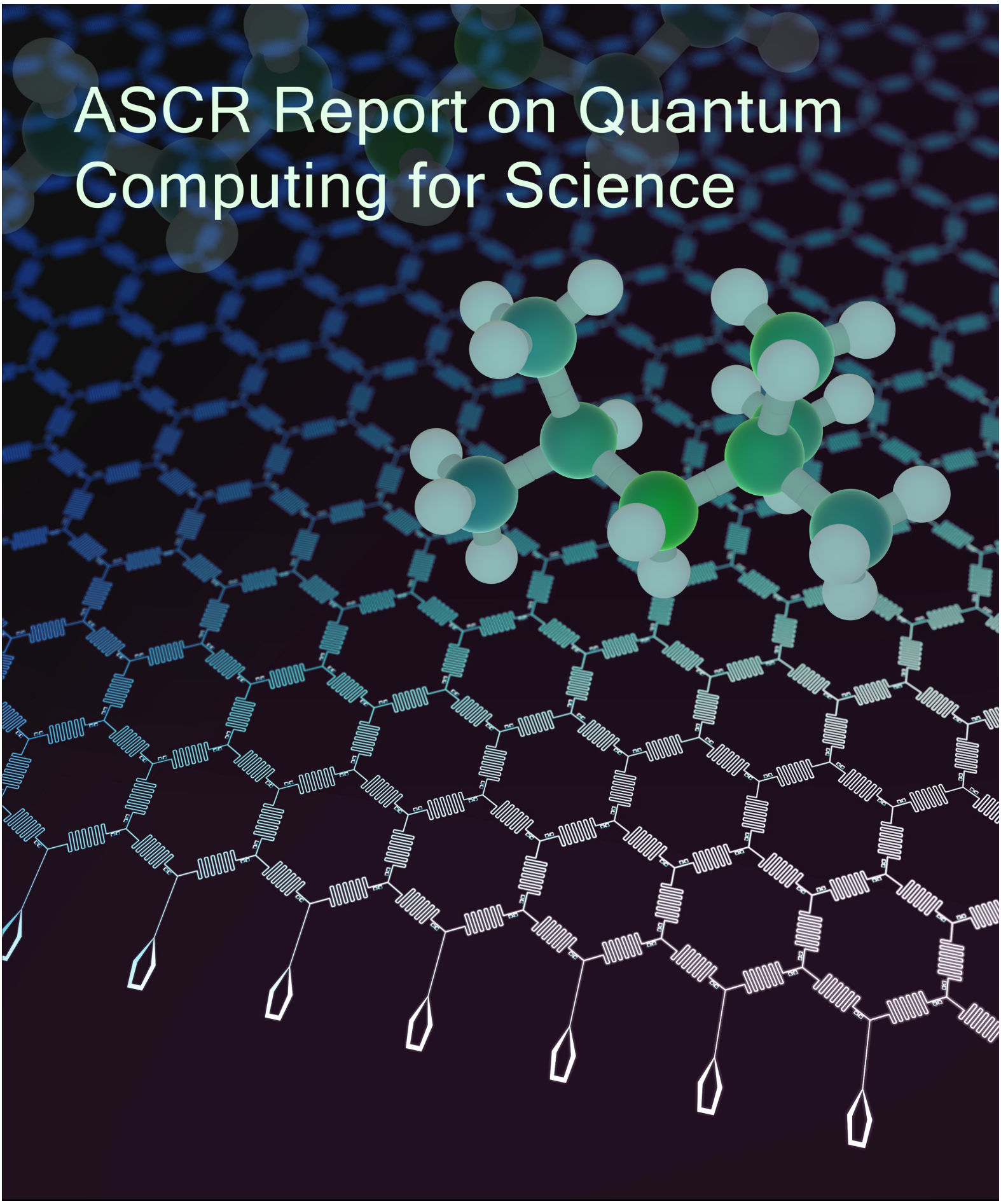
ASCR Report on Quantum Computing for Science



ASCR Report on Quantum Computing for Science

Sponsored by
U.S. Department of Energy
Office of Science
Advanced Scientific Computing Research Program

Quantum Computing Working Group
Alán Aspuru-Guzik, Wim van Dam, Edward Farhi, Frank Gaitan, Travis Humble, Stephen Jordan, Andrew Landahl, Peter Love, Robert Lucas, John Preskill, Richard Muller, Krysta Svore, Nathan Wiebe, Carl Williams

DOE POC
Ceren Susut

Abstract:

This report details the findings of the DOE ASCR Workshop on Quantum Computing for Science that was organized to assess the viability of quantum computing technologies to meet the computational requirements of the DOE's science and energy mission, and to identify the potential impact of quantum technologies. The workshop was held on February 17-18, 2015, in Bethesda, MD, to solicit input from members of the quantum computing community. The workshop considered models of quantum computation and programming environments, physical science applications relevant to DOE's science mission as well as quantum simulation, and applied mathematics topics including potential quantum algorithms for linear algebra, graph theory, and machine learning. This report summarizes these perspectives into an outlook on the opportunities for quantum computing to impact problems relevant to the DOE's mission as well as the additional research required to bring quantum computing to the point where it can have such impact.

Executive Summary

Quantum computing uses computational elements that obey quantum mechanical laws to potentially provide transformative changes in computational power for certain problems of interest to the U.S. Department of Energy. The DOE Office of Science Advanced Scientific Computing Research (ASCR) program sponsored a workshop to assess the viability of quantum computing technologies to meet computational requirements in support of the DOE's science and energy mission. Discussion at the workshop focused on models of quantum computation and programming environments and applications of quantum computing technology to physical science domains and applied mathematics domains relevant to DOE's science mission. This report summarizes the discussions held during the workshop. It includes a broad overview of quantum computing and concludes a summary of research opportunities in quantum computing.

Peter Shor's 1994 breakthrough discovery of a polynomial time quantum algorithm for integer factorization sparked great interest in discovering additional quantum algorithms and developing hardware on which to run them. The subsequent research efforts yielded quantum algorithms offering speedups for widely varying problems, and several promising hardware platforms for quantum computation. These platforms include analog systems (usually cold atoms) used for simulating quantum lattice models from condensed-matter and high-energy physics, quantum annealers for combinatorial optimization, boson samplers, and small-scale noisy prototypes of digital gate-model quantum computers. Potential applications of the various forms of present-day and near-term quantum computation to DOE's mission were explored at the workshop.

In the longer term, the emergence of scalable, fault-tolerant, digital quantum computers offers a new direction for progress in high performance computing as conventional technologies reach their fundamental limitations. Quantum speedups have been discovered for a number of areas of DOE interest, including simulations for chemistry, nuclear and particle physics, and materials science, as well as data analysis and machine learning. In addition, quantum speedups have been discovered for basic primitives of applied mathematics such as linear algebra, integration, optimization, and graph theory. These demonstrate the potential of quantum computers to yield better-scaling methods (in some cases exponentially better) for performing a wide variety of scientific computing tasks. Practical realization of this potential will depend not only on advances in quantum computing hardware but also advances in optimizing languages and compilers to translate these abstract algorithms into concrete sequences of realizable quantum gates, and simulators to test and verify these sequences. The development of such software has recently seen rapid progress, which can be expected to continue given sufficient support.

Quantum computation presents a number of research opportunities for ASCR. These opportunities include development and benchmarking of near-term applications of

quantum hardware, development of programming environments, languages, libraries, compilers, and simulators for quantum computers, and research and development on quantum algorithms for physical simulation and applied mathematics.

The consensus of participants in the workshop is that quantum computing has reached a level of maturity that warrants considering how it will impact the DOE mission in the near and long term. As summarized below, the workshop participants encouraged the community supported by DOE to investigate how quantum computers could be used to perform quantum simulations, how quantum algorithms could be leveraged to solve problems in applied mathematics, and how quantum computing devices could be made accessible to a broad range of scientists and engineers.

Table 1: Summary of quantum computing research opportunities

- Quantum Simulation: Solve problems in chemistry, materials science, and nuclear and particle physics.
    - Research present-day special purpose analog quantum simulators.
    - Develop and optimize simulation algorithms for future general-purpose digital quantum computers.
- Quantum Algorithms for Applied Mathematics: Develop speedups for the fundamental primitives of applied and computational mathematics.
    - Research quantum algorithms for universal digital quantum computers to speed up primitives such as linear algebra, optimization and graph theory.
    - Investigate the potential of quantum annealers to outperform classical computing for optimization.
- Models of Computation and Programming Environments: Develop software infrastructure for quantum computation.
    - Develop high-level languages for implementing quantum algorithms and compilers to translate quantum programs into optimized quantum circuits.
    - Develop tools to simulate and debug quantum programs.
    - Formulate metrics for assessing, comparing, and benchmarking quantum computations.
    - Quantum co-design: adapt algorithms to hardware and vice-versa.

Table of Contents

I. Introduction

High-fidelity modeling and simulation of physical systems is critical for the U.S. Department of Energy (DOE) to address some of the most challenging problems in energy, the environment, and national security. Problems range from understanding the basic science of new materials to the complex interactions that occur over global length scales. To address these scientific challenges, the DOE has a long history of advancing computation and increasing the capabilities for its high-performance computing (HPC) systems. However, current approaches to increase HPC capability have begun to stall due to constraints on processor technology and system complexity. In particular, longstanding computational methods with the classical von Neumann model processor implemented in CMOS technology have hit development walls with respect to power, processor frequency, and communication. The impact that these limitations will have on future HPC development appears bleak. While computing at exascale appears possible, albeit challenging to realize, the ability to develop HPC systems beyond exascale is highly uncertain. This forces the consideration of alternative computing technologies and how they might be leveraged for future scientific inquiries.

A potential "over-the-horizon" technology for future computing systems is quantum computing [1, 2]. As detailed below, quantum computing offers a fundamentally new approach to computation that promises capabilities not available with today's existing transistor-based processing. So far, the theory of quantum computing has found significant speed-ups to a few prominent algorithms in modeling, simulation and mathematics, and experimental efforts in quantum computer science have recently made great strides demonstrating crude quantum algorithms to solve modest problems in physical simulation and applied mathematics. In addition, it is believed that the operation of an idealized 100-qubit quantum computer may exceed the simulation capabilities of even future exascale computers. This suggests that quantum computers may have the potential to enable some aspects of computational science to progress far beyond exascale.

Therefore, it is pertinent to assess whether the potential of quantum computing is both feasible and practical for DOE mission needs. Solving the wide variety of computational problems addressed by the DOE would require quantum computing to support a robust and versatile set of algorithms, software and hardware architectures. Fundamental questions in computer science and mathematics need to be addressed in order to develop quantum computing into this robust computational platform. The timeline for this research effort is likely to be lengthy, in part because of the concurrent development of quantum computing hardware but also because of the underdeveloped potential of quantum computer science.

As a primary stakeholder in future HPC, the DOE is in a unique position to steer the growth of quantum computer science to meet future mission needs. Breakthrough developments in quantum software, architectures, and algorithms can support preparation of new computing strategies including beyond exascale. This offers the

DOE the opportunity to maintain its long history of being at the forefront of computing technology by guiding the development of quantum computing software and hardware. Indeed, similar "co-design" strategies have been used by the DOE to develop next-generation processors and software infrastructures for exascale computing systems. The reward for embracing this quantum computing co-design opportunity is that the DOE will be strategically prepared for its leadership position in modeling and simulation as quantum computing technology matures. This will permit DOE to address problems of ever-increasing complexity beyond what current-scale or even exascale conventional computing may provide.

The DOE Office of Science Advanced Scientific Computing Research (ASCR) Program sponsored a workshop to assess the viability of quantum computing technologies to meet computational requirements in support of the DOE's science and energy mission. The Workshop on Quantum Computing for Science was held on February 17-18, 2015 in Bethesda, MD. The workshop explored the viability of quantum computing for addressing DOE mission problems, the impact that the quantum technologies are expected to have, and the challenges and opportunities that are anticipated in adopting quantum computing technologies and developing the related infrastructure. Position papers were solicited from across university, industrial, and government partners. The submitted papers were then invited to make a short presentation in one of three topical areas. The three topical areas were:

1. Models of quantum computation and programming environments, including computational models (quantum walks, gate- and Hamiltonian-based computation), topological computing, error correction and fault tolerance, programming and compiling platforms, control theory, resource requirements, computational complexity theory.
2. Physical science applications relevant to DOE's science mission including but not limited to quantum field theories, lattice models (Hubbard, Ising, QCD), quantum chemistry and molecular structure, materials manufacturing, genomics, complexity and thermalization.
3. Applied mathematics topics including potential quantum algorithms for linear algebra, numerical integration, optimization, and graph theory.

This report summarizes the technical and programmatic discussions held during the workshop by offering overviews of the potential for quantum computing in the three topical areas. It includes a broad overview of quantum computing and concludes with a summary of research opportunities for quantum computing.

II. Overview of Quantum Information and Quantum Computing

Quantum computing promises dramatic changes to many aspects of information technology. Standard (hereafter *classical*) computing operates on bits; typically bits are expressed in hardware as voltage levels that define the 0 or 1 values fundamental to a computing system. There are a variety of classical computing devices based on circuits of transistors that perform operations on these voltage values to affect computations on the corresponding bits.

One of the triumphs of 20th century physics was understanding how physics at the subatomic level is different from our everyday experience. The resulting quantum mechanical rules were found to operate differently than similar rules from classical physics. More recently, these same quantum mechanical laws were applied to the principles of information. Quantum computing and quantum information sciences now define the academic disciplines that consider how quantum mechanical rules impact computation and information theory. For example, quantum bits (*qubits*) are significantly different than classical bits: a qubit is defined as any of the linear *superposition* states $\alpha|0> + \beta|1>$, with |0> and |1> the computational basis, α and β complex numbers such that $|\alpha|^2 + |\beta|^2 = 1$, and the extreme classical values of 0 and 1 that correspond to $\alpha = 1$ and $\beta = 1$, respectively. Moreover, *entangling* operations are possible that create correlations between two states such that the resulting state cannot be factored into a product of the individual states. Corresponding to the digital logic of classical computing is a set of single- and two-qubit operations (often called *gates*) that can create superpositions and entangling operations and can be shown to form a basis for universal computation. Several algorithms, including integer factorization, unstructured search, and the simulation of quantum many-body systems, have been shown or are believed to be more efficient using qubits. This section will consider the quantum computational primitives that produce this efficiency and their implications for computation.

A. Quantum Computational Primitives

1. Quantum Walks

Classical random walks provide a general framework for exploring possibly large spaces by using simple local displacement rules. They have been applied (as algorithms) successfully in many different contexts such as finding satisfying assignments in k-SAT problems [3]. Motivated by the success of their classical counterparts, quantum walks now play a central role in several quantum algorithms (for a review see [4], [5]). The quantum walker begins with a localized wave function, and its subsequent evolution—be it discrete time or continuous time—is generated by the application of unitary operators that allow the walker to explore the space. Unlike the classical random walker whose state is described by a probability distribution over the positions, the quantum walker's state can be in a superposition of positions. This key difference has been used to develop quantum algorithms that provide polynomial as well as exponential speedups, over classical algorithms. For example, in many classical random walk algorithms, a fast hitting time, i.e., the first time the walker arrives at a subset of the state space, is crucial to

solve the problem. A quantum walk can give rise to an exponential separation between the classical hitting time and the quantum hitting time [4]. Ultimately, a quantum walk can give rise to an exponential speedup for some oracular problems [6]. Grover's unstructured search algorithm [7], which gives a quadratic speedup over its classical counterpart, although not strictly a quantum walk algorithm, can be interpreted as such [8], and this new viewpoint has led to important generalizations [9]. Ultimately, the quantum walk can be viewed as a universal model of quantum computation; any quantum algorithm can be recast as a quantum walk [10].

2. Quantum Fourier Transform and Phase Estimation

The efficiency of the quantum Fourier transform is an integral part of many quantum algorithms, such as Shor's factoring algorithm [2], quantum phase estimation [11], and the hidden subgroup problem for finite Abelian groups [11]. It is the quantum analogue of the discrete Fourier transform; the input of the algorithm is a quantum state with weights (in the computational basis) representing the input of the discrete Fourier transform, and the output is a new quantum state whose weights are the discrete Fourier transform (note that both input and output are in a quantum superposition state). The most efficient quantum Fourier transform algorithm to date requires only $\mathcal{O}(n \log(n))$ gates [12], where n is the number of qubits. This is in contrast to the classical algorithm that requires $\mathcal{O}(n\ 2^n)$ operations for n bits.

