OPEN ACCESS
IOP Publishing

Reports on Progress in Physics

Rep. Prog. Phys. 87 (2024) 116001 (51pp)

https://doi.org/10.1088/1361-6633/ad85f0

Review

Quantum algorithms for scientific computing

R Au-Yeung[1,*], B Camino[2], O Rathore[3] and V Kendon[1,*]

[1] Department of Physics, University of Strathclyde, Glasgow G4 0NG, United Kingdom
[2] Department of Chemistry, UCL, London WC1E 6BT, United Kingdom
[3] Department of Physics, Durham University, Durham DH1 3LE, United Kingdom

E-mail: rhonda.au-yeung@strath.ac.uk and viv.kendon@strath.ac.uk

Received 23 January 2024, revised 9 August 2024
Accepted for publication 11 October 2024
Published 29 October 2024

Corresponding editor: Professor Zi Yang Meng

Abstract

Quantum computing promises to provide the next step up in computational power for diverse application areas. In this review, we examine the science behind the quantum hype, and the breakthroughs required to achieve true quantum advantage in real world applications. Areas that are likely to have the greatest impact on high performance computing (HPC) include simulation of quantum systems, optimization, and machine learning. We draw our examples from electronic structure calculations and computational fluid dynamics which account for a large fraction of current scientific and engineering use of HPC. Potential challenges include encoding and decoding classical data for quantum devices, and mismatched clock speeds between classical and quantum processors. Even a modest quantum enhancement to current classical techniques would have far-reaching impacts in areas such as weather forecasting, aerospace engineering, and the design of 'green' materials for sustainable development. This requires significant effort from the computational science, engineering and quantum computing communities working together.

Keywords: quantum algorithms, quantum computing, scientific computing

Contents

[*] Authors to whom any correspondence should be addressed.

1. Introduction

The largest quantum computers are now approaching a size and capability where it is no longer possible to efficiently simulate them classically, even using the largest available near-exascale classical computing capabilities. For most scientific and engineering applications, 'exascale' means the system's aggregate performance crosses a threshold of 10^{18} IEEE 754 double precision (64-bit) operations (multiplications and/or additions) per second (exaFLOPS). To put this in context, the Sony PlayStation 5 digital edition is listed as having a peak of $\sim$10 teraFLOPS ($\sim 10^{12}$ FLOPS), a million times smaller. Classical machines just below exascale have been used to verify quantum supremacy claims. For example, random circuit sampling and Gaussian boson sampling methods respectively used the Summit ($\sim$150 petaFLOPs) [1] and Sunway TaihuLight ($\sim$93 petaFLOPs) [2] supercomputers. These supremacy tasks are not yet known to have any industrial or other useful applications, and the classical algorithms have since been improved (see section 2.7.2). However, these supremacy headlines do indicate that we are nearing the time when a quantum co-processor running a well-chosen sub-task could provide a significant boost to large scale computational capabilities.

Reaching for exascale computing calls for a paradigm shift from traditional computing advances [3, 4]. Historical advances in computing were achieved via increasing clock speed and memory performance, adding more cores and parallelization. Moore's law [5] predicts that the exponential growth of silicon-based transistors on computer chips is about to hit physical limits [6, 7]. The growth predicted was borne out by major chip and computer system vendors until about 2005, when CPU clock speeds reached around 4 GHz (figure 1). Significantly increasing the clock speed beyond this would require enormous effort to cool the processor to prevent malfunction or permanent hardware damage from overheating.

Launched in 1993, the TOP500 project assembles and maintains a list of the world's 500 most powerful computer systems [9]. As of June 2024, Oak Ridge National Laboratory's Frontier system and Argonne National Laboratory's Aurora system are the only true exascale machines worldwide. In recent years, GPUs have enabled a step change in processing power—all of the GREEN500 [10] use them. Only Fugaku in the top ten of the current TOP500 list [9] does not use GPUs. Physical constraints and power requirements are forcing computer vendors to develop new strategies to achieve more compute power. Novel hardware systems, including quantum computing, will be required to significantly extend global computational capacity beyond current capabilities.

The idea of using quantum systems to process information more efficiently than classical von Neumann computers was introduced some 40 years ago [11, 12]. Since then there has been steady progress in quantum computing, building on the initial achievements of Shor's algorithm for factorization [13] that could break current methods of encryption, and Grover's algorithm [14, 15] for searching unsorted data. Quantum simulation is another promising field of quantum information processing [16, 17]. Many near-term applications of quantum computers fall under this umbrella. Quantum simulation involves modelling the quantum properties of systems that are directly relevant to understanding modern materials science, condensed matter physics, high-energy physics (HEP), and quantum chemistry.

As the practical relevance of quantum computing becomes clearer, interest has grown beyond the confines of academia and many countries now have national strategies to develop quantum computing and quantum technology more generally, e.g. [18–21]. In June 2024, the United Nations proclaimed 2025 as the International Year of Quantum Science and Technology [22]. The primary goal is to recognize the importance of quantum science and increase public awareness of its future impact.

