 9 M.-H. Yung, J. D. Whitfield, S. Boixo, D. G. Tempel and A. Aspuru-Guzik, *Adv. Chem. Phys.*, 2012, **In press**.
- 10 R. Feynman, *Int. J. Theoretical Phys.*, 1982, **21**, 467.
- 11 L.-A. Wu, M. S. Byrd and D. A. Lidar, *Phys. Rev. Lett.*, 2002, **89**, 057904.
- 12 I. Kassal, S. P. Jordan, P. J. Love, M. Mohseni and A. Aspuru-Guzik, *Proc. Natl. Acad. Sci.*, 2008, **105**, 18681.
- 13 L. Veis and J. Pittner, *J. Chem. Phys.*, 2010, **133**, 194106.
- 14 J. D. Biamonte, V. Bergholm, J. D. Whitfield, J. Fitzsimons and A. Aspuru-Guzik, *AIP Advances*, 2011, **1**, 022126.

^{††} One may ask why not use the norm of the Hamiltonian? For a Hamiltonian, H , the operator norm is the maximum eigenvalue and computing the maximum eigenvalue is just as hard problem as computing the ground state of $-H$, thus also a difficult problem.

^{‡‡} The term PCP refers to probabilistically checkable proof systems where the verifier is only allowed to randomly access parts of a non-deterministically generated proof. Using only polynomial time and a fixed amount of random bits, the verifier is to validate or invalidate the proof.