Quantum phase estimation allows for the estimation of the eigenphase of an eigenvector of a unitary operation U. To accomplish this, the initial state, which is the eigenvector of U, is operated on via a sequence of powers of U followed by the inverse quantum Fourier transform. This is efficient, assuming that the powers of U can be implemented efficiently as quantum circuits. Phase estimation is an important subroutine in many quantum algorithms, such as Shor's algorithm [2] and solving well-conditioned sparse systems of linear equations [13].

3. Topological Quantum Computation

Universal quantum computation can, in principle, be performed by "braiding" a class of particles called "anyons". Anyons are not believed to exist in nature as fundamental particles, but they are believed to arise as quasiparticle excitations in certain effectively two-dimensional low temperature condensed-matter systems such as those that exhibit the fractional quantum hall effect. Topological quantum computation is of interest for two main reasons. First, topological quantum computation is a promising architecture for the implementation of quantum computers due to its intrinsic robustness against control error and environmental noise. Second, topological quantum computation provides a conceptual framework for the design of new quantum algorithms. These quantum algorithms can be run efficiently on any universal quantum computer, regardless of whether the architecture is, for example, topological or circuit-based. Quantum algorithms for approximating topological invariants such as Jones polynomials of knots and

Turaev-Viro invariants of three-dimensional manifolds arose quite naturally within the framework of topological quantum computation, whereas they might never have been discovered by thinking directly in terms of circuit-model quantum computation.

B. Computational Complexity

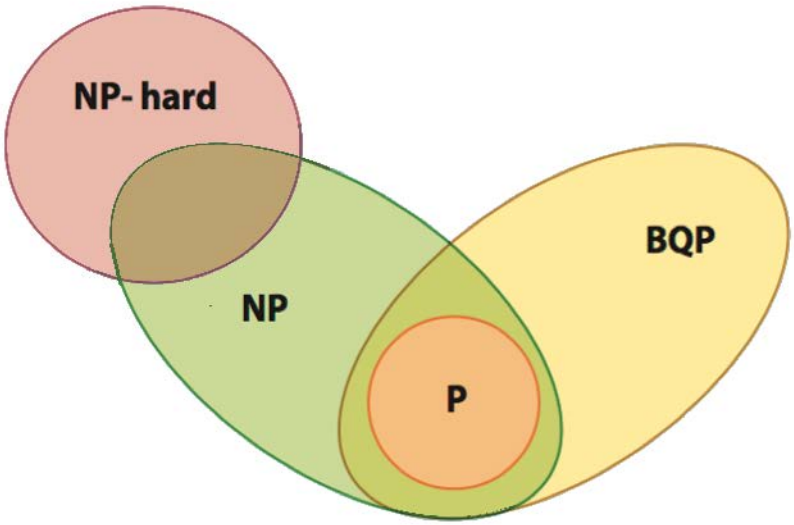


Figure 1: Anticipated relationship between complexity classes relevant to quantum computing.

Complexity classes are a useful way to categorize a set of problems based on their resource complexity. They usually have a definition of the form: "the set of problems with input size n that can be solved by machine M using $\mathcal{O}(f(n))$ resource R." For example, the class P is the set of decision problems (problems with a yes-no answer depending on the input values) that can be solved by a deterministic Turing machine using polynomial time. We can contrast this with the class NP, which is the set of decision problems where the 'yes' answer can be verified in polynomial time by a deterministic Turing machine. It is clear that P is contained in NP, and it is widely believed but unproven that NP is strictly larger than P.

An important class of problems is the NP-hard problems, to which all problems in NP can be reduced in polynomial time. Thus, if any NP-hard problem were solved by a polynomial-time algorithm, then all problems in NP would become efficiently solvable. Although no proof has been obtained, mathematical evidence strongly suggests that neither quantum nor classical computers can solve worst-case NP-hard problems in polynomial time. Many problems of great importance to science and industry are NP-hard. Some important examples include combinatorial optimization problems such as the travelling-salesman problem and Boolean satisfiability. Due to the overwhelming importance of NP-hard problems, much effort within the classical and quantum computer science communities has gone into development of polynomial-time approximation algorithms, polynomial-time heuristic algorithms for average-case instances, and improved exponential-time algorithms for worst-case instances.

The class BQP is the class of decision problems solvable by a quantum computer in polynomial time. Because quantum computers are inherently probabilistic, the definition of BQP allows a small probability of failure. (This probability can be made exponentially small by repeating the algorithm). BQP is the quantum analogue of the class P. This is clearly an important class of problems for quantum computation since it is hoped that many problems that belong in this class cannot be computed efficiently using a classical algorithm. Examples of problems that can be solved in polynomial time by quantum computers but apparently not by classical computers include integer factorization [2], computing discrete logarithms [2], and calculating the Jones polynomial at any primitive root of unity [14].

C. The Scientific Problems That Quantum Information Can Address

Since its inception, quantum computing has been naturally tied to simulating properties of physical systems. In 1982, in what is perhaps the first paper on quantum computing, Feynman noted [1] that classical computing systems would require exponential resources to simulate quantum systems. Since quantum systems can presumably simulate themselves, Feynman suggested that "the computer itself be built of quantum mechanical elements which obey quantum mechanical laws" [1]. In other words, quantum processors could simulate other quantum systems without the exponential overhead required for classical processors. This anticipated exponential speedup for quantum simulation remains one of the primary drivers behind the development of quantum computing resources. Furthermore, the linear algebra underlying the propagation of a quantum state closely resembles both solving a linear system and finding the eigenvalues of linear operators, suggesting that a variety of problems in applied math might also yield quantum speedups on quantum hardware. Given the broad nature of the DOE's mission, the potential enhancement of high-fidelity physical simulation, linear algebra, optimization, and other direct applications of quantum primitives offer numerous opportunities. The report that follows outlines these opportunities for the DOE and ASCR communities.

III. Advances Required For Quantum Computation to Transform Scientific Computing

A. Models of quantum computation and programming environments

In order to turn mature quantum technologies into information-processing devices that can meet DOE needs, it is paramount to develop comprehensive software architectures for quantum computers. Such a platform will allow one to map quantum algorithms expressed in a high-level programming language into fault-tolerant quantum computation primitives and finally onto quantum hardware operations. The software environment will also allow users to effectively program quantum algorithms and optimize them based on various resource constraints, such as a given quantum hardware layout, a restricted number of available qubits, or a desired computation time. The platform should also include a simulation environment, allowing one to assess the performance and correctness of those quantum algorithms executed in a small scale prior to full execution on a scalable, programmable quantum computer. Examples of objectives within these categories include the following:

Enabling reliable quantum computation:

- Discover and develop hardware-compatible models of quantum computation.
- Discover and develop methods for fault-tolerant quantum computing.
- Discover and develop methods for optimizing an algorithm expressed in a given model of quantum computation.

Programming quantum algorithms:

- Develop effective languages with which to express quantum algorithms at both high- and low-levels of abstraction.
- Develop automatic methods of synthesizing a classical reversible circuit into an efficient quantum circuit.
- Develop a series of software tools for, e.g. automatic optimization, scheduling, layout, success probability optimization, etc.
- Develop theorem proving systems to handle verification of a quantum program.
- Develop effective debugging quantum programming environments.

Assessing quantum computers:

- Establish benchmarking quantum computing tests to validate and verify performance.
- Develop methods for emulating features of quantum computers with classical computers.
- Develop automatic methods for estimating resource consumption of a given quantum program (quantum algorithm), most notably in terms of quantum

device implementation requirements such as number of qubits and quantum gates.

In order to have viable software architecture choices, a number of research problems in each of these areas needs to be addressed. These research problems do not fall naturally into one of the existing program subdivisions of the DOE ASCR. Solving these problems requires a different skill set both from a technical and a program management perspective than existing DOE ASCR subdivisions. Nevertheless, addressing these research challenges is vital for DOE ASCR supported communities to exploit the burgeoning development of quantum computing technology for DOE's modeling and simulation needs in the physical sciences.

1. Enabling Quantum Computation

The most fundamental challenge facing quantum software architectures is discovering and developing methods to enable quantum computation in the first place. The two chief hurdles to overcome are discovering and developing viable quantum computation models and discovering and developing fault-tolerant quantum computing strategies. The choices for each then become integrated at the lowest levels of the overall software architecture.

a. Models of quantum computation

Although several theoretical models of quantum computation exist and are well-studied, such as quantum circuits, topological quantum computation, dissipative quantum computing, quantum walks, and the adiabatic quantum computing model, each model has its pros and cons in the context of an actual hardware implementation. The space of possible quantum computational models is far from fully charted, and developing models in a co-design approach with quantum hardware development may benefit both.

In the following, we give three examples of where co-design has led to new quantum computational models and fresh insights into harnessing existing quantum technologies for information processing tasks. These examples also point to future research opportunities where meaningful progress could be made.

Adiabatic quantum computing

The adiabatic quantum computing (AQC) model was originally proposed as a method to solve discrete optimization problems; it was not considered as a model of computation in its own right [15]. By adapting a quantum-walk construction by Feynman [16], Kitaev not only proved that the AQC model can be made universal, so that it can solve BQP-complete problems, but in the course of his proof, he established the complexity class now known as QMA, which is the quantum analogue of the class NP (or more correctly, MA), namely the class of problems for which a "yes" answer can be verified efficiently with high probability on a quantum

computer [17]. A series of papers soon followed simplifying the nature of AQC Hamiltonians that could achieve universal AQC, suggesting hardware design possibilities for future quantum computers [18-24]. There is an opportunity to close the co-design loop available by building machines that realize these proposed architectures.

In the absence of universal AQC machines, there has been a flourishing research environment surrounding AQC for optimization problems. AQC confined to problems in this setting is typically called "quantum annealing." D-Wave Systems, Inc. has built machines that it bills as quantum annealers; questions surrounding these machines have led to the development of numerous assessment methods for "quantumness" in information processing devices [25-34]. This continues to be an active research area and any attempt to reference all work in this topic is bound to be incomplete; progress here is expected to translate to other quantum computational models as well.

Practical limitations of the D-Wave machines have led to a better appreciation for the need for effective quantum compiling and effective quantum error correction. In the former case, the "embedding problem" has been identified as a key bottleneck: this is the problem of mapping a computational task of interest to the machine; coming up with the optimal embedding is itself an NP-hard problem, so there is ample room for R&D into heuristics [35-37]. In the latter case, the dramatic improvements in performance that have been observed by using simple quantum and classical error-correcting codes in D-Wave machines suggest that additional research into this area will pay big dividends [38-42]. The initial optimism of bringing conventional quantum error correction to the AQC model was soon tempered by a series of no-go results for practical fault-tolerant quantum computing in the AQC model [39, 43-47]. As is often the case in science, this critical thinking pointing to potential engineering shortcomings led to more creative approaches. By expanding the AQC model to allow for information processing in degenerate as well as nondegenerate groundspaces, a variant of the AQC model known as holonomic quantum computing (HQC) has been developed into a fully fault-tolerant computational model [48-50]. Current research along these lines focuses on streamlining fault-tolerant HQC protocols to accelerate the possibility of experimental demonstrations.

Finally, the mere existence of the quantum annealing model itself has recently inspired a new approximation algorithm that appears to perform better than any classical algorithm can [51, 52]. This is an example of where a computational model has inspired a new quantum algorithm that can in principle be translated to any other quantum computational model, such as the quantum circuit model. Investing in R&D in alternative quantum computational models like AQC can foster algorithmic advances that transcend the model itself; the well of new quantum algorithms is expected to be far from dried up at this point.

Dissipative quantum computing

Like the AQC model, the goal of dissipative quantum computing (DQC) model is to operate in a manifold of states separated from the rest by an energy gap. This model has only recently been explored, but already quantum simulation [53, 54] and, more generally, universal quantum computing have been proven to be possible in the model [55]. Because dissipative dynamics offer new control possibilities, this model opens up new co-design possibilities with hardware developers; the full geometric nature of quantum computation in this model has only recently been unraveled [56]. Additionally, it may inspire new quantum algorithms. One hope is that the model may also offer a certain intrinsic robustness to noise in a real implementation that would reduce the complexity of quantum error correction that is needed for a fault-tolerant implementation [57]. Many low-hanging research fruits exist for this new model of quantum computation.

Topological quantum computing

Topological quantum computing (TQC) is a model of quantum computing in which transformations are generated by anyons as they braid around each other in spacetime [58-60]. Anyons are particles whose exchange statistics differ from bosons or fermions in that they can generate "any" phase (or, more generally, even nontrivial unitary transformations) when exchanged. Anyons can arise in quantum systems ranging from fundamental particles in quantum field theories to effective excitations in condensed matter systems [61-66]. Over a decade ago, Fields-medalist Mike Freedman and collaborators proved that this model is universal for quantum computation [67]. Since then, a number of groups around the world have been inspired by the model to build systems exhibiting such anyons, such as Majorana fermions in solid-state systems [68-70]. The TQC model has further inspired numerous quantum algorithms such as additive approximations of Turaev-Viro invariants [71-74] and Jones Polynomial invariants [14, 67, 75-80]. The model has also inspired numerous new quantum error-correcting methodologies, including surface codes [81], color codes [82], Turaev-Viro codes [83], and quantum double codes [84]. TQC continues to inspire new software architecture ideas, ranging from quantum compiling [85-89] to fault-tolerant quantum computing protocols [90-92] Continued investment in TQC can be expected to bring quantum computing applications of interest to the DOE closer to realization.

b. Fault-tolerant quantum computing methods

Because physical instantiations of quantum computers are expected to be exceptionally sensitive to noise, fault-tolerant design is expected to be necessary for any software architecture, regardless of which computational model a device realizes. Fault-tolerance methods for the quantum circuit model are well-developed, but their cost is high: most estimates suggest that over 90% of the resources of a quantum computer might need to be allocated to correcting its own faults rather than advancing quantum algorithms [93-95]. Research into methods

for reducing this overhead in the circuit model is therefore sorely needed, as is research into fault-tolerance in other models of quantum computation where the overheads required might be far less.

In the following, we give examples of what some of the challenges in the research frontier are in developing fault-tolerant quantum computing methods. The list is far from complete; it is meant merely to highlight some of the key issues that need to be resolved. In particular, we discuss how the cornerstone of fault-tolerant design, quantum error-correcting codes, might be brought to the adiabatic and holonomic quantum computing models; how ideas from topological quantum computing are inspiring more resource-efficient methods for fault-tolerant design in the quantum circuit model, and how advances from quantum control theory are helping to bridge the gap between quantum hardware capabilities and quantum fault-tolerant design requirements.

Error correction for adiabatic and holonomic quantum computers

Developing error correction for Hamiltonian-based quantum computing models such as the AQC model is difficult because good codes require high-weight logical operators that are impractical to realize with Hamiltonians. Nevertheless, even using small quantum codes can have a positive impact. Jordan *et al.* developed the first AQC codes [38], and since then several others have also been developed [39, 41, 42, 96]. Some of these have been applied to actual hardware, with an obvious improvement in performance [41]. Developing new strategies for suppressing errors in the AQC model continues to be an active research area with clear payoffs.

The related holonomic quantum computation (HQC) model [97] utilizes Hamiltonian evolutions with degenerate, instead of non-degenerate ground spaces. In this model, it is possible to realize a much richer set of quantum error correction possibilities, and in fact full fault-tolerant quantum computation is possible by suitable Hamiltonian evolutions [48, 50, 98, 99]. An example of such an evolution is depicted in Figure 2. Developing improved error-correction protocols for Hamiltonian-based quantum computational models is an active research area that is yielding results that can fold into overall quantum software architectures for hardware realizing these computational models.