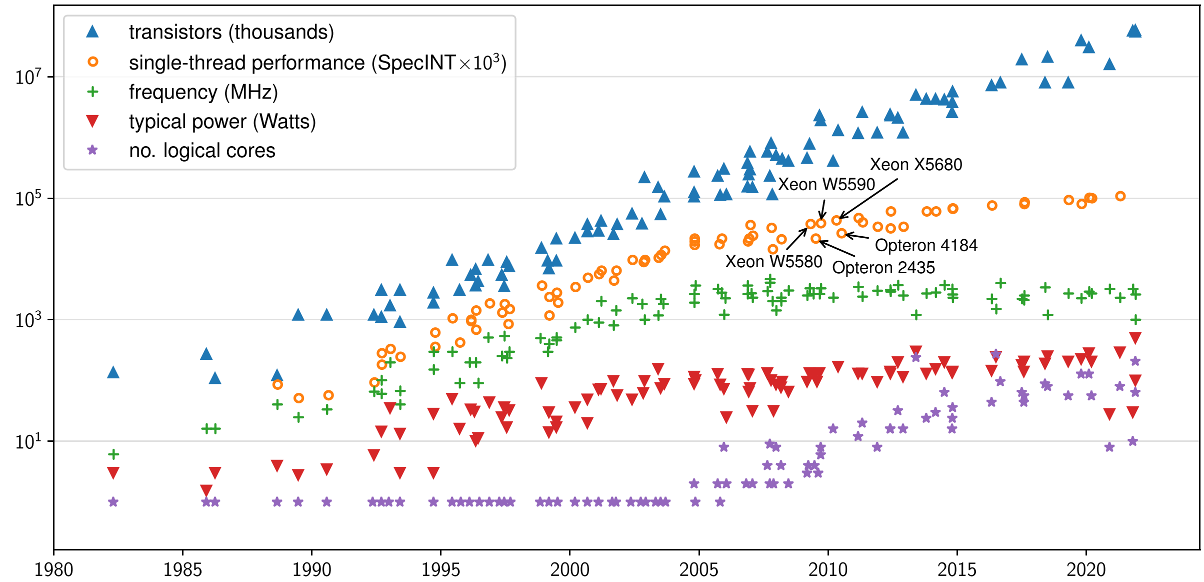


Figure 1. Microprocessor trend data, 1980–2021. Data up to year 2010 collected by M Horowitz, F Labonte, O Shacham, K Olukotun, L Hammond, and C Batten, data for 2010–2021 by K Rupp. Both datasets taken from [8].

The consensus is that quantum devices will not simply replace high performance computing (HPC). It makes little sense (in terms of engineering and economics) to use quantum methods for problems that classical computers already do well. Hence, a hybrid solution is the most efficient, cost-effective and productive way to approach quantum computation. Instead of simply sending jobs to one or more CPUs, hybrid architectures delegate different parts of the problem to different types of processors, the most common currently being GPUs. Conceptually, quantum processor units (QPUs) are a natural extension to enhance the processing power of HPC. However, increasing the number and diversity of processors compounds the challenges for efficient programming and scheduling. Quantum computing hardware is not yet mature enough to seamlessly integrate with HPC (see section 5 for discussion of some of the engineering hurdles).

After several decades of largely academic-based theoretical and experimental development, growing commercial interest has resulted in venture capital [23–25] providing funding for many existing and start-up companies focused on aspects of quantum computing. The current landscape shows growing collaborations between quantum start-ups, established big players, and with well-established financial, pharmaceutical and automotive companies who wish to integrate quantum technologies into their development and production processes (figure 2). Notable players include Google, IBM and Rigetti, who are developing superconducting qubit systems, and Intel who are building systems based on quantum dots, with chips that can be made in its existing foundries. Cloud providers such as Amazon Web Services, are now selling time on different quantum systems, including QuEra, which uses neutral atoms in tweezer arrays, IonQ ion trap quantum processors, and superconducting qubit systems from Rigetti. Current systems are small and imperfect, known as noisy, intermediate scale quantum computers (NISQ) [26, 27]. They are not yet powerful enough to solve useful problems, but do allow researchers to test and develop quantum algorithms and applications to prepare for more advanced quantum hardware.

Several commercial partnerships are focused on integrating quantum processors with HPC. Finnish quantum start-up IQM is mapping quantum applications and algorithms directly to quantum processors to develop application-specific superconducting computers [28]. The result is a quantum system optimized to run particular applications such as HPC workloads. French company Atos and partners Pasqal and IQM are involved in two major quantum hybridization projects in France and Germany [29, 30]: the European HPCQS (Quantum Simulation) project [31] aims to build the first European hybrid HPC with an integrated quantum accelerator, and the German Government's Q-EXA project to integrate the first German quantum computer into HPC [32]. NVIDIA's cuQuantum software development kit (SDK) provides the tools needed to integrate and run quantum simulations in HPC environments using GPUs [33], with the goal of adding QPUs to the SDK. Quantum software vendor Zapata anticipates the convergence of quantum computing, HPC, and machine learning, and has created the Orquestra platform to develop and deploy generative artificial intelligence (AI) applications [34].

Given the growing government and commercial investment in the development of quantum computing, it is timely to consider how it might accelerate computational science and engineering. Now considered the third pillar of scientific research alongside theoretical and experimental science, scientific computing has become ubiquitous in scientific and engineering applications. It leverages the exponential growth in computing power of the last decades, and receives even more government investment than quantum technology, given its essential role in science, engineering, and innovation.