- 15 D. Lu, B. Xu, N. Xu, Z. Li, H. Chen, X. Peng, R. Xu and J. Du, *Phys. Chem. Chem. Phys.*, 2012, **14**, 9411–9420.
- 16 A. Aspuru-Guzik and P. Walther, *Nature Phys.*, 2012, **8**, 285–291.
- 17 S. Aaronson, *Sci. Am.*, 2008, 62.
- 18 P. J. Love, *Adv. Chem. Phys.*, 2012, **In press.**
- 19 T. J. Osborne, *Rep. Prog. Phys.*, 2012, **75**, 022001.
- 20 V. A. Rassolov and S. Garashchuk, *Chem. Phys. Lett.*, 2008, **464**, 262–264.
- 21 N. Schuch, M. M. Wolf, F. Verstraete and J. I. Cirac, *Phys. Rev. Lett.*, 2007, **98**, 140506.
- 22 N. Schuch, I. Cirac and F. Verstraete, *Phys. Rev. Lett.*, 2008, **100**, 250501.
- 23 F. Verstraete, V. Murg and J. I. Cirac, *Advances in Physics*, 2008, **57**, 143–224.
- 24 G. K.-L. Chan, J. J. Dorando, D. Ghosh, J. Hachmann, E. Neuscamman, H. Wang and T. Yanai, *Prog. Theor. Chem. and Phys.*, 2008, **18**, 49.
- 25 G. K.-L. Chan and S. Sharma, *Ann. Rev. Phys. Chem.*, 2011, **62**, 465.
- 26 G. K.-L. Chan, *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 2012, **In press.**
- 27 M. Sipser, *Introduction to the Theory of Computation*, PWS Publishing Company, 1997.
- 28 S. Arora and B. Barak, *Computational Complexity: A Modern Approach*, Cambridge University Press, 2009.
- 29 S. Ben-David, B. Chor, O. Goldreich and M. Luby, *Journal of Computer and System Sciences*, 1992, **44**, 193–219.
- 30 A. Bogdanov and L. Trevisan, *Found. Trends Theor. Comput. Sci.*, 2006, **2**, 1–106.
- 31 G. Moore, *Electronic Mag.*, 1965, 4.
- 32 A. Turing, *Proc. London Math. Soc.*, 1937, **1**, 230–265.
- 33 D. Deutsch, *Proc. R. Soc. A*, 1985, **400**, 97–117.
- 34 E. Bernstein and U. Vazirani, *Proc. STOC '93*, 1993, 11.
- 35 B. L. Hammond, W. A. Lester Jr. and P. J. Reynolds, *Monte Carlo Methods in Ab Initio Quantum Chemistry*, World Scientific, 1994.
- 36 W. M. C. Foulkes, L. Mitas, R. J. Needs and G. Rajagopal, *Rev. Mod. Phys.*, 2001, **73**, 33–83.
- 37 W. A. Lester Jr., L. Mitas and B. Hammond, *Chem. Phys. Lett.*, 2009, **478**, 1–10.
- 38 M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, 2001.
- 39 F. Barahona, *J. Phys. A: Math. Gen.*, 1982, **15**, 3241.
- 40 S. Istrail, *Proc. 32nd ACM Symp. on Theory of Comp. (STOC '00)*, 2000, 87.
- 41 A. Kitaev, A. Shen and M. Vyalıy, *Classical and quantum computation*, American Mathematics Society, Graduate Studies in Mathematics, 2002, vol. 47.
- 42 L. Onsager, *Phys. Rev.*, 1944, **65**, 117–149.
- 43 J. Kempe, A. Kitaev and O. Regev, *SIAM J. Computing*, 2006, **35**, 1070–1097.
- 44 R. Oliveira and B. Terhal, *Quant. Inf. Comp.*, 2008, **8**, 0900.
- 45 J. D. Biamonte and P. J. Love, *Phys. Rev. A*, 2008, **78**, 012352.
- 46 C. L. Cleveland and R. Medina A., *Am. J. Phys.*, 1976, **44**, 44.
- 47 A. Auerbach, *Interaction electrons and quantum magnetism*, Springer, 1994.
- 48 Y.-K. Liu, M. Christandl and F. Verstraete, *Phys. Rev. Lett.*, 2007, **98**, 110503.
- 49 N. Schuch and F. Verstraete, *Nature Phys.*, 2009, **5**, 732. Also see arXiv:0712.0483.
- 50 T. Wei, M. Mosca and A. Nayak, *Phys. Rev. Lett.*, 2010, **104**, 040501.
- 51 P. Jordan and E. Wigner, *Z. Phys. A.*, 1928, **47**, 631.
- 52 E. Lieb, T. Schultz and D. Mattis, *Ann. Phys.*, 1961, **16**, 407–466.
- 53 A. Dutta, U. Divakaran, D. Sen, B. K. Chakrabarti, T. F. Rosenbaum and G. Aeppli, arXiv:1012.0653, 2010.
- 54 S. B. Bravyi and A. Y. Kitaev, *Ann. Phys.*, 2002, **298**, 210–226.
- 55 F. Verstraete and J. I. Cirac, *J. Stat. Mech.*, 2005, **2005**, P09012.
- 56 C. D. Batista and G. Ortiz, *Phys. Rev. Lett.*, 2001, **86**, 1092–1085.
- 57 G. Ortiz, J. Gubernatis, E. Knill and R. Laflamme, *Phys. Rev. A*, 2001, **64**, 022319.
- 58 R. Somma, G. Ortiz, J. E. Gubernatis, E. Knill and R. Laflamme, *Phys. Rev. A*, 2002, **65**, 042323.
- 59 S. Lloyd, *Science*, 1996, **273**, 1073–8.
- 60 D. W. Berry, G. Ahokas, R. Cleve and B. C. Sanders, *Commun. Math. Phys.*, 2007, **270**, 359–371.
- 61 A. Kitaev, arXiv:quant-ph/9511026, 1995.
- 62 I. Levine, *Quantum Chemistry*, Prentice-Hall, 2006.
- 63 C. A. Coulson, *Rev. Mod. Phys.*, 1960, **32**, 170.
- 64 *Reduced-Density-Matrix Mechanics: with application to many-electron atoms and molecules*, ed. D. A. Mazziotti, Wiley, 2007, vol. 134.
- 65 R. G. Parr and W. Yang, *Density-Functional Theory of Atoms and Molecules*, Oxford University Press, 1989.
- 66 D. A. Mazziotti, *Phys. Rev. Lett.*, 2012, **108**, 263002.
- 67 D. A. Mazziotti, *Phys. Rev. A*, 2012, **85**, 062507.
- 68 S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.
- 69 M. Grötschel, L. Lovász and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Springer-Verlag, 1988.
- 70 D. Aharonov and O. Regev, *Proc. Annual Symp. on Found. of Comp. Sci.*, 2003, p. 210.
- 71 A. Harrow and A. Montanaro, *Proc. of the 2010 IEEE 51st Annual Symp. on Found. of Comp. Sci. (FOCS '10)*, 2010, 633–642.
- 72 C. K. Hong, Z. Y. Ou and L. Mandel, *Phys. Rev. Lett.*, 1987, **59**, 2044–2046.
- 73 W. Kohn, *Rev. Mod. Phys.*, 1999, **71**, 1253–1266.
- 74 P. Hohenberg and W. Kohn, *Phys. Rev.*, 1964, **136**, B864.
- 75 B. Brown, S. T. Flammia and N. Schuch, *Phys. Rev. Lett.*, 2011, **107**, 040501.
- 76 M. Troyer and U.-J. Wiese, *Phys. Rev. Lett.*, 2005, **94**, 170201.
- 77 D. Tempel and A. Aspuru-Guzik, *Sci. Reports*, 2012, **2**, 391.
- 78 S. Arora and S. Safra, *J. ACM*, 1998, **45**, 70–122.
- 79 S. Arora, C. Lund, R. Motwani, M. Sudan and M. Szegedy, *J. ACM*, 1998, **45**, 501–555.