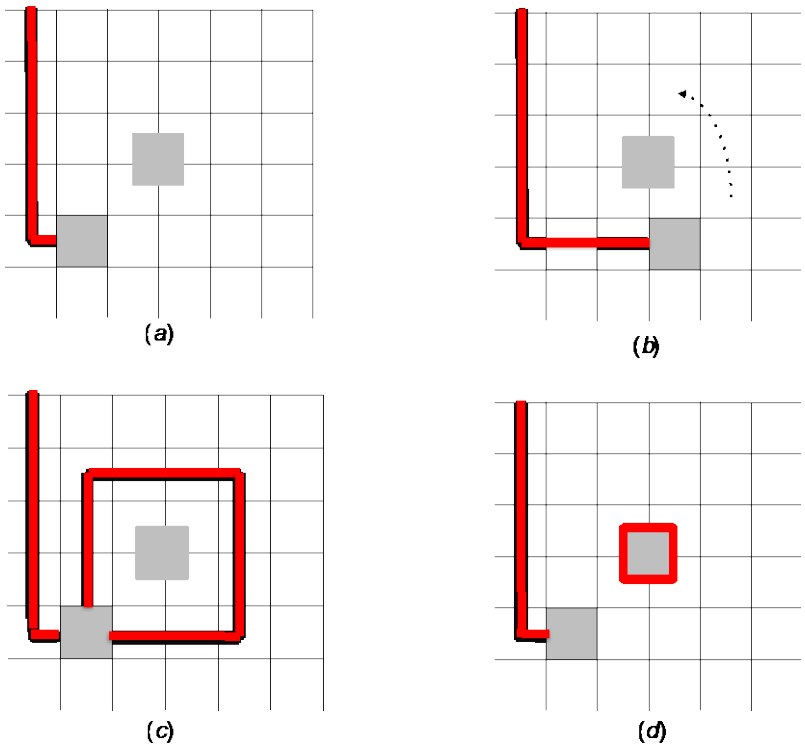


Figure 2: A Hamiltonian in which some terms are inactive (gray) in (a) is adiabatically evolved through a sequence of Hamiltonians in which other terms are turned off (b-d). The net effect is an encoded logical operation on the ground space of the system.

Topological quantum codes and multi-qubit block code extensions

Topological quantum codes such as surface codes and color codes are among the most favored codes at the core of quantum software architectures considered today. Devising new methods for programming with and optimizing such codes as well as managing the classical computing associated with using such codes are key challenges requiring computer science and mathematics expertise. Much of the improvements in these areas to date have been generated by hand, such as improvements to surface-code magic-state distillation methods [100-102] and minimum-weight perfect matching syndrome decoding algorithms [103-108]. By bringing to bear automated methods for optimization and tools and techniques well-known in the computer science and mathematics communities, significant improvements may be possible [109].

Part of the current research frontier with these codes is developing lower-overhead block codes that share as many of the desirable properties of topological codes as possible. For example, quantum low-density parity check (LDPC) codes offer fast syndrome extraction at the cost of non-local data access [110-112]. Gauge-fixing methods via code deformation offer methods for realizing a universal set of gates without having to resort to costly magic-state distillation [113-116]. Finally, clever multi-qubit block-codes augmented with teleportation and code deformation methods can realize fault-tolerant quantum computing with just a constant-factor

overhead [117, 118]. Continued research along these lines is expected to reduce the overhead required to achieve fault-tolerant quantum computing in real hardware.

Dynamical decoupling and quantum control

To combat non-stochastic errors, methods other than quantum error correction can sometimes be significantly less resource-intensive. For example, methods such as dynamical decoupling (DD) [119] and its generalization, dynamically corrected gates (DCG) [120, 121] , can establish noiseless subsystems immune to certain classes of quasi-static Hamiltonian noise. Being able to efficiently compile the pulse sequences needed to achieve the gains of these methods is an open research area, and further incorporating optimizations of these into programming environments, is a quantum software architecture goal. More broadly, the field of quantum control, which seeks to engineer the overall waveforms used to process quantum states, has an opportunity to provide gains over either quantum error correction or methods like DCG by themselves [122-124] .

2. Programming Quantum Computers

Even with a quantum technology enabled to compute through a mixture of hardware and software protocols, it is a nontrivial task to write effective programs for quantum computers using those primitives. At the very least, one needs a layered software environment that includes high- and low-level languages, compilers, translators, optimizers, and verifiers [125]. It may be beneficial to consider an embedded domain-specific language that may take advantage of already existing analysis and compilation tools. The quantum program should be able to be efficiently compiled, checked, and extended to a given hardware platform. A good high-level language is also flexible enough to be readily translated to new quantum computing models and new methods for fault-tolerant quantum computing. Compilation and optimization tools should allow for reduction of quantum resources in a given quantum program. In addition, they should allow compilation to a variety of targets, where a target is specified based on a hardware architecture. Such specifications must include communication constraints, layout constraints, a set of available operations, etc.

In the following, we give examples of what some of the research challenges are in programming quantum computers, to highlight opportunities for investment.

Quantum compiling

As mentioned earlier, studies of the AQC model have identified the embedding problem as a compiling research challenge. In the circuit model, Ross and Selinger recently solved the optimal ancilla-free quantum compiling problem for Z-rotations [126, 127], but research continues into probabilistic compiling algorithms, such as Las Vegas or “repeat-until-success” compiling algorithms whose expected runtimes

are much faster [128, 129]. When fault-tolerance requirements are taken into account, much of the quantum compiling landscape remains open; for example, relating the sequence of compiled elements to lower-level noise models and performance estimates is an active research area [106, 130]. Developing integrated environments that allow the layout, scheduling, and control of quantum circuit primitives when implemented via a quantum error-correcting code fault-tolerantly remains an open research challenge.

Verification and validation of quantum programs

The first step in being able to reason about quantum programs is creating languages based on well-grounded semantic models. This is a well-studied area in conventional computer science but a relatively nascent idea in quantum computer science. At the same time, the desire to have high-level constructs needs to be balanced with the ability to synthesize efficient quantum circuits, which gets back into quantum compiling R&D. With a well-formed language, it should be possible to assert a variety of purely quantum correctness constraints, such as the no-cloning rule on quantum programs [131]. Ideally, it will be possible to develop automated theorem provers for quantum code in a well-designed language. At the very least, it will be important to include standard computer-science constructs that facilitate debugging, such as types, model checking, and symbolic execution. Very few of these ideas have been rigorously translated to quantum programming—the field is ripe for the picking. In the short term, before large-scale quantum computers are available, it will furthermore be important to develop small-scale quantum computer emulators so that quantum software can be tested and improved for when large-scale quantum computers do become available.

3. Assessing Quantum Computers

As quantum computers become available, it will be critical to ascertain whether, in fact, the performance of quantum computers exceeds that of conventional computers and if so, whether that gain is due to quantum effects that can be expected to scale. Developing clear metrics for assessing quantum computers now is therefore an important exercise to understand how to fairly compare resource usage between quantum and "classical" computers, and how to even compare different types of quantum computers.

4. Computational Support for Quantum Algorithm Development

In this next section, we address some of the computational challenges facing the development of quantum algorithms for applied mathematics. These challenges come in the form of both software and architectural issues that can impact the expected performance of an algorithm implementation. Whereas conventional algorithmic development draws on a robust base of programming and profiling tools to test new ideas, the same computational support for quantum algorithms is currently underdeveloped or does not exist. We explain that quantum algorithms research is a relatively young field and that it currently lacks many of the

computational tools available to conventional efforts. Future development of quantum algorithms will benefit from both exploratory mathematics research and development of computational tools for testing implementations.

The benefits from any new algorithm are due to fundamental speed ups in its complexity as well as to its ease of integration with existing computational frameworks. Quantum algorithms are amenable to demonstrating these benefits when they are designed with considerations for the target architecture. For example, accelerator-based HPC systems currently make liberal use of hierarchical management and process control during application development, and we can anticipate that the utility of quantum algorithms will depend significantly on their ability to accommodate these system constraints. Many quantum algorithms are now developed independently of a machine model. This is due partly to the relative immaturity of quantum processor platforms, which have yet to offer a well-defined model for testing against.

This suggests the possibility that quantum algorithms will be mismatched with the future hybrid HPC systems that might be developed. For example, it is generally appreciated that speedups within a classical parallel computing platform are at best linear in the number of processors, as given by Amdahl's law. But this is not the behavior expected for the scaling of quantum algorithms, which very frequently are not parallelizable in the sense of Amdahl's law. Instead, quantum algorithms have the potential to offer exponential speedup on a per-node basis; that is to say, as the size of each processor increases linearly. This argument suggests that an asymmetric multiprocessor model is better suited for the direct adoption of quantum algorithms as it segments behavior into independent systems. But this approach ignores other application concerns such as minimizing data movement and system complexity. Consequently, the now well-defined problem of HPC co-design becomes an essential issue for assessing the performance of quantum algorithms.

It may be several more years before quantum hardware is sufficiently robust for large- or even modest-scale demonstrations. Nonetheless, it remains possible to use abstract machine models for algorithm testing purposes. The benefits derived from quantum algorithms can then be measured relative to the forecasted impact on a model HPC system. Abstract machine models can offer representations of both the quantum processor and the hybrid HPC system level. These representations can provide meaningful feedback to algorithm developers on which architectural constraints and issues must be addressed. Ultimately, how system architecture constrains algorithm implementations is likely to be a key bottleneck for quantum algorithm performance in future HPC platforms.

Alongside architectural issues, we expect that programming and execution models for hybrid HPC systems will also play a role in shaping quantum algorithms for applied mathematics. Programming models define the means by which end users make use of quantum algorithm implementations. Highly optimized libraries are

frequently used to make implementations of conventional algorithms accessible for a wide range of application developers. This permits details of the algorithm implementation to be abstracted into a well-defined interface. However, the judicious use of these libraries depends to some extent on the acceptable algorithmic use cases and details of the system infrastructure. This applies to both quantum and classical algorithms. There are currently few conventions for developing quantum algorithmic libraries but it is clear that future adoption of these libraries will need to reconcile design features with HPC system concerns. This is especially true for algorithms that pass reference to quantum register elements, i.e., qubits, whose value cannot be copied, but rather must be managed in concert with processing statements. The computational models and programming models discussed in previous sections are likely to play a role in determining the use cases for a quantum algorithm as well as its implementation and overall performance as part of an applied mathematics library. In addition, issues related to non-idealized implementations including finite-bits of precision and overhead arising from error-correction methods need to be better understood. Similarly, execution models that define the order and precedence with which resources are used and the methods by which execution is negotiated must be specified. Even in an abstract setting, these execution models can provide insight into the best choices for algorithm implementation.

Finally, there is an overall challenge in assessing both the relative and absolute measures of performance in quantum computing systems. Time-to-solution and scaling laws are often cited for measuring the benefit of a quantum algorithm relative to other solutions. However, current HPC co-design efforts have emphasized that system-level and end user concerns also play an important role in the future of computer. Metrics such as power consumption, processor programmability, source code accessibility and portability are not yet defined for quantum algorithm usage and may ultimately be deciding factors in adoption of these methods.

5. Relevant Whitepaper Submissions

With respect to computational tools for algorithm development, **WP1** cited the importance of benchmarking methods for emerging quantum computing platforms. **WP6** discussed recent work on finite precision quantum algorithms. **WP7** discussed the significant interplay between quantum and classical computing resources. **WP11** cited recent work on programming environments for the development of software tools for algorithm design and simulation. **WP12** provided an instance of how a quantum computer would need to connect with a classical database. **WP14** discussed the integration of quantum optimization algorithms with other classical methods. **WP16** emphasized the need for computational tools to support algorithmic development, including quantum programming languages. **WP18** discussed integration between quantum and classical computing resources. **WP21** highlighted tradeoffs in quantum circuit synthesis methods. **WP27** presented new methods for large-scale classical simulations of quantum computing systems. **WP28**

emphasized the need for computational tools to support quantum computing environments.

B. Applications for Physical Systems

Many of the potential applications of quantum computers relate to simulating physical systems. As Feynman argued early on in the history of quantum computing, building a computer out of quantum mechanical elements could help circumvent or at least reduce the exponential overhead that comes from simulating quantum mechanical systems on standard (classical) computing hardware [1]. To date, much of work in applying quantum computers to physical systems has been along these lines of quantum simulation. A variety of techniques are used to map physical systems onto qubits, and the types of systems considered include systems from chemistry, materials science, and particle physics.

Algorithms for using quantum computing and quantum annealing for machine learning have also been proposed. Given the importance of machine learning in developing models for many important scientific systems, this application of quantum computing and annealing has generated interest.

1. Quantum Simulation

Quantum simulation is the emulation by a controlled quantum system of another quantum system of interest in the physical sciences. Much recent progress has been made, especially in trapped ion and trapped atom systems, in the analogue simulation of physical systems. Typically, some lattice system of interest in condensed matter physics is mapped to the Hamiltonian of a lattice of trapped ions or atoms, and properties of the condensed matter system are obtained by measurement of the atomic or ionic system. In contrast, digital quantum simulation assumes that the controlled quantum system that is used is a universal digital quantum computer. Digital simulation then uses well-established techniques, principally Trotter formulae, to implement a simulated time evolution under a given Hamiltonian as a sequence of elementary gates. Given such a gate sequence, one can then apply quantum error correction to it, so that once the physical device has reached the error correction threshold, simulations that exceed the decoherence time of the device can be performed.

Digital Quantum Simulation. There are four proposed approaches to digital quantum simulation (DQS) of physical systems. Firstly, one may use a grid to discretize space, represent the position of each particle on this grid by the binary expansion of its components, and evolve forward in time according to the Hamiltonian. Typically in this representation the potential will be diagonal in the position basis, and all off diagonal terms in the position basis will arise from the kinetic term. However, the kinetic term is diagonal in the momentum basis, and the change of basis is accomplished by the Fourier transform, which may be performed exponentially faster on a quantum computer than even the Fast Fourier Transform on a classical computer. The cost of applying the potential is simply given by the cost of classically computing the potential at a point. For Coulomb interactions this is

tractable. This approach is appealing for the simulation of chemical reactive scattering problems, where it is in fact preferable to retain all nuclear coordinates explicitly, thus also allowing for the direct treatment of diabatic effects such as those of interest at conical intersections in chemistry [132].

Secondly, one may perform a simulation based directly on the second quantized Hamiltonian. This is appealing for problems such as molecular electronic structure where a localized orbital basis provides a more economical representation of the wavefunction than a Cartesian grid. Here the Jordan-Wigner or Bravyi-Kitaev transformations are used to map the fermionic second quantized operators to operators on qubits, and evolution under the Hamiltonian is performed using the Trotter method. This approach has the advantage that the couplings in the Hamiltonian are control parameters for the quantum simulation and may be classically precomputed. The digital simulation of molecular electronic structure has been well-studied over the last ten years, with particularly rapid progress being made in the last year [133-137].

Thirdly, one may take the Configuration-Interaction (CI) matrix of a fermionic system as the starting point for the simulation. In this case the challenge is simulation of the time evolution using the Trotter method in the absence of any tensor product structure or natural decomposition into local terms. Here it is the sparsity of the matrix that makes it amenable to quantum simulation. The CI matrix is sparse, and can be decomposed into one-sparse matrices that contain only a single nonzero entry in each row and each column. The time evolution operator is simulated by Trotterizing the CI matrix into a product of evolutions under one-sparse Hamiltonians. Each one-sparse matrix can be efficiently simulated provided that the matrix elements can be quantum computed coherently [138].

Fourthly, a different approach is needed for the simulation of the quantum dynamics of fields. Unlike a collection of finitely-many particles, which can be fully described by a finite list of their spatial coordinates, a field is fully determined only by its values at all points in space, which are continuously infinite. This creates additional challenges for digital simulation of quantum field theories, such as those that describe relativistic phenomena observed in particle accelerators and within nuclei. Nevertheless, by discretizing space onto a lattice, storing approximate field values at each lattice point using qubits, and carefully taking into account discretization errors as determined through renormalization, one can show that quantum computers have the potential to efficiently simulate processes in quantum field theory that require exponential time to simulate using classical supercomputers [139].

For future work there are several productive directions that could impact ASCR goals. In the context of first quantized grid-based methods, new methods for implementation of the potential computation require study to determine their efficacy on problems of chemical interest [140]. This type of study would echo the

recent work [133-137] that has been done in the case of electronic structure calculations.