In the UK, HPC investments by the Engineering and Physical Sciences Research Council totaled £466 million over 10 years up to 2019 [35, 36]. This is anticipated to add between £3 billion and £9.1 billion to the UK economy. The

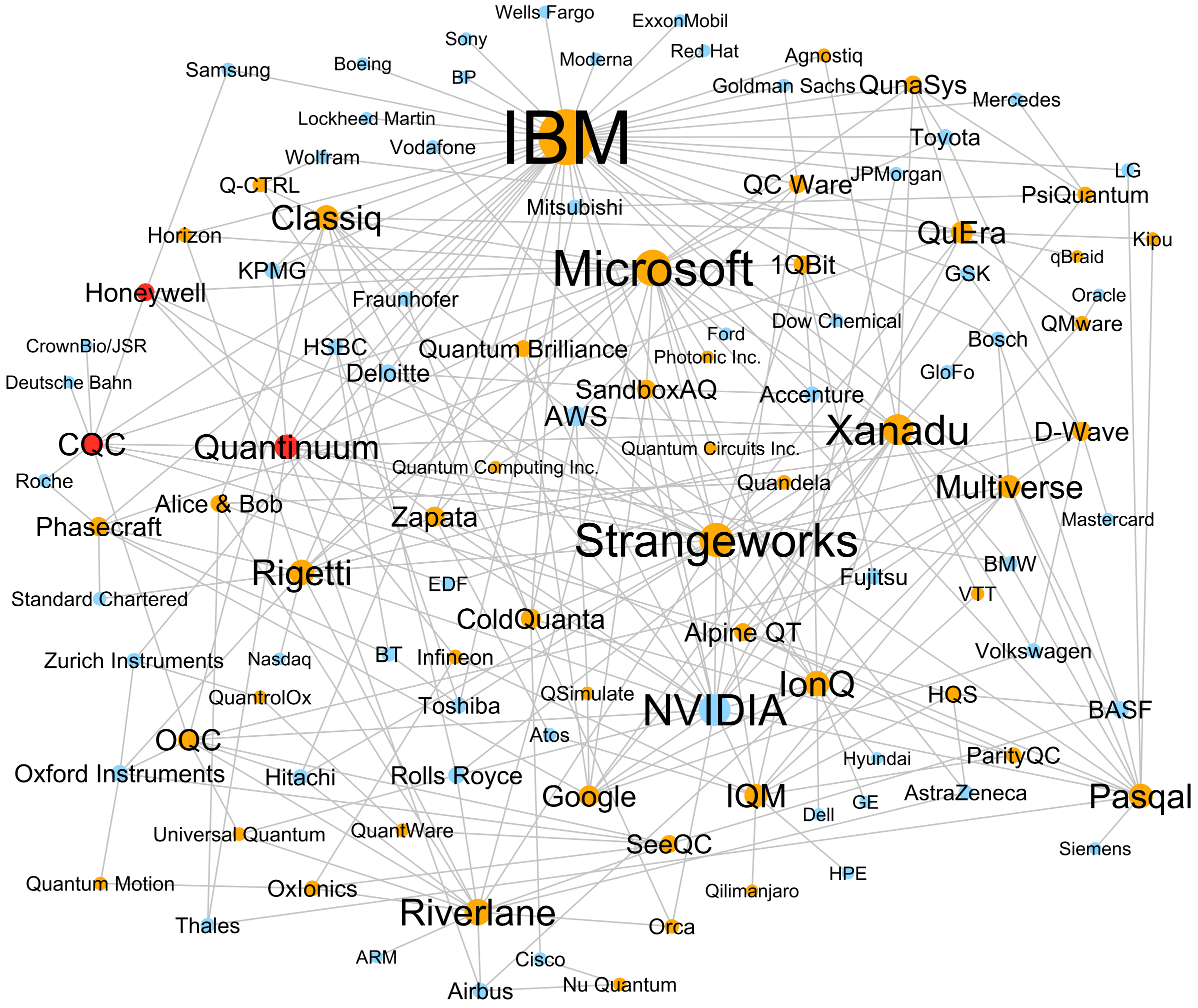


Figure 2. Quantum computing commercial landscape. Some examples of companies (blue) partnered with quantum tech companies (orange) to advance QC research. Cambridge Quantum Computing (CQC) and Honeywell now merged as Quantinuum (red). We include CQC, Honeywell and Quantinuum separately as some collaborations were formed before the merger in Dec 2021. Universities, research institutes and government organizations not shown. Compiled by R Au-Yeung from press releases and publicly disclosed interactions with end users.

European High-Performance Computing Joint Undertaking is a €7 billion joint initiative between the EU, European countries and private partners to advance the HPC ecosystem in Europe [37]. These examples show that governments are willing to invest significant resources into developing computational technologies that can provide numerous societal and economic benefits. Future UK plans include a significant increase in digital research infrastructure across the whole UKRI remit [38] to enable all disciplines to take advantage of computational and data based methods in their research, including quantum computing, when it can offer an advantage.

1.1. Scope of this review

Scientific computing has a broad reach across applications, so this review necessarily has to select which areas to focus on. We choose two important applications: electronic structure calculations as used in computational chemistry, materials, and life sciences; and fluid dynamics simulations, important across all length scales from cells to cosmology. In computational chemistry, the hard problems are fully quantum and therefore natural for quantum computers [11]. In fluid dynamics, the problem is purely classical, the hardness stems from the need to solve nonlinear differential equations. It is less clear that there will be substantial quantum advantages here, although there are multiple interesting proposals for possible quantum algorithms. Our chosen applications make up a significant fraction of current HPC use. Between them, they are diverse enough to offer insights for other areas of computational science and engineering simulations. Although we discuss quantum machine learning (QML) techniques relevant for chemistry and fluid simulations, the wider field of 'big data' and AI is outside the scope of this review. There are indeed potential quantum enhancements in AI, but they merit their own dedicated reviews (see, e.g. [39]).