For the second quantized algorithms the emerging detailed understanding of errors in Trotter decompositions should be extended to realize near-optimal scaling for these methods. There are two approaches to the mapping of fermions to qubits that have been used in quantum simulation. These are the Jordan-Wigner and Bravyi-Kitaev transformations. Circuits based on the Jordan-Wigner transformation, which naively imposes a large overhead in terms of gates, have been optimized to remove this overhead. The Bravyi-Kitaev transformation, which removes the overhead at the level of the Hamiltonian, has not yet been so optimized and currently it is unclear which of these transforms is best for quantum simulation algorithms. Theoretical work to determine the truly optimal second quantized digital simulation algorithms is high impact as it brings the early implementation of these methods much closer.

In terms of the sparse algorithms there is also the possibility for rapid and high impact progress. The major opportunity in this area is the recent development of optimal techniques for the simulation of sparse Hamiltonians, with nearly optimal dependence on all parameters. For a given degree of error in simulating the time evolution operator, these methods scale as the log of the reciprocal of the error, as compared with a power of the reciprocal of the error for conventional Trotter methods. These new methods use the same type of sparse decomposition investigated for chemistry [141], and they face the same challenge. The matrix elements of the CI matrix must be computed within the quantum algorithm so that they may be accessed coherently in superposition. These elements are determined by molecular integrals over the basis functions, and this represents a large and potentially problematic obstacle to the practical use of these methods.

The careful characterization, optimization and simulation of these algorithms are required to determine whether they can in fact beat second quantized approaches for problems of interest to DOE. This area is very open at present as both the methods and their application to chemical problems are very much at the research frontier. The exponential improvement in scaling with precision, if not overwhelmed by the overhead of the method, makes these methods worthy of serious study.

It is interesting that classical chemistry faced a similar problem in the period prior to 1950. The so-called “nightmare of the integrals” was the problem of determining the molecular integrals in hydrogenic orbitals. Only with the introduction of Gaussian basis sets, which persist to the present day, was it possible to calculate the CI matrix elements efficiently. Today the major challenge in the use of the asymptotically optimal quantum algorithms for fermionic quantum simulation is also the “nightmare of the integrals” – the necessity to perform these calculations coherently as a subroutine of a quantum simulation. This naturally raises the

question of whether the continued use of 65 year old classical basis sets for the representation of the wavefunction is appropriate in a quantum context.

One very promising approach that might be realizable in the nearer term than full-scale digital quantum simulation is using shallow quantum circuits to evaluate the energy of a wave function that can be prepared as a function of a variety of parameters that may then be varied and optimized on a classical computer. This variational quantum eigensolver approach promises significant improvements for, notably, unitary coupled cluster wave functions, which could serve as a "gold standard" for high-accuracy chemical electronic structure calculations [142].

Analog quantum simulation. Analog quantum simulation (AQS) refers to arranging physical qubits so that their Hamiltonian (or master equation) closely approximates that of an interesting quantum system that we wish to simulate, then studying their evolution or equilibrium behavior. AQS has real promise for computing properties of lattice systems, materials, and perhaps even molecules. Potentially, it avoids the need for active error correction precisely because interesting properties of useful physical systems *must* themselves be robust to environmental perturbation, and if the analog device is sufficiently faithful to the system being simulated, then it inherits that robustness. AQS is a rapidly developing field full of interesting scientific challenges and opportunities [143]. There is a critical need for both (1) theoretical analysis to model and understand the source and reliability of analogue simulation's robustness, and (2) experimental implementation of controllable, configurable hardware that will confirm or deny the usefulness of this methodology.

AQS has been applied to study hard problems in a number of different fields of physics. Greiner et al. [144] used a two-dimensional optical lattice of atoms to simulate different parameter regimes of the Bose-Hubbard Hamiltonian, controlling the lattice potential depth so as to probe different values of the ratio of the on-site to tunneling energy parameters. The observation of a phase transition in the optical lattice, where individual sites began to be doubly occupied, corresponds to a phase transition in the Bose-Hubbard Hamiltonian.

In a similar experiment, Friedenauer et al. [145] used two trapped calcium ions to simulate the quantum Ising model. In this instance, varying external laser fields provided an analog interaction to the spin-spin interaction term in the Ising Hamiltonian, and paramagnetic to ferromagnetic to antiferromagnetic phase transitions could be observed.

Algorithms using AQS to solve lattice quantum chromodynamics (QCD) problems have also been proposed. The physics of neutron stars, including their formation, composition, and ultimate fate is an urgent subject of modern science. Despite several decades of intense study, the inaccessibility of cold dense nuclear matter to experimental probes, coupled with the restriction of lattice QCD simulations to zero density by the "fermion sign problem," has inhibited any significant advancement in

our understanding of neutron star matter. In particular, the importance of 3-, 4-, and higher-order many-body forces to the nuclear equation of state, which determines neutron star structure, are only poorly understood. As a result of these limitations, all calculations of the neutron star equation of state have relied either on nucleonic potential models inapplicable at high density, or on symmetry-based models designed to capture the essential aspects of dense QCD. But given the richness of the QCD phase diagram, it is extremely difficult to unambiguously identify these essential aspects, and a new computational architecture capable of probing the strongly-coupled, dense regime of QCD is required.

Quantum simulation of QCD provides a natural resolution of the fermion sign problem, as a quantum simulator is inherently a phase-manipulating system. That is, by employing an inherently quantum system to simulate another quantum system, one computes in precisely the same way that nature itself "computes". Not only would quantum simulation of QCD eliminate the technical challenge posed by the sign problem, it would also vastly expand the universe of soluble problems to include the real-time evolution of strongly-correlated systems, direct measurement of transport properties, and many others. While the full implementation of a QCD simulator will require a number of theoretical and technical developments, the ability to simulate certain classes of both Abelian and non-Abelian gauge fields in the laboratory, using highly tunable ultracold atomic gases, has already been realized. These systems provide an excellent testing ground for the simulation of quantum gauge theories and the further development of more complex simulation techniques. Moreover, the work required to take steps toward a fully quantum mechanical simulation of QCD promises to revolutionize the study of complex many-body phenomena including superfluidity, the spin-Hall effect, and strongly-coupled physics, enabling an exhaustive exploration of these phenomena in a manner impossible with conventional experimental probes.

One notable advantage to AQS is that it is anticipated that these simulators will be robust to the presence of errors up to a certain threshold. It is anticipated that interesting properties of useful physical systems must themselves be robust to environmental perturbation, and, if the analog device is sufficiently faithful to the system being simulated, then it inherits that robustness. In particular, the study of quantum phase transitions, due to their being collective phenomena, should be stable to minor variations in the control parameters. Such robustness could reduce or eliminate the need for active quantum error correction, dramatically reducing the physical requirements needed for a useful implementation of quantum simulation. The existence and the extent of this robustness is an open and important problem in the field.

2. Machine Learning

Data science, and, in particular, machine learning (ML), have emerged as central components of scientific computation. Recent results such as the detection of the Higgs boson would not have been possible without the development of

sophisticated machine learning techniques to detect faint signals within the data. Although such techniques have led to a revolution in data processing, they are often computationally intensive and as a result many of the most sophisticated methods have not yet been brought to bear against scientific problems. Quantum algorithms promise to rectify this by not only dramatically speeding up existing machine learning approaches but also promising new methods that are not known to be otherwise classically tractable.

Apart from accelerating data processing, quantum machine learning algorithms can be used in tandem with quantum simulators. This would allow a quantum computer to scan through an exponentially large family of candidate materials to find a family of candidates that are promising for use in, for example, solar cells, dramatically accelerating the development time for such materials. As such, quantum machine learning is a natural partner with quantum simulation within any program that wishes to investigate automated discovery of materials, drugs or quantum device certification.

Machine learning is a multi-disciplinary field of study that combines techniques from artificial intelligence, statistics, mathematics and condensed matter physics to train computers to solve problems involving pattern recognition in complex sets of data. Machine learning is a broad field and, in fact, many of the techniques that scientists and engineers use on a daily basis, such as least squares regression, can be thought of as a form of machine learning.

The successes of machine learning in understanding big data sets have sparked a revolution within industry. Entire businesses, such as Google search and Microsoft's Bing are powered by these insights. The major leaps that have been made in natural language processing and image recognition also owe no small debt to the revolution that is ongoing within machine learning. Similarly, modern genomics is only possible because of machine learning methods. Despite their relative ubiquity in industry, machine learning methods rarely appear within the physical sciences outside of high-energy physics. That being said, the success of machine learning at those problems strongly suggests that machine learning will also play a transformative role more broadly within the next few years.

Classification is a critical application of machine learning. The goal of such algorithms is to train a computer to automatically classify data from an experiment or other data source. In such applications, the user chooses a rich family of models that can generate a rich set of data sets. In the simplest approach to training, each data example is paired with a label that is supplied by a trusted expert and the goal is to find the model within the class of allowable models that is most likely to assign the correct label to each of the data examples in the set. This form of learning is called *supervised machine learning* and is perhaps the simplest form of machine learning. *Unsupervised machine learning*, on the other hand, solves a similar problem except that the computer assigns its own labels to tag the data without direct human intervention.

As an example, classification methods have been applied to the joint European torus (JET) fusion experiment to predict when the plasma in the device becomes unstable. Such instabilities not only cause the experiment to fail but also dissipate enough power to cause the torus to literally leap off the ground resulting in substantial downtimes. The use of a supervised machine learning algorithm (the support vector machine algorithm) has been used to successfully predict when the plasma is about to become unstable with accuracy greater than 80%. The use of machine learning promises to predict when such instabilities occur and perhaps lead to methods that can re-stabilize the plasma.

Unsupervised classification algorithms are also extremely important in data processing. Unsupervised machine learning searches for patterns in data sets that the user may not even know exist. Perhaps the simplest such algorithm is clustering. Clustering algorithms take a set of data for which no a priori pattern is known to exist in the data and assigns the data to some number of distinct clusters that best represents the data. Such algorithms are useful because not only they provide insight into the data but they also allow computers to identify such patterns without human assistance. This application may be of particular use in material and drug discovery algorithms and may also help scientists find patterns in complex data sets that cannot be readily visualized.

Machine learning can also be used to predict the behavior of a system. For example, the exact same methods can be used to find energy usage patterns and predict how much power a city will use as a function of time based on usage patterns provided to the system beforehand. Similarly, the exact same approach could be used to learn an empirical model for a system of interest such as a high-temperature superconductor or a detector used in a cyclotron. These empirical models can then be used as part of a larger program to accurately predict the behavior of the device or material in a much larger system.

Training a classifier can be very time and data intensive. It can require millions of training examples to obtain a model that understands the pattern behind the examples without overfitting the data. Consequently, some of the more sophisticated methods (such as deep convolutional neural networks) may take days or weeks of computer time to merely train the classifier. The search for improved methods for training machine learning models is consequently a very important goal in machine learning and by extension throughout the physical sciences.

The development of machine learning algorithms for application in the physical sciences is of interest to ASCR supported communities as illustrated by the workshops that ASCR has sponsored on the topic in recent years. This significance stems from the fact that classical machine learning will be a major application for exascale computing when the technology arrives within the next several years. Similarly, the ASCR community is well poised to take a leadership role in the discovery and development of quantum approaches to machine learning. We lay out

the cases below for the various forms of quantum machine learning that arose in our discussion and discuss their promise in relation to the ASCR mission goals.

Machine learning via quantum computing: Although machine learning was proposed as a possible application of quantum computation nearly a decade ago, it is only relatively recently that concrete proposals have been made for using quantum computers for tasks within machine learning. Quantum algorithms for data fitting, clustering, nearest-neighbor classification and deep learning have recently been proposed. All of these methods offer polynomial speedups over their classical counterparts and, in some restricted cases, may even exponentially accelerate machine learning.

The discussions within the quantum simulation focus group identified several promising new avenues of inquiry that involve the use of quantum computing or quantum simulation.

- New quantum algorithms for machine learning.
- Applications of quantum machine learning to materials and drug design and to other problems within the DOE mission (examples are given below).
- Methods for certifying/testing analog quantum simulators.

The need for the development of new quantum algorithms for machine learning was a common theme in many of the discussions in the session. Although quantum algorithms are known to accelerate a diverse array of machine learning algorithms, a common challenge that many of them face is that a large quantum memory will be needed in most practical applications. In particular, if a machine learning algorithm is being trained with a database of a million training examples then a quantum database containing millions to billions of qubits may be needed. Since quantum computers of this scale are not anticipated in the foreseeable future, new algorithms that do not require a large quantum memory may be needed for the training process.

An example of such a machine learning algorithm was presented in the quantum simulation session. This result showed an algorithm for quantum deep learning that not only does not require extensive quantum memory but also provides a quantum speed advantage over classical methods and promises better training while revealing new classes of efficiently trainable models that are not known to be tractable classically. Since deep Boltzmann machines represent a state of the art method for machine learning, this represents an important step forward for quantum machine learning. Yet there is still much work that needs to be done. Further quantification of the training advantages of the algorithm relative to existing classical approaches is needed (a presently difficult task that will become easy once quantum hardware to run the algorithm on exists). Also all algorithms proposed so far strongly mimic existing classical training algorithms. The search for a manifestly quantum model for machine learning remains an important unrealized goal within quantum machine learning.

What impact would a quantum machine learning algorithm have on science? The most evocative answer that emerged from the panel discussion is that a quantum machine learning algorithm could be used to identify promising materials or drugs in concert with a quantum simulator. To see how this might work, consider the following. Imagine that you want to find a potential candidate for a cancer therapy. The user would begin by compiling a list of known compounds that are effective or ineffective for fighting a particular form of cancer. The user then decides a class of molecular features that they believe will be useful for deciding the effectiveness of a drug. Quantum simulation algorithms could then be used to calculate these features for use in data for a *supervised* quantum machine learning algorithm. A quantum computer could subsequently use Grover's search to rapidly scan over a database of potential candidate molecules in search of one that the trained model believes will have therapeutic properties. This approach is by no means unique to drug design. It can also be used in materials design tasks such as engineering photovoltaics or high-temperature superconductors, tailoring optimal catalytic agents by mapping the potential energy surface for adsorption onto a substrate by quantum principles, or even to design new experiments in high energy physics.

Such applications would otherwise require an exascale or larger classical cluster, whereas a quantum computer with a few hundred *logical quantum bits* may be able to achieve the same task with exponentially less power consumption and computation time. ASCR has the opportunity to foster the development of an end to end algorithm that combines ideas from quantum simulation and data science to move beyond simple simulation and lead to automated methods for gaining insight into the structure of physical systems. This has the potential to revolutionize the way we approach solving challenging design problems using quantum hardware.

Machine learning via quantum annealing. Quantum annealing is a recent technology that opens up the possibility of obtaining quantum speedups for important optimization and machine learning problems using existing hardware. Quantum annealers, such as the D-Wave 2, use a variant of adiabatic quantum computing where an initial state is slowly evolved into a target state that encodes the answer to the optimization or machine learning problem. The quantum adiabatic theorem promises that slow evolutions do not excite the system out of its (instantaneous) minimum energy configuration, which means that if both the initial and final states are chosen to be minimum energy (instantaneous) states of the slowly varying quantum system then the resultant quantum state will yield the answer to the problem at hand. This approach is known as quantum annealing because it is strongly analogous to annealing in metallurgy. Quantum annealing differs subtly from adiabatic quantum computing because the system is not required to be in an eigenstate throughout the computation in quantum annealing.

Unlike most circuit-based quantum computers, quantum annealing is robust to many common forms of noise because of the large energy barrier between the

ground states of the system and the manifold of excited states. In particular, a large gap creates a significant impediment for the environment exciting the system out of its ground state. In fact, such transitions will become effectively impossible if the environment cannot provide enough energy to excite the system out of the ground state. Therefore the quantum annealer will be expected to be robust against thermal excitement if the energy gap is much greater than the temperature of the environment. Controlling a quantum annealer is also comparatively simple since the slow evolutions require little bandwidth. These properties mean that devices consisting of hundreds of qubits are commercially available; however, existing devices are not universal and their quantum nature remains hotly debated.

Nearest-neighbor classification provides an excellent example of how quantum annealing can be used to classify data. The algorithm seeks to classify a piece of data by assigning it to the same class that the most similar piece of training data was assigned. A quantum annealer would solve this machine learning task by comparing the test data to each of the examples in the training set and applying an energy penalty for each bit where one of the data examples differs from the training example. Therefore the closest example (using the Hamming distance) is the one that has the least energy. Since quantum annealing aims to map low energy states to low energy states, the outcome of the protocol is likely to be a data point that is close to the one being classified. This solves the problem of nearest neighbor classification. Similar strategies have also been considered for classical machine learning (specifically training restricted Boltzmann machines).

No conclusive evidence has yet been found of quantum annealers providing an advantage (a speedup, reduction in the computational resources, or a reduction in the asymptotic scaling of the algorithm) over existing classical computers. Verification of this conjecture remains a major open problem in quantum information processing.

Members of our panel showed considerable interest in this form of machine learning. White papers were submitted on a number of different issues such as using this approach to process data from the LHC (Large Hadron Collider) and the use of quantum annealing to predict plasma instabilities in future fusion experiments. Other results were presented that show that quantum annealing may have advantages for rapidly mining large databases of materials or to model protein-DNA binding using experimental data, with enormous impact on understanding regulatory functions of non-coding genome sites [146]. In the latter case, quantum annealing outperformed other machine learning methods for some (but not all) instances of the problem. Similar advantages (e.g. finding improved optima or finding optima a larger percentage of the time) may also be seen for problems in material science and catalyst design, by using training-testing approaches to model potential energy surfaces of reactants on surfaces [147, 148].

Another potential application of quantum annealing within the umbrella of machine learning is software validation and verification. Software validation and verification

is a computationally important task for aircraft testing. Supercomputers are often required in such applications to exhaust enough of the input space for avionics systems to ensure that catastrophic bugs in the software are sufficiently rare. Software verification and validation is also of great interest outside the aerospace industry: it is highly relevant for other mission critical systems such as control software for nuclear reactors and other energy infrastructure. By assigning a lower energy to inputs that produce the correct output without crashing, quantum annealing can be used to search for bugs. The ability of quantum systems to tunnel through energy barriers into low energy sectors raises the hope that quantum annealers may find use for software validation if they, indeed, exploit quantum tunneling.

Perhaps the biggest opportunity that quantum annealing provides is the ability to use existing hardware as a "white box" to probe the advantages that quantum mechanics can provide long before scalable quantum computers become available. Serious scientific investigation into the potential that such quantum architectures may have for rapidly solving problems that would otherwise be beyond the capabilities of an exascale (or larger) supercomputer is needed. It should also be noted that, quantum annealing promises to disrupt current approaches to data processing and optimization within a short time frame and it therefore presents the possibility of yielding new quantum algorithms for optimization that are radically different than those that would arise from the traditional circuit model of quantum computing.

3. Boson Sampling

Boson sampling is a recent technology proposed by Scott Aaronson and Alex Arkhipov [149] that aimed at probing the strong Church-Turing thesis. The Church-Turing thesis is the hypothesis that all physically realistic processes in the universe can be efficiently simulated by a Turing machine. Colloquially, this is equivalent to saying that the universe is not intrinsically more powerful than a classical computer. Quantum theory has revealed that the strong Church-Turing thesis is likely false. Instead it is commonly believed that quantum computers are needed to efficiently simulate many physical phenomena. Not all quantum systems are difficult to simulate classically and it is unclear, in general, whether any particular class of physical systems will permit an efficient classical simulation.

Boson sampling was devised to address this issue. It works by preparing a set of single photon states and feeding them into a network of polarizers and beam-splitters. Aaronson and Arkhipov showed that if a classical computer could efficiently sample from the resulting distribution of photons yielded by the linear--optical network then the polynomial hierarchy would partially collapse. Such a collapse of the polynomial hierarchy has absurd consequences for computer science and as a result such collapses are nearly universally believed to be impossible. The simplicity of Boson sampling provides a platform that may be even more likely than quantum annealing to provide evidence for a quantum advantage over classical computing.

Boson sampling does not, unfortunately, answer a decision problem. Instead it illustrates that nature can sample from distributions that it would not be able to if the strong Church-Turing thesis were true (modulo complexity theoretic assumptions). Recent work discussed in our session suggests that Boson samplers may actually be able to solve problems in chemistry [150]. The vibrational structure of a molecule can be probed using a Boson sampler by tuning the beam splitter parameters precisely and replacing the input single photon states with squeezed vacuum states. This problem is not known to be tractable using classical computing. This form of Boson sampling is a form of analog quantum simulation that can investigate phonon dynamics in chemicals or materials using a much simpler system than would be required by quantum annealing or a full quantum computer.

4. Relevant Whitepapers

WP11 summarized recent progress in digital quantum simulation. **WP17** discussed opportunities around analog simulation and other simulations that could be run on near-term hardware. **WP8** described opportunities for analog simulation around neutron star material and lattice QCD. **WP8** discussed computation using configurable quantum networks. **WP23** surveyed deep learning progress and discussed opportunities for quantum machine learning. **WP10** discussed applications of machine learning via quantum annealing for Higgs particle search. **WP24** discussed applications of machine learning for predictive fusion applications.

C. Applications for Applied Mathematics

Some quantum algorithms correspond directly to certain scientific applications, such as the quantum chemistry and high-energy physics algorithms considered in the previous section. In that setting, a quantum algorithm is designed specifically towards an application goal, e.g., recovering ground state energies or scattering amplitudes. But other quantum algorithms can be used more generally, for example, as subroutines that support a broad range of applications or numerical solvers. In this applied mathematics setting, quantum and classical algorithms work together, perhaps in parallel, and may exhibit non-trivial dependencies on each other. Their use is not application specific but rather driven by the varying demands of HPC end users. Quantum algorithms for applied mathematics therefore represent a very broad and revolutionary approach to algorithmic development for high-performance computing.

In this section, we review state of the art and open challenges of quantum algorithms for applied mathematics. We examine use cases of quantum algorithms for linear algebra, integration and summation, optimization, and graph theory problems as prototypical examples of applied mathematics. A comparison of the algorithmic complexities to state of the art classical results emphasizes the potential for quantum algorithms to make substantive differences across many scientific domains.

Several quantum algorithms have been discovered that offer substantial speedup over classical algorithms for problems in applied mathematics that may be of relevance to DOE's mission. In particular, quantum algorithms achieving exponential speedup over known classical algorithms have been discovered for certain problems in linear algebra [13] and combinatorial optimization [51]. In addition, quantum algorithms achieving polynomial speedup have been discovered for integration [151] and summation [152], extraction of certain graph-theoretic properties, and optimization on structured objective functions [153, 154]. Adiabatic quantum computation and quantum annealing also show promise for optimization problems [15]. Their performance relative to classical computation is currently a highly active area of research [25-28, 31, 155]. This section summarizes known results and areas of ongoing research in quantum algorithms for applied mathematics.

1. Linear Algebra

Solving systems of linear equations is one of the most basic and ubiquitous problems in scientific computing. A canonical version of this problem is, given an invertible $N \times N$ matrix A, and a vector b, solve $Ax=b$. The standard classical algorithm for this problem runs in time $\mathcal{O}(N^3)$ for worst-case (i.e. non-sparse) instances. The asymptotically fastest classical algorithm for this problem runs in time $\mathcal{O}(N^{2.373})$ time, but is not practical [156]. For sparse A, classical algorithms still require time scaling at least linearly with N. In a 2009 breakthrough, Harrow, Hassidim, and Lloyd (HHL) discovered [13] a quantum algorithm for certain instances of this problem that runs in $\mathcal{O}(\log N)$ time, thereby obtaining an exponential speedup.

For an arbitrary $N \times N$ matrix A, it is impossible to solve $Ax = b$ using classical or quantum approaches in time less than $\mathcal{O}(N^2)$because it takes N^2 time just to read all the entries of A. However, for some applications, the entries of A may be determined by mathematical formulae either purely theoretically or from a smaller underlying set of empirical parameters. In this case, it is possible for quantum computers to query the matrix elements in superposition and achieve exponential speedup. For the HHL algorithm to achieve exponential speedup, the matrix A must be well-conditioned and sparse, and the vector b must correspond to an efficiently constructible quantum state. Furthermore, the answer x cannot be read out in its entirety (which would necessarily take time N). Rather, one can extract approximations to quantities of interest such as $x^T M x$ for sparse matrices M. At the core of the HHL algorithm is a technique for simulating Hamiltonian time evolution. Thus, advances in quantum simulation algorithms directly lead to improved versions of the HHL algorithm that either run faster or can handle broader classes of matrices.

Finding problems of practical interest that can be reduced to linear algebra problems fitting the conditions of the HHL algorithm is a current area of research that has attracted substantial attention from the quantum algorithms community. In particular, quantum algorithms based on the HHL algorithm have been proposed for

estimating classical electromagnetic scattering cross sections [157], solving linear differential equations [158], estimating electrical resistance of networks [159], least-squares curve-fitting [160], and machine learning [161-164]. Due to the highly varied and strongly problem-instance-dependent performance of machine learning algorithms, the advantages of quantum machine learning algorithms relative to classical heuristics are a continuing subject of research and debate [165]. In [166] it was shown that quantum computers can invert well-conditioned matrices using only logarithmically many qubits, whereas polynomially many classical bits are required, thus providing a substantial reduction in computational resources for large systems. Therefore, solving linear systems of equations may be a promising application for small scale near-term quantum computers.

2. Integration and Summation

Given a list of N numbers, the classical complexity of exactly computing their sum is of order N. One can also approximate the sum by random sampling. By standard statistics, approximating the sum to within ±ε requires $\mathcal{O}(1/\varepsilon^2)$ samples. In 1999, Nayak and Wu showed that by querying the numbers in quantum superposition rather than sampling randomly, one can approximate their sum to within ±ε in time $\mathcal{O}(1/\varepsilon)$, thus achieving a quadratic speedup over classical randomized algorithms [152]. Nayak and Wu also obtained quantum speedups for related tasks such as approximating the median of data sets. Their quantum algorithms for these tasks are generalizations of Grover's search algorithm. Building upon this work, a quantum algorithm achieving quadratic speedup for the approximation of continuous integrals (with sufficiently well-behaved integrands) was given in 2001 [151].

Quantum algorithms for integration and summation are no longer a highly active area of research in quantum algorithms. However, they are certainly worth mentioning due to their widespread applications. In addition, these algorithms serve as examples of a more general phenomenon: polynomial quantum speedups over classical randomized algorithms obtained by adapting Grover search [167] and its generalizations such as amplitude amplification [168] and quantum walks [169]. Any classical randomized algorithm that succeeds with probability p must be run $\mathcal{O}(1/p)$ times to achieve success with high probability, and the runtime of the algorithm has a corresponding factor of $1/p$. Using amplitude amplification, a corresponding quantum algorithm can be obtained where this factor is reduced to $\sqrt{1/p}$. A currently active area of research in quantum algorithms is to take state of the art classical algorithms and investigate what degree of speedup is achieved by applying amplitude estimation or quantum walks to any subroutines to which they can be applied. Speedups have been obtained this way for simulated annealing [170], verification of matrix products [171], and the subset sum problem [172]. It is likely that further quantum speedups for applied mathematics problems could be discovered in this vein.

3. Optimization

The general optimization problem is, given some objective function *f*, find the *x* within the domain of *f* that minimizes (or maximizes) *f*. Applications of optimization algorithms are extremely varied, ranging from airline scheduling, to circuit layout, to machine learning. The difficulty of optimization problems and the corresponding suitable strategies also vary widely depending on the structure of the objective function. Many of the widely used classical algorithms are heuristic methods that lack rigorous performance guarantees and may only find approximate solutions. Quantum algorithms for optimization similarly vary widely. For certain highly symmetric objective functions, quantum algorithms achieving provable polynomial speedups have been found [153, 154]. For problems amenable to solution by classical simulated annealing, a quadratic quantum speedup can be achieved [170]. For more difficult problems, including NP-complete combinatorial optimization problems, a number of quantum heuristic methods have been proposed [15, 51], whose performance is currently a subject of intense theoretical and experimental study.

For some optimization problems, the costliest part of the algorithm is the evaluation of the objective function. In such cases, the number of queries to the objective function is a good metric of computational cost. The query complexity is also a convenient metric for analysis— in simple cases upper and lower bounds on query complexity for classical and quantum algorithms can be proven. In particular, consider the final stages of a numerical optimization algorithm for a smooth objective function. Sufficiently close to the minimum, the objective function should be well approximated by a Taylor series about the minimum. In this region, it will be given approximately by a positive-definite quadratic form. The problem of minimizing a quadratic form is thus a simple but important case of numerical optimization. The classical query complexity of minimizing a quadratic form in d variables has been proven to be of order d^2 [173]. By evaluating the objective function in quantum superposition and taking advantage of the quantum algorithm for gradient estimation [174] one can find the minimum of a quadratic form using $\mathcal{O}(d)$ quantum queries [154]. More generally, any classical algorithm using numerical gradient estimation for an objective function on d variables can be similarly sped up by a factor of d using quantum gradient estimation. Finding a lower bound for the number of quantum queries needed to minimize a quadratic form is an open problem.

A second example of a simple optimization problem admitting polynomial speedup was given in [153]. The objective function takes strings of n bits as input and outputs the number of bits that differ from the optimal bit string. This objective function is in some sense a discrete analog of a conical basin; its value is proportional to the Hamming distance (rather than Euclidean distance) from the optimal solution. As shown in [153], the fastest possible classical algorithm for this problem requires n queries to find the minimum, whereas a single query suffices for the quantum algorithm. This quantum algorithm and the quantum algorithm for minimizing quadratic forms are similar in that they operate on objective functions

without local minima and achieve a speedup by a factor of the dimension of the domain of the objective function. Whether these two results can be subsumed as special cases of a more general quantum speedup for optimization problems with simple objective functions is an area of current research.

For objective functions that are relatively smooth but populated with local minima that would trap simple gradient descent, a popular classical heuristic algorithm is simulated annealing. Simulated annealing is a randomized algorithm in which "uphill" moves are accepted with some probability thus allowing escape from local minima. As shown in [170], classical simulated annealing can always be quadratically sped up by adapting the quantum algorithmic techniques from [169]. This quantum speedup is achieved within the standard quantum circuit model and should not be confused with quantum annealing, which is discussed below.

In 2000, Farhi et al. proposed a new form of quantum computation called adiabatic quantum computation [15]. Adiabatic quantum computation can be simulated by quantum circuits and vice-versa [175]. However, most attention has focused on the possibility of using adiabatic quantum computation to solve optimization problems, for which it appears naturally suited. The central question about quantum adiabatic algorithms is which optimization problems, if any, admit speedup by adiabatic quantum computation over classical computation. Through the adiabatic theorem, the runtime of an adiabatic quantum algorithm can be bounded using the eigenvalues of the algorithm's Hamiltonian. Numerical evaluation of these eigenvalues is very difficult due to the high dimension of the Hamiltonian matrices, although some progress has been made numerically (cf. [176, 177]). Analytically, we know that adiabatic optimization can find minima in polynomial time in cases where simulated annealing and other classical local search algorithms fail to do so [178, 179]. Conversely, on some examples, adiabatic quantum optimization can fail in cases where gradient descent succeeds [180]. Characterizing the instances in which adiabatic optimization can be expected to yield advantage is a difficult mathematical research problem currently being pursued.

One of the attractive features of adiabatic quantum optimization is that it may be easier to implement than a general purpose quantum computer based on the quantum circuit architecture. Prototype quantum optimization hardware with hundreds of qubits is now commercially available from D-Wave systems (The extent of the quantum nature of the D-Wave hardware is still a matter of intense debate). The computation performed by D-Wave hardware differs from Farhi's original proposal for adiabatic quantum computation in several respects. First, the system inevitably has coupling to thermal and other environmental noise, which may wash out some quantum effects. Secondly, the rate at which the algorithms are run is limited by the hardware rather than determined by the eigenvalues of the Hamiltonian. The form of computation performed by the D-Wave machine is often referred to as quantum annealing. The goal of quantum annealing is usually to find approximate solutions to optimization problems rather than obtaining true optima by remaining in the ground state, as is done in adiabatic algorithms. Whether

quantum annealing has the potential to solve optimization problems faster than classical computing is currently a matter of intense debate and research [25-28, 31, 155].