In this review, we begin by describing the concepts behind quantum computers and giving examples of important foundational quantum algorithms in section 2. For conciseness, we minimize the amount of mathematical detail where possible. Interested readers are encouraged to consult references such as

[40, 41] for details, and refer to the original papers for proofs. There are already many excellent reviews that cover different classes of quantum algorithms (see e.g. [42–44]), including a 'Quantum Algorithm Zoo' website [45] which cites over 400 papers. In this review, we aim to emphasize the physical mechanisms and qualitative insights into quantum algorithms and their real-world applications. We will not discuss abstract topics such as complexity analysis [46, 47] in depth. We also cover some of the alternative quantum computing methods, quantum annealing (section 2.4) for chemistry and materials science simulations (section 3.2.6) and variational quantum algorithms (VQAs) (section 2.5). We outline some of the many proposed performance benchmarks for quantum computers in section 2.7.

Next, we focus on our two example fields: quantum simulation of quantum chemical systems (section 3), and quantum algorithms for classical fluids simulation (section 4). We summarize the current (classical) methods and shortcomings, then explore how quantum computers may improve performance. With the theoretical potential established, we turn to the practical challenges of how to combine quantum components into HPC architectures in section 5. This includes the clock speed mismatches. and how to encode classical data into quantum states. Finally, we summarize and set out future research directions in section 6.

There are other important topics on the road to practical quantum computing that already have their own reviews. These include: quantum error correction (QEC) [48, 49]; verification and testing [50, 51] (we only cover benchmarks in section 2.7); and comparisons between different hardware and software platforms available on the market [23]. Quantum annealing has been well-reviewed recently: Hauke *et al* [52] cover methods and implementations while Yarkoni *et al* [53] focus on industrial applications of optimization problems.

Despite numerous publications hyping the quantum revolution (see, e.g. [1, 54, 55]), it pays to read the fine print [56–58]. The methodology in [59] is an instructive example of how to fairly and objectively evaluate different use cases. This helps to avoid evaluations that focus on the positives and ignore weaknesses. A key message from the IQM-Atos 2021 white paper [60] is that HPC centres must create a long-term strategic plan to successfully integrate quantum computing into their workflow. In particular, they recommend establishing a three-step roadmap:

(i) Gap analysis and quantum solution identification (now). Where are the bottlenecks in classical HPC? How can we improve the solution accuracy?
(ii) Quantum solution design and integration (mid-term). What quantum methods can we use to resolve the issues identified in gap analysis? How do we integrate this into HPC architectures?
(iii) Quantum computing use-case development and implementation (long-term). Which applications and problems can get the most benefit from quantum speedup?

Our review provides an introduction to the tools and topics needed to make such plans.

Table 1. Notation in our paper.

Notation	Description
a, A	Scalar, can be real or complex; greek letters often used for complex scalars
$a(t)$, $A(t)$	Scalar function defined in terms of variable t
$\vec{a}$	Vector, contains elements $\vec{a}_0, \vec{a}_1, \ldots$
$\mathbf{A}$	Matrix
$\lvert a\rangle$, $\lvert A\rangle$	Quantum state vector (with scalar label is usually a basis state)
$\lvert\psi\rangle$	General quantum state vector (with greek letter label)
$\hat{A}$	Quantum operator
j, k	Index in vector or sum
i	Imaginary unit $\sqrt{-1}$

2. Designing quantum algorithms

Developing useful quantum algorithms is extremely challenging, especially before quantum computing hardware is widely available. It is like building a car without having a road or fuel to take it for a test drive. Since quantum computing has a different logical basis from classical computing, it is not as straightforward as taking a successful classical algorithm and mapping it to the quantum domain. For example, a classical CNOT gate makes a copy of a classical bit. In quantum theory, the no-cloning theorem [61] states that it is impossible to make copies of unknown quantum states.

In this section, we outline the basic concepts underpinning quantum computing. Then, we describe some existing quantum algorithms that can potentially be adapted for practical scientific applications. To avoid confusion, we use the notation in table 1 to differentiate between different quantities. These include operators (hats), matrices (bold font), vectors (arrows), scalars and functions (italics), and bra-ket notation for quantum states.

2.1. Universal quantum computers

The most widely used model for quantum computing is the gate or circuit model. The basic unit of quantum information is the quantum bit (qubit), a two-state system. Qubits are the quantum analogue of classical bits. They are two-level quantum systems that represent linear combinations of two basis states

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{1}$$

where $\alpha, \beta \in \mathbb{C}$ and $|\alpha|^2 + |\beta|^2 = 1$. In vector notation, we usually choose

$$|0\rangle = \begin{bmatrix}1\\0\end{bmatrix}, \quad |1\rangle = \begin{bmatrix}0\\1\end{bmatrix}. \tag{2}$$

Multiple qubits can be entangled, forming systems with an exponentially large state space of size $N = 2^n$ for n qubits. This