Recently, a new form of quantum algorithm for optimization was proposed, called the Quantum Approximate Optimization Algorithm (QAOA) [51]. It shares with adiabatic algorithms the feature that they are both easy to define but its performance is nontrivial to analyze. Nevertheless, an example of a combinatorial optimization problem has now been found for which QAOA provably finds, in polynomial time, a better approximation than is found by any published polynomial-time classical algorithm [51]. Time will tell whether this exponential speedup generalizes to a broader class of problems and whether it holds its lead against further progress in classical approximation algorithms.

4. Graph Theory

Quantum algorithms offering polynomial speedup have been discovered for a variety of graph theoretic problems, including finding spanning trees, locating cliques, and deciding bipartiteness. Originally, these algorithms were discovered individually without the aid of any general and powerful underlying theory. Recently, however, a more systematic understanding has started to emerge through the use of span programs and generalized adversary bounds. Thanks to these new tools, rapid progress is being made in finding quantum algorithms offering polynomial speedup for a variety of problems that can be formulated in terms of query complexity. This includes many graph-theoretic problems of potential interest within applied and computational mathematics. We may anticipate further progress along these lines in the near term.

5. Relevant Whitepaper Submissions

With respect to quantum algorithm development for applied mathematics, **WP4**, and **WP10** addressed applications of machine learning. **WP6** discussed the development of finite precision algorithms for linear algebra. **WP9** presented efforts to find quantum algorithmic advantages beyond speedups. **WP14** discussed quantum optimization as a heuristic for combinatorial optimization solvers. **WP15** presented algorithms for optimization and quantum simulation. **WP19** discussed algorithms for linear systems and Hamiltonian simulation as well as ground state problems. **WP22** presented methods for topological and geometric data analysis using quantum machine learning. **WP23** discussed machine learning extensions based on quantum deep learning networks.

IV. Research Opportunities

This report has identified a number of key research problems in quantum computing that are relevant to its future applicability for DOE and ASCR missions. These problems have aligned broadly with the topics of quantum computational and programming models, quantum algorithms for applied mathematics, and scientific applications of quantum computing. We provide a summary outline of the near-term research opportunities in these areas.

Quantum computational devices capable of high-fidelity operations on multiple qubits are just starting to appear. There is an outstanding need for protocols that test, validate, and benchmark these emerging "micro-QC" devices with respect to their theoretical performance expectations. Clear research opportunities include the development of validation methods for micro-QC devices based on simple calculations and algorithm testing. This also hints at opportunities to explore the utility of few-qubit devices for solving computational problems. The results of those efforts may be useful for guiding the future hardware development by setting performance expectations. These questions can be further focused to ask: What quantum algorithms can be implemented on a micro-QC that is too small or too noisy to incorporate useful error correction? To what degree can those algorithms demonstrate the quantum nature or advantage of the device? Which algorithms and applications are best suited to demonstrate "quantum supremacy" (i.e., a computational improvement over the best performance achievable with purely classical resources)?

A related but distinct research opportunity is to address the question of "How do we compare and rank different quantum computational devices?" Validating and benchmarking methods are needed for a variety of development scenarios. Meaningful measures of comparison are to evaluate quantum computational devices that (a) operate using different base technologies, (b) operate with different noise models, (c) operate with widely disparate numbers of qubits, and (d) operate with different computational models. Similarly, methods are needed to validate that a device or processor distinctly behaves quantum mechanically. A striking example of this research challenge is illustrated by early attempts to validate and benchmark first generation quantum annealing machines for performance and capability relative to classical hardware, a question which still remains only partially answered. We anticipate that advancing the size and features of quantum computing technologies will make such comparisons only more difficult. Ultimately, these efforts will likely need to lead to specifications for quantum technology certification that might be used in future HPC acceptance criteria.

Quantum software development is another broad and critical area of research need. What tools can or will play the roles of programming languages, libraries, compilers, and debuggers for quantum software? Are there useful abstract machine models that can simplify the process of addressing different hardware platforms? Programming tools and software will be needed for expressing quantum

algorithms and building quantum-enabled applications. These tools include programming languages capable of expressing quantum constructs, compilers for translating these languages into machine instructions, and system execution models for carrying out those instructions. In addition, program debuggers and simulators will be necessary for validating code and verifying implementations.

Sustained efforts in both the foundational mathematics of quantum algorithms and the computational infrastructure to support their development are needed. Each of the quantum algorithms discussed in this document presents specific research opportunities, many of which are discussed in their relevant sections. However, quantum simulation deserves specific mention here. In digital quantum simulation, there is urgent need for research that reduces resource requirements (notably computation time), and that generalizes simulation algorithms beyond molecular systems. Analog quantum simulation also appears promising (for near-term hardware), but there is critical need for better understanding of analog simulations' robustness to noise and decoherence. Another promising research opportunity is the development of methods to apply results from one analog simulation to more general systems, along the lines of what is currently done with Ising or Hubbard models.

Quantum algorithms that offer speedup to the basic primitives of applied mathematics, such as linear algebra, integration and summation, discrete and continuous optimization, and graph theory have already been developed. Further fundamental research is needed for discovering new quantum algorithmic techniques, and new classes of mathematical problems admitting quantum speedup. Further development is needed to improve the efficiency of known classes of quantum algorithms and adapt them to realistic hardware and to problem instances of practical interest. Speedups for these basic operations of applied mathematics can have broad impact due to their widespread application throughout scientific computing.

In addition to mathematical research, the field of quantum algorithms requires computational infrastructure to support robust testing and validation of new ideas. In the absence of reliable quantum processor platforms, it seems likely that some form of quantum programming and simulation environment is needed. Small scale quantum computers and classical simulations of quantum computers can influence the development of new quantum algorithms, and the discovery of new quantum algorithms can in turn influence the design of quantum hardware. At a higher level, abstract machine models for hypothetical quantum coprocessors and the larger HPC environments that they operate in are needed to provide both context and direction to algorithm developers. These machine models will themselves require research and development, in a quantum variation of the now familiar HPC co-design problem.

Near-term research problems facing quantum computing are naturally cross-disciplinary as they span the domains of computer science, mathematics, and

physics. These problems cross the traditional boundaries between the program offices within the DOE Office of Science. In particular, whereas ASCR has responsibility for research and development of computer science and mathematics to address future high-performance computing applications, quantum physics falls naturally into the Basic Energy Science (BES), High-Energy Physics (HEP), and Nuclear Physics (NP) program offices.

Some problems in quantum computer science can be undertaken as standalone efforts. For example, quantum algorithm development is nearly solely within the domain of applied mathematics, while high-level languages, programming tools, and abstract machine models are familiar to computer science. The feedback from domain scientists required for these efforts is likely similar to existing HPC co-design efforts. But there are also problems whose solutions will benefit from mixing the physical and computational disciplines. This includes the development of specific scientific applications, such as quantum simulation, that will require close collaboration between the expected use cases and the application development.

V. Conclusions

This report summarizes the findings of the DOE ASCR Workshop on Quantum Computing for Science. The workshop goals included assessing the viability of quantum computing technologies to meet the computational requirements in support of the DOE's science and energy mission, and identifying the potential impact of quantum technologies. With input from the presentations and discussions at the workshop, this report has broadly surveyed those topics in quantum computing that are most likely to bear future relevance on DOE's scientific computing mission.

The variety of research opportunities found in the topics of quantum computational and programming models, quantum algorithms for applied mathematics, and quantum applications for scientific problems indicate that there is great potential for quantum computing to impact the DOE mission. However, much work in the fundamentals of computer science and mathematics will be needed to mature these ideas into computationally viable solutions for DOE. This includes the development of robust programming environments, languages, libraries, compilers, and computing environments for developing quantum applications, as well as the exploration and refinement of quantum algorithms. We expect these tools will be needed well before the availability of large-scale quantum computing resources, for example, to develop and benchmark near-term quantum computing applications using few-qubit processors.

Quantum computing research opportunities in computer science and applied mathematics are naturally aligned with the ASCR supported program areas, while others, specifically scientific application development, are likely to require coordination with BES, HEP, and NP stakeholders. We expect that solving the challenges facing quantum computing will require the broad skill sets represented by all the program offices with the Office of Science. Addressing these near-term research challenges is vital for DOE ASCR supported communities to exploit the burgeoning development of quantum computing technology for its modeling and simulation needs in the physical sciences.

VI. Submitted Whitepapers

[WP1] Helmut G. Katzgraber, Zheng Zhu, Andrew J. Ochoa and Firas Hamze, "Development of benchmarking strategies to understand quantum speedup in novel quantum computing paradigms"

[WP2] Alexey Gorshkov, "Position Paper for the ASCR Workshop on Quantum Computing for Science"

[WP3] Paolo Zanardi and Daniel Lidar, "Quantum Information Processing: The Dissipative Way"

[WP4] Rosa Di Felice, Remo Rohs, Daniel Lidar and Itay Hen, "Quantum Machine Learning Approach to Transcription Factor-DNA Binding"

[WP5] Andrew Kerman, "Position paper for Andrew J. Kerman"

[WP6] Rolando Somma, "High-Precision Quantum Algorithms"

[WP7] Ned Allen, Kristen Pudenz, Adam Salamon, and Greg Tallant, "Quantum Verification and Validation"

[WP8] Philip Powell, Pavlos Vranas, Evan Berkowitz and Enrico Rinaldi, "Towards quantum simulation of neutron star matter and lattice QCD"

[WP9] Itay Hen and Daniel Lidar, "Near-Future Quantum Advantages Beyond Speedup"

[WP10] Alex Mott, Josh Job, Daniel Lidar and Maria Spiropulu, "Higgs Search via Quantum Machine Learning"

[WP11] David B Wecker, "Quantum Chemistry and Materials Algorithms"

[WP12] Rosa Di Felice, Marco Buongiorno Nardelli, Marco Fornari, Stefano Curtarolo, Bobby G. Sumpter, Itay Hen, Stephan Haas and Daniel Lidar, "Quantum Manufacturing"

[WP13] Hari Krovi, "Anyon computation and Kitaev models"

[WP14] Federico Spedalieri and Itay Hen, "Quantum Annealing as an alternative heuristic for solving combinatorial optimization problems"

[WP15] Stephen Jordan, "Quantum Algorithms for Optimization and Simulation"

[WP16] Fred Chong, Ravi Chugh, Margaret Martonosi and John Reppy, “Quantum Programming Languages for Specification and Analysis”

[WP17] Richard P. Muller, Robin Blume-Kohout, Jonathan E. Moussa, Andrew J. Landahl, “Robust quantum simulation for near-term realizable quantum hardware”

[WP18] Graeme Smith, “Position statement”

[WP19] Andrew Childs, “From quantum simulation to quantum algorithms for linear algebra”

[WP20] John A. Smolin, “Position Statement for John A. Smolin”

[WP21] Alex Bocharov, Martin Roetteler and Krysta Svore, “Moving targets: quantum circuit synthesis for fault-tolerant gate sets”

[WP22] Seth Lloyd and Paolo Zanardi, “Position paper for ASCR Workshop on Quantum Computing for Science”

[WP23] Nathan Wiebe, Ashish Kapoor and Krysta Svore, “Quantum Deep Learning”

[WP24] Federico Spedalieri and William Tang, “Potential quantum computing enhancement of machine-learning capabilities in predictive fusion energy applications”

[WP25] Davide Venturelli, Eleanor G. Rieffel, Bryan O’Gorman, Alejandro Perdomo, Dominic Marchand, and Galo Rojo, “Quantum Annealing Programming Techniques for Discrete Optimization Problems”

[WP26] Todd Brun, Daniel Lidar, Ben Reichardt and Paolo Zanardi, “Quantum Error Correction and Fault Tolerance”

[WP27] Itay Hen, Daniel Lidar and Stephan Haas, “Classical Simulations of Large-Scale Quantum Computers”

[WP28] Kesheng Wu, “Software Will Be King, How To Get Started Now?”

[WP29] Alán Aspuru-Guzik, “Quantum Computing for Chemistry and Materials”

[WP30] Dirk Robert Englund, “Efficient quantum simulation using a programmable quantum photonic processor”

[WP31] Irfan Siddiqi, “CoQuNAC: Configurable Quantum Networks for Advanced Computation”

VII. References

1. Feynman, R.P., *Simulating physics with computers.* International Journal of Theoretical Physics, 1982. **21**(6-7): p. 467-488.
2. Shor, P., *Algorithms for Quantum Computation: Discrete Logarithms and Factoring.* 35th Annual Symposium on Foundations of Computer Science, 1994: p. 124-134.
3. Schoning, U., *A probabilistic algorithm for k-SAT and constraint satisfaction problems.* Foundations of Computer Science, 1999. **40**: p. 410-414.
4. Kempe, J., *Quantum random walks: An introductory overview.* Contemporary Physics, 2003. **44**(4): p. 307-327.
5. Venegas-Andraca, S.E., *Quantum walks: a comprehensive review.* Quantum Information Processing, 2012. **11**(5): p. 1015-1106.
6. Childs, A.M., et al., *Exponential algorithmic speedup by a quantum walk.* STOC 2003: Proceedings of the thirty fifth annual ACM symposium on Theory of Computing, 2003. **35**: p. 59.
7. Grover, L.K., *A fast quantum mechanical algorithm for database search.* ACM Symposium on the Theory of Computation, 1996. **96**: p. 212.
8. Ambainis, A., *Quantum walks and their algorithmic applications.* International Journal of Quantum Information, 2003. **01**(04): p. 507-518.
9. Ambainis, A., *Quantum Walk Algorithm for Element Distinctness.* Siam Journal On Computing, 2007. **37**(1): p. 210-239.
10. Childs, A.M., *Universal Computation by Quantum Walk.* Physical Review Letters, 2009. **102**: p. 180501.
11. Kitaev, A.Y., *Quantum measurements and the Abelian Stabilizer Problem.* arXiv:quant-ph/9511026, 1995.
12. Hales, L. and S. Hallgren. *An improved quantum Fourier transform algorithm and applications*. in *41st Annual Symposium on Foundations of Computer Science*. 2000. IEEE Comput. Soc.
13. Harrow, A.W., A. Hassidim, and S. Lloyd, *Quantum Algorithm for Linear Systems of Equations.* Physical Review Letters, 2009. **103**(15): p. 150502.
14. Aharonov, D., V. Jones, and Z. Landau, *A Polynomial Quantum Algorithm for Approximating the Jones Polynomial - Springer.* Algorithmica, 2009.
15. Farhi, E., et al., *Quantum Computation by Adiabatic Evolution.* arXiv:quant-ph/0001106, 2000.
16. Feynman, R.P., *Quantum mechanical computers.* Found. Phys., 1986. **16**(6): p. 507--531.
17. Kitaev, A.Y., A. Shen, and M.N. Vyalyi, *Classical and Quantum Computation*. 2002: American Mathematical Society.
18. Aharonov, D., et al., *Adiabatic quantum computation is equivalent to standard quantum computation.* arXiv:quant-ph/0405098, 2004: p. 42--51.
19. Kempe, J., A.Y. Kitaev, and O. Regev, *The complexity of the local Hamiltonian problem.* Siam Journal On Computing, 2006. **35**: p. 1070-1097.
20. Oliveira, R. and B.M. Terhal, *The complexity of quantum spin systems on a two-dimensional square lattice.* Quant. Inf. Comp., 2008. **8**(10): p. 0900.

21. Mizel, A., D.A. Lidar, and M. Mitchell, *Simple proof of equivalence between adiabatic quantum computation and the circuit model.* Phys. Rev. Lett., 2007. **99**: p. 070502.
22. Aharonov, D., et al., *The power of quantum systems on a line.* arXiv:0705.4077, 2007.
23. Chase, B.A. and A.J. Landahl, *Universal quantum walks and adiabatic algorithms by 1D Hamiltonians.* arxiv:0802.1207, 2008.
24. Gottesman, D. and S. Irani, *The quantum and classical complexity of translationally invariant tiling and Hamiltonian problems.* arXiv:1210.8395, 2009.
25. Boixo, S., et al., *Experimental signature of programmable quantum annealing.* Nature Communications, 2013. **4**: p. 1.
26. Boixo, S., et al., *Evidence for quantum annealing with more than one hundred qubits : Nature Physics : Nature Publishing Group.* Nature Physics, 2014. **10**: p. 218.
27. Smolin, J.A. and G. Smith, *Classical signature of quantum annealing.* arXiv:1305.4904, 2013.
28. Wang, L., et al., *Comment on: "Classical signature of quantum annealing".* arXiv:1305.5837, 2013.
29. Rønnow, T.F., et al., *Defining and detecting quantum speedup.* Science, 2014. **345**: p. 420.
30. Albash, T., et al., *Reexamining classical and quantum models for the D-Wave One processor.* EPJ-ST, 2015. **224**: p. 111.
31. Shin, S.W., et al., *How "Quantum" is the D-Wave Machine?* arXiv:1401.7087, 2014.
32. Shin, S.W., et al., *Comment on ``Distinguishing classical and quantum models for the D-Wave device.* arXiv:1404.6499, 2014.
33. Lanting, T., et al., *Entanglement in a quantum annealing processor.* arXiv:1401.3500, 2014.
34. Boixo, S., et al., *Computational role for multiqubit tunneling in a quantum annealer.* 2015.
35. Choi, V., *Minor-embedding in adiabatic quantum computation: I. The parameter setting problem.* Quant. Inf. Proc., 2008. **7**(5): p. 193--209.
36. Choi, V., *Minor-embedding in adiabatic quantum computation: II. Minor-universal graph design.* Quant. Inf. Proc., 2011. **10**: p. 343.
37. Klymko, C., B.D. Sullivan, and T.S. Humble, *Adiabatic quantum programming: Minor embedding with hard faults.* arXiv:1210.8395, 2012.
38. Jordan, S.P., E. Farhi, and P.W. Shor, *Error-correcting codes for adiabatic quantum computation.* Physical Review A, 2006. **74**(5): p. 052322.
39. Lidar, D.A., *Towards fault tolerant adiabatic quantum computation.* Phys. Rev. Lett., 2008. **100**: p. 160506.
40. Ganti, A., U. Onunkwo, and K. Young, *A family of $[[6k, 2k, 2]]$ codes for practical, scalable adiabatic quantum computation.* 2013.
41. Pudenz, K.L., T. Albash, and D.A. Lidar, *Error-corrected quantum annealing with hundreds of qubits.* Nature Communications, 2014. **5**.

42. Bookatz, A.D., E. Farhi, and L. Zhou, *Error suppression in Hamiltonian based quantum computation using energy penalties.* arXiv preprint arXiv:1407.1485, 2014.
43. Young, K.C., R. Blume-Kohout, and D.A. Lidar, *Adiabatic quantum optimization with the wrong Hamiltonian.* 2013.
44. Young, K.C., M. Sarovar, and R. Blume-Kohout, *Error suppression and error correction in adiabatic quantum computation I: techniques and challenges.* Phys. Rev. X, 2013. **3**: p. 041013.
45. Sarovar, M. and K. Young, *Error suppression and error correction in adiabatic quantum computation II: non-equilibrium dynamics.* New Journal Of Physics, 2013. **15**: p. 125032.
46. Marvian, I. and D.A. Lidar, *Quantum error suppression with communting Hamiltonians: Two-local is too local.* Phys. Rev. Lett., 2014. **113**: p. 260504.
47. Mizel, A., *Fault-tolerant, universal adiabatic quantum computation.* arXiv:1403.7694, 2014.
48. Oreshkov, O., *Holonomic Quantum Computation in Subsystems.* Phys. Rev. Lett., 2009. **103**(9): p. 090502.
49. Bacon, D., et al., *Adiabatic topological quantum computing.* Unpublished notes., 2010.
50. Zheng, Y.-C. and T.A. Brun, *Fault-tolerant holonomic quantum computation in surface codes.* Phys. Rev. A, 2015. **91**: p. 022302.
51. Farhi, E., J. Goldstone, and S. Gutmann, *A Quantum Approximate Optimization Algorithm Applied to a Bounded Occurrence Constraint Problem.* arXiv:1412.6062, 2014.
52. Farhi, E., J. Goldstone, and S. Gutmann, *A quantum approximate optimization algorithm.* arXiv:1411.4028, 2014.
53. Barreiro, J.T., et al., *An open-system quantum simulator with trapped ions.* Nature, 2011. **470**: p. 486.
54. Schindler, P., et al., *Quantum simulation of open-system dynamical maps with trapped ions.* Nature Phys., 2013. **9**: p. 361.
55. Verstraete, F., M.M. Wolf, and J.I. Cirac, *Quantum computation, quantum state engineering, and quantum phase transitions driven by dissipation.* New Journal Of Physics, 2009. **5**: p. 633.
56. Zanardi, P. and L.C. Venuti, *Geometry, robustness, and emerging unitarity in dissipation-projected dynamics.* 2014.
57. Zanardi, P. and L.C. Venuti, *Coherent quantum dynamics in steady-state manifolds of strongly dissipative systems.* Phys. Rev. Lett., 2014. **113**: p. 240406.
58. Nayak, C., et al., *Non-Abelian anyons and topological quantum computation.* Reviews Of Modern Physics, 2008. **80**(3): p. 1083.
59. Wang, Z., *Topological Quantum Computation*, in *Topological Quantum Computing*. 2010, AMS. p. 115.
60. Pachos, J.K., *Introduction to topological quantum computation*. 2012: Cambridge University Press.
61. Leinaas, J.M. and J. Myrheim, *On the theory of identical particles.* Il Nuovo Cimento B Series 11, 1977. **37**(1): p. 1-23.

62. Goldin, G.A., R. Menikoff, and D.H. Sharp, *Representations of a local current algebra in nonsimply connected space and the Aharonov–Bohm effect.* Journal of Mathematical Physics, 1981. **22**(8): p. 1664-1668.
63. Wilczek, F., *Quantum mechanics of fractional-spin particles.* Physical Review Letters, 1982. **49**(14): p. 957.
64. Wilczek, F., *Magnetic flux, angular momentum, and statistics.* Physical Review Letters, 1982. **48**(17): p. 1144.
65. Wilczek, F., *Remarks on dyons.* Physical Review Letters, 1982. **48**: p. 1146-1149.
66. Arovas, D., J.R. Schrieffer, and F. Wilczek, *Fractional statistics and the quantum Hall effect.* Physical review letters, 1984. **53**(7): p. 722.
67. Freedman, M.H., A. Kitaev, and Z. Wang, *Simulation of Topological Field Theories by Quantum Computers.* Communications in Mathematical Physics, 2002. **227**(3): p. 587-603.
68. Beenakker, C.W.J., *Search for Majorana fermions in superconductors.* arXiv preprint arXiv:1112.1950, 2011.
69. Alicea, J., *New directions in the pursuit of Majorana fermions in solid state systems.* Reports on Progress in Physics, 2012. **75**(7): p. 076501.
70. Leijnse, M. and K. Flensberg, *Introduction to topological superconductivity and Majorana fermions.* Semiconductor Science and Technology, 2012. **27**(12): p. 124003.
71. Garnerone, S., P. Zanardi, and D.A. Lidar, *Adiabatic quantum algorithm for search engine ranking.* Physical review letters, 2012. **108**(23): p. 230506.
72. Alagic, G., et al., *Estimating Turaev-Viro three-manifold invariants is universal for quantum computation.* Phys. Rev. A, 2010. **82**: p. 40302(R).
73. Jordan, S.P. and G. Alagic, *Approximating the Turaev-Viro invariant of mapping tori is complete for one clean qubit*, in *Theory of Quantum Computation, Communication, and Cryptography*. 2014, Springer. p. 53-72.
74. Alagic, G., I.V. Bering, and A. Edgar, *Quantum algorithms for invariants of triangulated manifolds.* arXiv preprint arXiv:1108.5424, 2011.
75. Freedman, M.H., M. Larsen, and Z. Wang, *A Modular Functor Which is Universal for Quantum Computation.* Communications in Mathematical Physics, 2002. **227**(3): p. 605-622.
76. Wocjan, P. and J. Yard, *The Jones polynomial: quantum algorithms and applications in quantum complexity theory.* arXiv preprint quant-ph/0603069, 2006.
77. Rasetti, M., S. Garnerone, and A. Marzuoli, *An efficient quantum algorithm for colored Jones polynomials.* International Journal of Quantum Information, 2008. **6**(supp01): p. 773-778.
78. Shor, P.W. and S.P. Jordan, *Estimating Jones polynomials is a complete problem for one clean qubit.* Quantum Information & Computation, 2008. **8**(8): p. 681-714.
79. Jordan, S.P. and P. Wocjan, *Estimating Jones and HOMFLY polynomials with one clean qubit.* Quant. Inf. Comp., 2009. **9**: p. 264.
80. Aharonov, D. and I. Arad, *The BQP-hardness of approximating the Jones polynomial.* New Journal of Physics, 2011. **13**(3): p. 035019.

81. Kitaev, A.Y., *Quantum computations: algorithms and error correction.* Russian Mathematical Surveys, 1997. **52**(6): p. 1191-1249.
82. Bombin, H. and M.A. Martin-Delgado, *Topological quantum distillation.* Physical review letters, 2006. **97**(18): p. 180501.
83. Koenig, R., G. Kuperberg, and B.W. Reichardt, *Quantum computation with Turaev–Viro codes.* Annals of Physics, 2010. **325**(12): p. 2707-2749.
84. Kitaev, A.Y., *Fault-tolerant quantum computation by anyons.* Annals of Physics, 2003. **303**(1): p. 2-30.
85. Bonesteel, N.E., et al., *Braid topologies for quantum computation.* Physical review letters, 2005. **95**(14): p. 140503.
86. Simon, S.H., et al., *Topological quantum computing with only one mobile quasiparticle.* Physical review letters, 2006. **96**(7): p. 070503.
87. Bravyi, S., *Universal quantum computation with the ν= 5/ 2 fractional quantum Hall state.* Physical Review A, 2006. **73**(4): p. 042313.
88. Hormozi, L., et al., *Topological quantum compiling.* Physical Review B, 2007. **75**(16): p. 165310.
89. McDonald, R.B. and H.G. Katzgraber, *Genetic braid optimization: A heuristic approach to compute quasiparticle braids.* Physical Review B, 2013. **87**(5): p. 054414.
90. Dennis, E., et al., *Topological quantum memory.* Journal of Mathematical Physics, 2002. **43**(9): p. 4452-4505.
91. Bonderson, P., et al., *A blueprint for a topologically fault-tolerant quantum computer.* arXiv preprint arXiv:1003.2856, 2010.
92. Fujii, K., *Quantum Computation with Topological Codes: from qubit to topological fault-tolerance.* arXiv preprint arXiv:1504.01444, 2015.
93. Thaker, D.D., et al., *Quantum memory hierarchies: efficient designs to match available parallelism in quantum computing.* arXiv preprint quant-ph/0604070, 2006.
94. Steane, A.M., *How to build a 300 bit, 1 giga-operation quantum computer.* Quant. Inf. Comp., 2007. **7**(3): p. 171.
95. Isailovic, N., et al. *Running a quantum circuit at the speed of data*. IEEE Computer Society.
96. Ganti, A. and R. Somma, *On the gap of Hamiltonians for the adiabatic simulation of quantum circuits.* International Journal of Quantum Information, 2013. **11**(07).
97. Zanardi, P. and M. Rasetti, *Holonomic quantum computation.* Physics Letters A, 1999. **264**(2): p. 94-99.
98. Cesare, C., et al., *Adiabatic topological quantum computing.* arXiv preprint arXiv:1406.2690, 2014.
99. Zheng, Y.-C. and T.A. Brun, *Fault-tolerant scheme of holonomic quantum computation on stabilizer codes with robustness to low-weight thermal noise.* Physical Review A, 2014. **89**(3): p. 032317.
100. Fowler, A.G. and S.J. Devitt, *A bridge to lower overhead quantum computation.* arXiv preprint arXiv:1209.0510, 2012.
101. Fowler, A.G., S.J. Devitt, and C. Jones, *Surface code implementation of block code state distillation.* Scientific reports, 2013. **3**.

102. Mishra, P. and A. Fowler, *Resource comparison of two surface code implementations of small angle Z rotations.* arXiv preprint arXiv:1406.4948, 2014.
103. Raussendorf, R. and J. Harrington, *Fault-tolerant quantum computation with high threshold in two dimensions.* Physical review letters, 2007. **98**(19): p. 190504.
104. Stace, T.M. and S.D. Barrett, *Error correction and degeneracy in surface codes suffering loss.* Physical Review A, 2010. **81**(2): p. 022317.
105. Wang, D.S., A.G. Fowler, and L.C.L. Hollenberg, *Surface code quantum computing with error rates over 1%.* Physical Review A, 2011. **83**(2): p. 020302.
106. Fowler, A.G., *Optimal complexity correction of correlated errors in the surface code.* arXiv preprint arXiv:1310.0863, 2013.
107. Paler, A., et al., *Software Pauli Tracking for Quantum Computation.* arXiv preprint arXiv:1401.5872, 2014.
108. Devitt, S.J., *Classical Control of Large-Scale Quantum Computers*, in *Reversible Computation*. 2014, Springer. p. 26-39.
109. Paetznick, A. and A.G. Fowler, *Quantum circuit optimization by topological compaction in the surface code.* arXiv preprint arXiv:1304.2807, 2013.
110. Hastings, M.B., *Decoding in Hyperbolic Spaces: LDPC Codes With Linear Rate and Efficient Error Correction.* arXiv preprint arXiv:1312.2546, 2013.
111. Bravyi, S. and M.B. Hastings. *Homological product codes*. ACM.
112. Leverrier, A., J.-P. Tillich, and G. Zémor, *Quantum Expander Codes.* arXiv preprint arXiv:1504.00822, 2015.
113. Paetznick, A. and B.W. Reichardt, *Universal fault-tolerant quantum computation with only transversal gates and error correction.* Physical review letters, 2013. **111**(9): p. 090505.
114. Jochym-O'Connor, T. and R. Laflamme, *Using concatenated quantum codes for universal fault-tolerant quantum gates.* Physical review letters, 2014. **112**(1): p. 010505.
115. Anderson, J.T., G. Duclos-Cianci, and D. Poulin, *Fault-tolerant conversion between the Steane and Reed-Muller quantum codes.* Physical review letters, 2014. **113**(8): p. 080501.
116. Bombin, H., *Dimensional Jump in Quantum Error Correction.* arXiv preprint arXiv:1412.5079, 2014.
117. Gottesman, D., *Fault-tolerant quantum computation with constant overhead.* Quantum Information & Computation, 2014. **14**(15-16): p. 1338-1372.
118. Brun, T.A., et al., *Teleportation-based Fault-tolerant Quantum Computation in Multi-qubit Large Block Codes.* arXiv preprint arXiv:1504.03913, 2015.
119. Viola, L., E. Knill, and S. Lloyd, *Dynamical decoupling of open quantum systems.* Physical Review Letters, 1999. **82**(12): p. 2417.
120. Khodjasteh, K. and L. Viola, *Dynamically error-corrected gates for universal quantum computation.* Physical review letters, 2009. **102**(8): p. 080501.
121. Khodjasteh, K., H. Bluhm, and L. Viola, *Automated synthesis of dynamically corrected quantum gates.* Physical Review A, 2012. **86**(4): p. 042329.