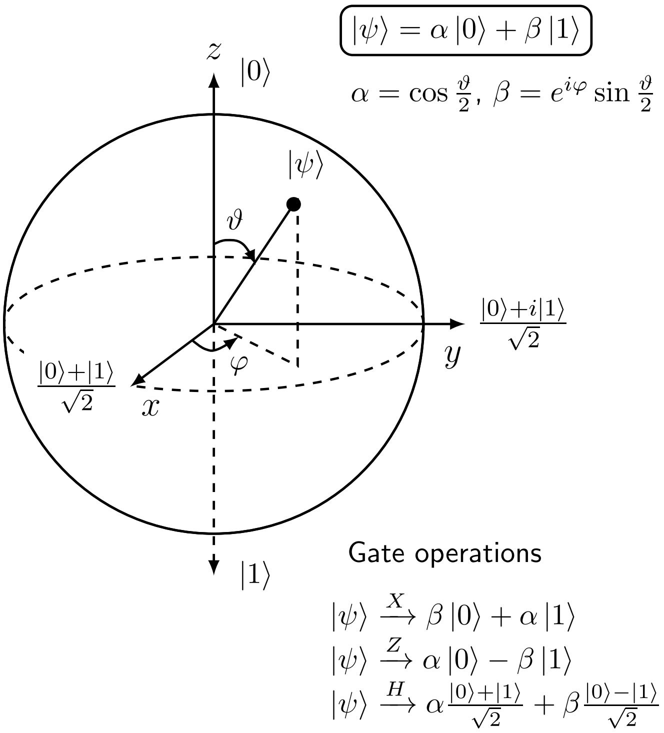

$\boxed{|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle}$
$\alpha = \cos\frac{\vartheta}{2},\ \beta = e^{i\varphi}\sin\frac{\vartheta}{2}$
z
$|0\rangle$
$|\psi\rangle$
ϑ
$\frac{|0\rangle+i|1\rangle}{\sqrt{2}}$
y
$\frac{|0\rangle+|1\rangle}{\sqrt{2}}$
φ
x
$|1\rangle$
Gate operations
$|\psi\rangle \xrightarrow{X} \beta\,|0\rangle + \alpha\,|1\rangle$
$|\psi\rangle \xrightarrow{Z} \alpha\,|0\rangle - \beta\,|1\rangle$
$|\psi\rangle \xrightarrow{H} \alpha\frac{|0\rangle+|1\rangle}{\sqrt{2}} + \beta\frac{|0\rangle-|1\rangle}{\sqrt{2}}$

Figure 3. Bloch sphere representation of qubit state $|\psi\rangle$ with examples of gate operations.

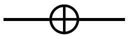
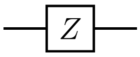
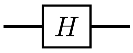
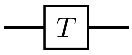
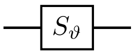
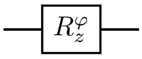
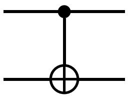
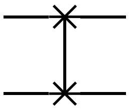
corresponds to the tensor product of matrices, for example,

$$|0\rangle\otimes|1\rangle = \begin{bmatrix}0\\1\\0\\0\end{bmatrix}. \tag{3}$$

The superposition property means that an N-qubit system can represent 2^N states whereas classical bits represent only one state at a time (a single N-bit string). We visualize single qubits using a Bloch sphere (figure 3). To perform calculations, we manipulate the qubit state using (reversible) gate operations. These are analogous to the classical logic gates, such as AND, OR and NOT.

Gate operations are unitary operators applied to the qubits. Applying such an operator $\hat{U}$ to a qubit state, then its conjugate transpose $\hat{U}^\dagger$ brings the qubit back to its original state ($\hat{U}\hat{U}^\dagger = \mathbb{1}$). Gate operations can be written as matrices which act on the state vectors (see examples in table 2). The building blocks of the transformations are 2×2 and 4×4 unitary matrices on single qubits and two qubits respectively. As in classical computing, any computation can be built up from a small set of universal gates. A universal gate set includes at least one entangling gate acting on two or more qubits, and one or more single qubit rotations. The subset of gates that are efficiently classically simulable and thus not sufficient for universal quantum computing are known as Clifford gates, and can be generated by the Hadamard, CNOT and a $\pi/2$ phase gate $S = \sqrt{Z}$. The single qubit Pauli gates (X, Y, and Z) are included in the set of Clifford gates. The addition of a T gate or a Z-rotation with a well-chosen angle is sufficient to generate universal quantum computing. Symbols for diagramatic representation of quantum circuits are also shown in table 2.

Table 2. Examples of commonly used quantum gates. Middle column uses circuit notation where horizontal lines represent qubits. Boxes, crosses and dots indicate the gate applied. Time flows from left to right. Right-hand column shows the matrix representation for acting on computational basis states, (2).

One-qubit gate	Circuit symbol	Matrix
Pauli-X or NOT (bit flip)		$\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}$
Pauli-Z (phase flip)		$\begin{bmatrix}1 & 0\\0 & -1\end{bmatrix}$
Hadamard (create superposition)		$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}$
T gate		$\begin{bmatrix}1 & 0\\0 & e^{i\pi/4}\end{bmatrix}$
Phase shift		$\begin{bmatrix}1 & 0\\0 & e^{i\vartheta}\end{bmatrix}$
Z-rotation		$\begin{bmatrix}e^{-i\varphi/2} & 0\\0 & e^{i\varphi/2}\end{bmatrix}$
Two-qubit gates	**Circuit symbol**	**Matrix**
CNOT (flip target qubit if control qubit is $|1\rangle$)		$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix}$
SWAP (swap two qubit states)		$\begin{bmatrix}1&0&0&0\\0&0&1&0\\0&1&0&0\\0&0&0&1\end{bmatrix}$

2.2. Quantum computing stack

As with classical computers, we can decompose the quantum computer architecture in terms of its various levels of abstraction, called the 'stack'. In gate-based models, this extends from the user interface and compiler down to the low-level gate operations on the physical hardware itself (figure 4). At the highest levels, quantum algorithms give instructions for solving the numerical problem. Compiling and translating quantum gates in the middle levels (software stack) compress the algorithms to accelerate performance, while error-correction techniques mitigate quantum hardware errors. The lower levels comprise the quantum computer's building blocks (hardware) for manipulating and storing qubits [62].

Inside a quantum computer, physical qubits are physical devices that behave as two-level quantum systems. They are usually not perfect, and error correction [48] is designed to combine groups of physical qubits into one or a few logical qubits with lower error rates. This is implemented at a low level in the stack, so that users can work with near perfect logical qubits in a quantum algorithm. Many physical qubits are needed for QEC procedures [49] to produce one logical qubit to perform useful computations. Engineering qubits in

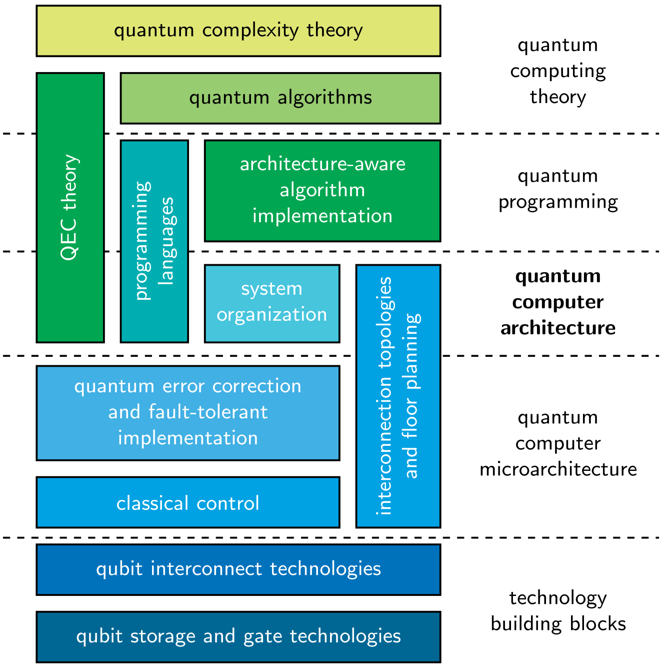

quantum complexity theory
quantum algorithms
QEC theory
programming languages
architecture-aware algorithm implementation
system organization
interconnection topologies and floor planning
quantum error correction and fault-tolerant implementation
classical control
qubit interconnect technologies
qubit storage and gate technologies
quantum computing theory
quantum programming
quantum computer architecture
quantum computer microarchitecture
technology building blocks

Figure 4. Quantum computer architecture among some sub-fields of quantum computation. Reproduced with permission from [62]. Copyright © 2013 Owner/Author.

real physical hardware is highly challenging. The requirements for scalable quantum computing laid out by DiVincenzo [63] are still relevant today: scalable, well-characterized qubits; the ability to initialize the qubits; long decoherence times; a 'universal' set of quantum gates; and the ability to measure the qubits.

For example, Reiher *et al* [64] examine the cost of performing quantum simulations to find the potential energy landscape of a quantum chemical structure (FeMoco, more details in section 3.3.2). They estimated 313 physical qubits are needed for each logical qubit, if there is one error for every 10^9 quantum gates. By extension, we would likely need thousands of physical qubits per logical qubit, or much better error rates, for complicated real-world quantum chemistry simulations. In the rest of our review, we will discuss in terms of logical qubits unless otherwise stated.

2.3. Foundational quantum algorithms

In this section, we introduce some foundational quantum algorithms which can be adapted or used as components in quantum algorithms for a wide variety of applications. They also illustrate how quantum algorithms can provide powerful speed ups over classical algorithms for the same problems. For readers seeking a comprehensive graduate-level introduction, we recommend [40, 41, 65] plus IBM's Qiskit software documentation [66].

2.3.1. Quantum Fourier transform (QFT). The QFT is a key subroutine in many quantum algorithms, most prominently Shor's factoring algorithm [13]. The QFT operation changes the quantum state from the computational (Z) basis to the Fourier basis. It is the quantum analogue of the discrete Fourier transform and has exponential speed-up (complexity $O((\log N)^2)$) compared to the fast Fourier transform's $O(N\log N)$ complexity for problem size N. The QFT performs a discrete Fourier transform on a list of complex numbers encoded as the amplitudes of a quantum state vector. It stores the result 'in place' as amplitudes of the updated quantum state vector. Measurements performed on the quantum state identify individual Fourier components—the QFT is not directly useful for determining the Fourier-transformed coefficients of the original list of numbers, since these are stored as amplitudes (see section 5.2 for more details on quantum encoding).

When applying the QFT to an arbitrary multi-qubit input state, we express the operation as

$$\hat{U}_{\mathrm{QFT}} = \frac{1}{\sqrt{N}} \sum_{j,k=0}^{N-1} e^{2\pi ijk/N} |j\rangle\langle k| \tag{4}$$

which acts on a quantum state in N-dimensional Hilbert space $|x\rangle = \sum_{k=0}^{N-1} x_k |k\rangle$ and maps it to $|y\rangle = \sum_{j=0}^{N-1} y_j |j\rangle$. This is a unitary operation on n qubits that can also be expressed as the complex $N\times N$ matrix

$$\mathbf{F}_N = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \dots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \omega^6 & \dots & \omega^{2(N-1)} \\ 1 & \omega^3 & \omega^6 & \omega^9 & \dots & \omega^{3(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \omega^{3(N-1)} & \dots & \omega^{(N-1)^2} \end{bmatrix} \tag{5}$$

where $\omega = e^{2\pi i/N}$ and $N = 2^n$. Note that the Hadamard gate (table 2) acts as the single-qubit QFT. It transforms between the Z-basis states $|0\rangle$ and $|1\rangle$ to the X-basis states

$$|\pm\rangle = \frac{1}{\sqrt{2}} (|0\rangle \pm |1\rangle) . \tag{6}$$

Figure 5(a) shows the quantum circuit for the QFT operation. Each multi-qubit gate block can be decomposed into a sequence of single or two-qubit gates. For example, the $\mathbf{B}_{k+1}$ block is written in terms of controlled operations involving two qubits (figure 5(b)). The $\mathbf{B}_{k+1}$ gate is applied to the last $k+1$ qubits. All controlled-R gates are diagonal matrices and commute,

$$\mathbf{R}_N = \begin{bmatrix} 1 & 0 \\ 0 & e^{-2\pi i/N} \end{bmatrix} . \tag{7}$$

So the order in which they are applied does not change the outcome. The bit-reversal permutation matrix $\mathbf{P}_n$ can first be written in terms of $\lfloor n/2 \rfloor$ swap gates (figure 5(c)). Since each swap can be implemented with three CNOT gates, $\mathbf{P}_n$ requires $\lfloor 3n/2 \rfloor$ CNOT gates. Hence the QFT requires $O(n^2)$ elementary gates in total.