122. Doherty, A.C., et al., *Quantum feedback control and classical control theory.* Physical Review A, 2000. **62**(1): p. 012105.
123. Werschnik, J. and E.K.U. Gross, *Quantum optimal control theory.* Journal of Physics B: Atomic, Molecular and Optical Physics, 2007. **40**(18): p. R175.
124. Dong, D. and I.R. Petersen, *Quantum control theory and applications: a survey.* IET Control Theory & Applications, 2010. **4**(12): p. 2651-2671.
125. Jones, N.C., et al., *Layered architecture for quantum computing.* Physical Review X, 2012. **2**(3): p. 031007.
126. Ross, N.J. and P. Selinger, *Optimal ancilla-free Clifford+ T approximation of z-rotations.* arXiv preprint arXiv:1403.2975, 2014.
127. Ross, N.J., *Optimal ancilla-free Pauli+ V approximation of z-rotations.* arXiv preprint arXiv:1409.4355, 2014.
128. Bocharov, A., M. Roetteler, and K.M. Svore, *Efficient synthesis of universal repeat-until-success quantum circuits.* Physical review letters, 2015. **114**(8): p. 080502.
129. Bocharov, A., M. Roetteler, and K.M. Svore, *Efficient synthesis of probabilistic quantum circuits with fallback.* Phys. Rev. A, 2015. **91**: p. 052317.
130. Trout, C.J. and K.R. Brown, *Magic state distillation and gate compilation in quantum algorithms for quantum chemistry.* International Journal of Quantum Chemistry, 2015. **2015**.
131. Selinger, P. and B. Valiron, *A linear-non-linear model for a computational call-by-value lambda calculus*, in *Foundations of Software Science and Computational Structures*. 2008, Springer. p. 81-96.
132. Kassal, I., et al., *Polynomial-time quantum algorithm for the simulation of chemical dynamics.* Proceedings of the National Academy of Sciences, 2008. **105**(48): p. 18681-18686.
133. Aspuru-Guzik, A., et al., *Simulated quantum computation of molecular energies.* Science, 2005. **309**(5741): p. 1704-1707.
134. Wecker, D., et al., *Can quantum chemistry be performed on a small quantum computer?* arXiv:1312.1695, 2013.
135. McClean, J.R., et al., *Exploiting Locality in Quantum Computation for Quantum Chemistry - The Journal of Physical Chemistry Letters (ACS Publications).* Journal Of Physical Chemistry Letters, 2014. **5**: p. 4368-4380.
136. Hastings, M.B., et al., *Improving quantum algorithms for quantum chemistry.* Quantum Information & Computation, 2015. **15**(1-2): p. 1-21.
137. Babbush, R., et al., *Chemical basis of Trotter-Suzuki errors in quantum chemistry simulation.* Physical Review A, 2015. **91**(2): p. 022311.
138. Toloui, B. and P.J. Love, *Quantum Algorithms for Quantum Chemistry based on the sparsity of the CI-matrix.* arXiv:1312.2579, 2013.
139. Jordan, S.P., K.S.M. Lee, and J. Preskill, *Quantum algorithms for quantum field theories.* Science, 2012. **336**(6085): p. 1130-1133.
140. Welch, J., et al., *Efficient quantum circuits for diagonal unitaries without ancillas.* New Journal Of Physics, 2014. **16**(3): p. 033040.
141. Berry, D.W., A.M. Childs, and R. Kothari, *Hamiltonian simulation with nearly optimal dependence on all parameters.* arXiv:1501.01715, 2015.

142. Peruzzo, A., et al., *A variational eigenvalue solver on a photonic quantum processor.* Nature Communications, 2014. **5**.
143. Georgescu, I.M., S. Ashhab, and F. Nori, *Quantum simulation.* Reviews Of Modern Physics, 2014. **86**(1): p. 153-185.
144. Greiner, M., et al., *Quantum phase transition from a superfluid to a Mott insulator in a gas of ultracold atoms.* Nature, 2002. **415**(6867): p. 39-44.
145. Friedenauer, A., et al., *Simulating a quantum magnet with trapped ions.* Nature Publishing Group, 2008. **4**(10): p. 757-761.
146. Zhou, T., et al., *Quantitative modeling of transcription factor binding specificities using DNA shape.* Procedings of the National Academy of Sciences, 2015.
147. Sumpter, B.G., C. Getino, and D.W. Noid, *Theory and Applications of Neural Computing in Chemical Sciences.* Annual Reviews of Physical Chemistry, 1994. **45**: p. 439.
148. Lorenz, S., A. Groß, and M. Scheffler, *Representing high-dimensional potential-energy surfaces for reactions at surfaces by neural networks.* Chemical Physics Letters, 2004. **395**(4): p. 210-215.
149. Aaronson, S. and A. Arkhipov, *The Computational Complexity of Linear Optics.* arXiv:1011.3245, 2010.
150. Huh, J., et al., *Boson Sampling for Molecular Vibronic Spectra.* arXiv:1412.8427, 2014.
151. Novak, E., *Quantum Complexity of Integration.* Journal of Complexity, 2001.
152. Nayak, A. and F. Wu, *The quantum query complexity of approximating the median and related statistics.* STOC 1999: Procedings of the thirty first ACM Symposium on Theory of Computing, 1999. **31**: p. 384.
153. Hogg, T., *Highly Structured Searches with Quantum Computers.* Physical Review Letters, 1998. **80**: p. 2473.
154. Jordan, S.P., *Quantum Computation Beyond the Circuit Model*, 2008, Massachusetts Institute of Technology: arxiv:0809.2307.
155. Rieffel, E.G., et al., *A case study in programming a quantum annealer for hard operational planning problems.* arXiv:1407.2887, 2014.
156. Williams, V.V., *Multiplying matrices faster than coppersmith-winograd.* Proceedings of the forty-fourth annual ACM symposium on the Theory of Computing, 2012. **44**: p. 887-898.
157. Clader, B.D., B.C. Jacobs, and C.R. Sprouse, *Preconditioned Quantum Linear System Algorithm.* Physical Review Letters, 2013. **110**: p. 250504.
158. Berry, D.W., *High-order quantum algorithm for solving linear differential equations.* Journal of Physics A-Mathematical and Theoretical, 2014. **47**: p. 105301.
159. Wang, G., *Quantum Algorithms for Approximating the Effective Resistances in Electrical Networks.* arXiv:1311.1851, 2013.
160. Wiebe, N., D. Braun, and S. Lloyd, *Quantum Algorithm for Data Fitting.* Physical Review Letters, 2012. **109**(5): p. 050505.
161. Lloyd, S., M. Mohseni, and P. Rebentrost, *Quantum algorithms for supervised and unsupervised machine learning.* arXiv:1307.0411, 2013.

162. Lloyd, S., M. Mohseni, and P. Rebentrost, *Quantum principal component analysis.* Nature Physics, 2014. **10**(9): p. 631-633.
163. Lloyd, S., S. Garnerone, and P. Zanardi, *Quantum algorithms for topological and geometric analysis of big data.* arXiv:1408.3106, 2014.
164. Rebentrost, P., M. Mohseni, and S. Lloyd, *Quantum Support Vector Machine for Big Data Classification.* Physical Review Letters, 2014. **113**: p. 130503.
165. Aaronson, S. *Quantum Machine Learning Algorithms: Read the Fine Print*. Available from: http://www.scottaaronson.com/papers/qml.pdf.
166. Ta-Shma, A., *Inverting well conditioned matrices in quantum logspace.* Proceedings of the forty-fifth annual ACM symposium on the Theory of Computing, 2013. **45**: p. 881.
167. Grover, L.K., *Quantum Mechanics Helps in Searching for a Needle in a Haystack.* Physical Review Letters, 1997. **79**(2): p. 325.
168. Brassard, G., et al., *Quantum Amplitude Amplification and Estimation.* arXiv:quant-ph/0005055, 2000.
169. Szegedy, M., *Spectra of Quantized Walks and a sqrt(delta-epsilon) rule.* arXiv:quant-ph/0401053, 2004.
170. Somma, R., S. Boixo, and H. Barnum, *Quantum Simulated Annealing.* arXiv:0712.1008, 2007.
171. Buhrman, H. and R. Špalek. *Quantum verification of matrix products*. in *SODA 2006: Procedings of the seventh annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics.
172. Bernstein, D.J., et al. *Quantum algorithms for the subset-sum problem*. cr.yp.to 2013; Available from: http://cr.yp.to/qsubsetsum.html.
173. Yao, A.C.-C., *On computing the minima of quadratic forms (Preliminary Report).* seventh annual ACM symposium, 1975: p. 23-26.
174. Jordan, S.P., *Fast Quantum Algorithm for Numerical Gradient Estimation.* Physical Review Letters, 2005. **95**(5): p. 050501.
175. Aharonov, D., et al., *Adiabatic Quantum Computation Is Equivalent to Standard Quantum Computation.* SIAM Review, 2008. **50**(4): p. 755-787.
176. Farhi, E., et al., *Performance of the quantum adiabatic algorithm on random instances of two optimization problems on regular hypergraphs.* Physical Review A, 2012. **86**: p. 052334.
177. Crosson, E., et al., *Different Strategies for Optimization Using the Quantum Adiabatic Algorithm.* arXiv:1401.7320, 2014.
178. Reichardt, B.W., *The quantum adiabatic optimization algorithm and local minima.* Proceedings of the thirty-sixth annual ACM symposium on Theory of computing, 2004. **36**: p. 502.
179. Farhi, E., J. Goldstone, and S. Gutmann, *Quantum Adiabatic Evolution Algorithms versus Simulated Annealing.* arXiv:quant-ph/0201031, 2002.
180. Jarret, M. and S.P. Jordan, *Adiabatic optimization without local minima.* Quantum Information & Computation, 2015. **15**(3-4): p. 181.

Appendix A: Program Committee

The workshop agenda was set by a program committee of scientists from university, industry, and government stakeholders. Table A.1 lists the names and affiliations of program committee members.

Table A.1 Names and Affiliations of the workshop program committee

Name	Affiliation
Paul Alsing	Air Force Research Laboratory
Jim Amundson	Fermi Lab
Alán Aspuru-Guzik	Harvard University
Wim van Dam	University of California Santa Barbara
Edward Farhi	Massachusetts Institute of Technology
Chris Fuchs	Perimeter Institute
Frank Gaitan	Laboratory for Physical Sciences
Travis Humble	Oak Ridge National Laboratory
Stephen Jordan	National Institute of Standards and Technology
Andrew Landahl	Sandia National Laboratories
Robert Lucas	University of Southern California Information Sciences Institute
John Preskill	California Institute of Technology
Rick Muller	Sandia National Laboratories
Krysta Svore	Microsoft
Carl Williams	National Institute of Standards and Technology

Appendix B: Workshop Agenda

Presentations from the workshop are available for download from

http://www.csm.ornl.gov/workshops/ascrqcs2015/agenda.html

Table B.1 Workshop Agenda

Tuesday, February 17, 2015

8:00am - 9:00am	Continental Breakfast	
9:00am - 9:30am	Welcome and Introduction – DOE Perspective Ballroom CD	Steve Binkley Associate Director of Advanced Scientific Computing Research
9:30am - 10:15am	***Topic 2***: Quantum Algorithms for Applied Mathematics Ballroom CD	Stephen Jordan National Institute of Standards and Technology
10:15am - 10:30am	Break	
10:30am - 11:15am	***Topic 1***: Quantum Software Architecture: Mapping Quantum Algorithms to Devices Ballroom CD	Krysta Svore Microsoft Corporation
11:15am - 12:00pm	***Topic 3***: Physical Science Application Area, Including Quantum Simulation Ballroom CD	Peter Love Haverford University
12:00pm - 1:00pm	Working Lunch – QIS Impact of HEP Ballroom AB	John Preskill California Institute of Technology
1:00pm - 3:00pm	Breakout Sessions: Topic 1 – Jasmine Room Topic 2 – Juniper Room Topic 3 – Lavender Room	Topic 1 – Andrew Landahl (SNL) Topic 2 – Stephen Jordan (NIST) Topic 3 – Rick Muller (SNL)
3:00pm - 3:30pm	Break	
3:30pm - 5:00pm	Breakout Sessions Resume: Topic 1 – Jasmine Room Topic 2 – Juniper Room Topic 3 – Lavender Room	Topic 1 – Andrew Landahl (SNL) Topic 2 – Stephen Jordan (NIST) Topic 3 – Rick Muller (SNL)
5:00pm	Adjourn	

Wednesday, February 18, 2015

7:00am - 8:00am	Continental Breakfast	
8:00am - 8:30a.m	Review and Discussion of Breakout Sessions	Andrew Landahl (SNL) Travis Humble (ORNL) Rick Muller (SNL)
8:30am - 10:00a.m	Breakout Sessions Resume: Topic 1 – Jasmine Room Topic 2 – Juniper Room Topic 3 – Lavender Room	Topic 1 – Andrew Landahl (SNL) Topic 2 – Stephen Jordan (NIST) Topic 3 – Rick Muller (SNL)
10:00am - 10:30a.m	Break	
10:30am - 11:00a.m	Review of Breakout Topic 1 - Models of Quantum Computation, Quantum Error Correction, and their Programming Environments	Andrew Landahl (SNL)
11:00am - 11:30am	Review of Breakout Topic 2 - Quantum Algorithms for Applied Mathematics and Linear Algebra	Stephen Jordan (NIST)
11:30am - 12:00pm	Review of Breakout Topic 3 - Physical Science Application Area, Including Quantum Simulation	Rick Muller (SNL)
12:00pm	Adjourn	

Breakout Session 1: Computational Models

Table B.2 Agenda for breakout session 1 on computational models

Topic 1

Quantum Software Architecture: Mapping Quantum Algorithms to Devices
Andrew Landahl – Sandia National Laboratories

Quantum error correction and fault tolerance
Todd Brun - USC

Dissipation-assisted quantum information processing
Todd Brun (on behalf of Paolo Zanardi) - USC

Anyon computation and Kitaev models
Hari Krovi - Raytheon BBN

How do we best utilize alternative computational models?
Alexey Gorshkov - JQI

Moving targets: Quantum circuit synthesis for fault-tolerant gate sets
Martin Roetteler - MSR

Quantum programming languages for specification and analysis
Fred Chong - UCSB

Solving practical problems with quantum computing hardware
K. John Wu - LBNL

Seeking quantum speedup through spin glasses: Learning from statistical physics
Helmut Katzgraber - TX A&M

Classical simulations of large-scale quantum computers
Itay Hen - USC/ISI

What problems can small quantum computers solve (when augmented with large classical computers)?
John Smolin (on behalf of Graeme Smith) - IBM

How can scientific skepticism help us make good choices about quantum computing investment?
John Smolin - IBM

Breakout Session 2: Quantum Algorithms for Applied Mathematics

Table B.3 Agenda for breakout session on quantum algorithms for applied mathematics

Topic 2
Quantum Algorithms for Applied Mathematics
Stephen Jordan – National Institute for Standards and Technology
Travis Humble – Oak Ridge National Laboratory

From quantum simulation to quantum algorithms for linear algebra
Andrew Childs - University of Maryland

High-Precision Quantum Algorithms
Rolando Somma - Los Alamos National Laboratory

Quantum Algorithms for Optimization and Simulation
Stephen Jordan - National Institute of Standards and Technology

Near-Future Quantum Advantages Beyond Speedup
Itay Hen - University of Southern California

Quantum Annealing Programming Techniques For Discrete Optimization Problems
Davide Venturelli - NASA Ames Research Center

Quantum Annealing as an Alternative Heuristic for Solving Combinatorial Optimization Problems
Federico Spedalieri - University of Southern California

Breakout Session 3: Physical Science Applications

Table B.4 Agenda for breakout session on physical science applications

Topic 3

Quantum Algorithms for Quantum Simulation and Science Applications
Richard Muller – Sandia National Laboratories

Quantum Chemistry and Materials Algorithms
David Wecker - Microsoft Research

Robust Quantum Simulation for Near-term Realizable Quantum Hardware
Richard Muller - Sandia National Laboratories

Towards quantum simulation of neutron star matter and lattice QCD
Philip Powell – Lawrence Livermore National Laboratories

Configurable Quantum Networks for Advanced Computation
Irfan Siddiqi - Lawrence Berkeley National Laboratory

Quantum Deep Learning
Nathan Wiebe - Microsoft Research

Higgs Search Via Machine Learning
Maria Spiropulu – California Institute of Technology

Potential Quantum Computing Enhancement of Machine Learning in Predictive Fusion Energy Applications
William Tang - Princeton University