The semi-classical nature of the QFT circuit means it can be efficiently simulated on classical computers [68, 69] when

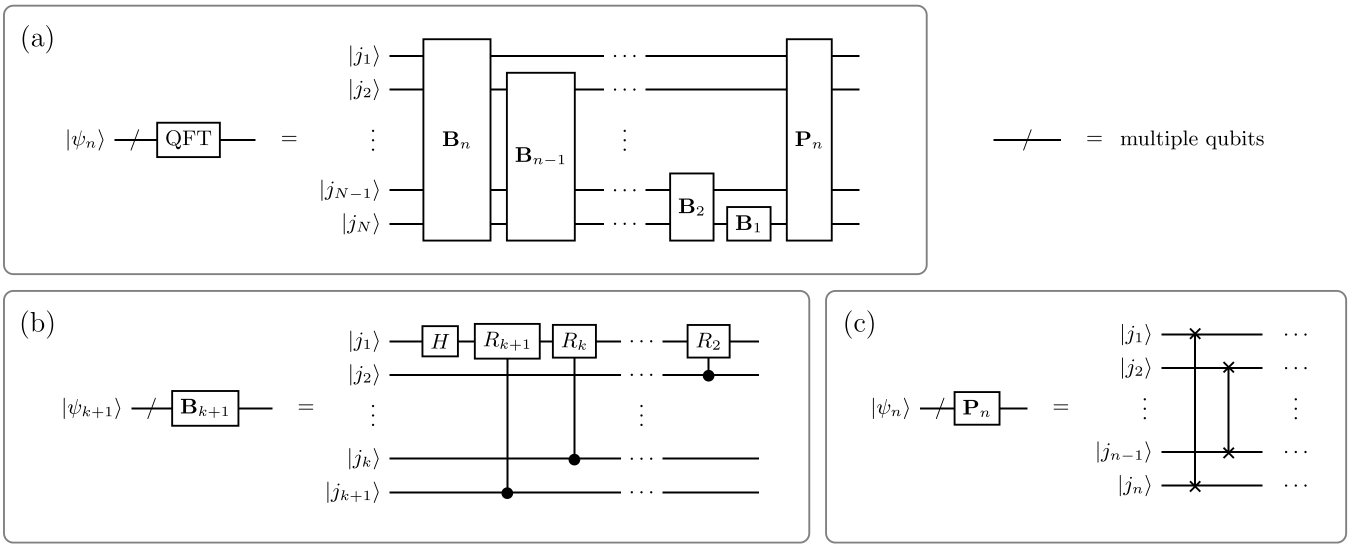


Figure 5. Quantum Fourier transform operation on n qubits. QFT $= \mathbf{F}_N$, (5). [67] John Wiley & Sons. © 2020 John Wiley & Sons Ltd.

used at either the start or end of a quantum computation. The idea is to measure a qubit, then use the result to produce a classical signal which controls a one-qubit transformation on the next qubit before it is measured, and so forth.

2.3.2. Quantum phase estimation (QPE). Kitaev's QPE algorithm [70] is a core component of many quantum algorithms and an important technique in algorithm design. While the standard procedure is based on the QFT circuit, various improvements can reduce the dependence on QFT while retaining its accuracy by using, for example, approximations to relax the QFT constraints [71] or a modified algorithm from quantum metrology [72]. We outline the QPE algorithm below.

The basic phase estimation problem is, given a unitary operator

$$\hat{U}|\psi\rangle = \lambda|\psi\rangle. \tag{8}$$

We want to find an eigenvalue λ of the eigenvector $|\psi\rangle$. Because $\hat{U}$ is unitary, the eigenvalue can be expressed as $\lambda = e^{2\pi i\theta}$ and the real aim is to estimate the eigenvalue phase θ.

The phase estimation algorithm contains two main steps (figure 6). In the circuit, the top register contains n 'counting' qubits in state $|0\rangle$. The second register contains as many qubits as needed to store $|\psi\rangle$, the eigenstate of $\hat{U}$. We apply a set of Hadamard gates to the first register (this is equivalent to a QFT on the all zeros state) and controlled-$\hat{U}$ operations on the second register with $\hat{U}$ raised to successive powers of two. The final state of the first register is

$$\frac{1}{\sqrt{2^n}}\sum_{k=0}^{2^n-1} e^{2k\pi i\theta}|k\rangle. \tag{9}$$

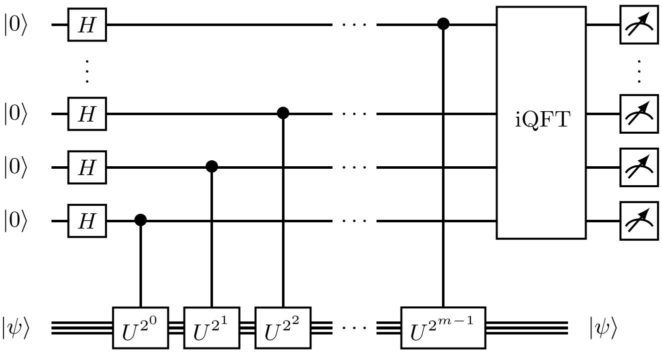


Figure 6. Structure of quantum phase estimation circuit, where iQFT is the inverse of the QFT operation.

Applying the inverse QFT (iQFT) gives

$$\frac{1}{\sqrt{2^n}}\sum_{k=0}^{2^n-1} e^{2k\pi i\theta}|k\rangle|\psi\rangle \to |\tilde{\theta}\rangle|\psi\rangle \tag{10}$$

where $\tilde{\theta}$ is an estimate for θ. The final stage is to read out the state of the first register by measuring in the computational basis. Hence, we estimate θ to n bits of precision.

QPE is equivalent to simulating quantum system dynamics. Given an eigenstate $|\psi\rangle$, we can estimate the eigenvalue for Hamiltonian $\hat{H}$ by applying $\hat{H}$ via the time-evolution operator $\hat{U} = e^{-i\hat{H}t}$. When the algorithm implements $\hat{U}^k$ for increasing powers k, this corresponds to evolving the system for increasing times kt [73]. The phases θ are eigenvalues of time evolution operator $\hat{U}$ which we can map back on to the eigenenergies E of $\hat{H}$ [73],

$$\hat{U}|\psi\rangle = e^{i\hat{H}t}|\psi\rangle = e^{2\pi i\theta}|\psi\rangle, \tag{11}$$

$$E = 2\pi\theta/t. \tag{12}$$

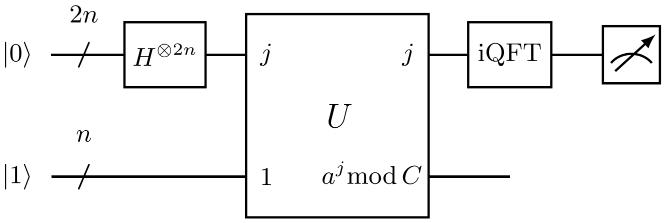


Figure 7. Quantum circuit schematics for Shor's algorithm.

Computational chemistry often requires finding the eigenenergies, as they can help deduce many chemical properties like ionization potential and equilibrium constants. We discuss some of these applications in section 3.

The phase estimation circuit outputs an eigenphase as the measurement outcomes of the top register. It also prepares the corresponding eigenstate $|\psi\rangle$ in the lower register. This is a useful starting point for calculating other observables besides energy. Once we have the eigenstates and eigenenergies of interest, we can understand how a (closed) system evolves in time by decomposing the initial state into a sum of eigenstates and evolving each eigenstate according to the phase found in QPE. The final state is an interferometric sum of all components.

There are two important performance metrics for the QPE: maximum run time, and total run time over all repetitions from each circuit in the algorithm. Maximum and total run time approximately measure the circuit depth and total cost of the algorithm respectively. The QPE algorithm in figure 6 is unsuitable for early fault-tolerant quantum computers since these devices can only implement low circuit depths and limited numbers of logical qubits [72].

2.3.3. Shor's algorithm.

While Feynman's 1982 arguments [11] formalized the idea of using quantum computers to solve quantum problems, what really caught people's attention was Shor's prime factoring algorithm in 1994 [13, 74]. It provided a way for large enough (future) quantum computers to break RSA encryption [75], then considered unbreakable due to the exponential computational cost of factoring large numbers. RSA encryption works by multiplying two large prime numbers p and q. The product $C = pq$ is so large that the computational cost of factorizing—and hence recovering the encrypted message—is impractical. Shor's algorithm showed that factoring could be solved by quantum computers in polynomial time.

Shor's factoring algorithm reduces the factorization problem into an order-finding problem. Given the modular exponential function

$$f(x) = a^x \bmod C,\ 0 < a < C \tag{13}$$

where a is a random number coprime to C, the order-finding problem involves finding the period r of function $f(x)$. For an even r and if $a^{r/2} \neq -1 \bmod C$, at least one prime factor of C is given by $\gcd(a^{r/2} \pm 1, C)$.

There are two main components of Shor's algorithm: modular exponentiation (calculating $a^x \bmod C$) and iQFT. Figure 7 shows a high-level circuit diagram of the algorithm. First we initialize the registers: we use $n = \lceil \log_2(C+1) \rceil$ qubits to store the number C. To extract the period r of $f(x)$, we need one register initialized as $|0\rangle$ and containing $2n$ qubits to store values x, and another register initialized as $|1\rangle$ with n qubits to store $f(x)$.

Applying the Hadamard gate on the qubits in $|0\rangle$ produces a superposition state of all integers, such that $|0\rangle \rightarrow \sum_{j=0}^{2^n-1} |j\rangle / \sqrt{2^n}$. Then the unitary $\hat{U}$ associates each input j with the value $a^j \bmod C$ (13), for some random a which has no common factors with C, to get

$$\frac{1}{\sqrt{2^n}} \sum_{j=1}^{2^n-1} |j\rangle |a^j \bmod C\rangle. \tag{14}$$

The first register is in a superposition of 2^n terms $|j\rangle$ and the circuit calculates the modular exponentiation for 2^n values of j in parallel. Next, iQFT on the first register gives

$$\frac{1}{2^n} \sum_{k=0}^{2^n-1} \sum_{j=0}^{2^n-1} e^{2\pi i jk/2^n} |k\rangle |a^j \bmod C\rangle, \tag{15}$$

where interference causes only k terms with integer multiples of $2^n/r$ to have a substantial amplitude. This is analogous to the QPE in (9) and (10). Finally, measuring the first register gives the period r. Ideally, the measurement outcome is an integer multiple of $2^n/r$ with high probability. We can then deduce r classically, with continued fractions.

Note that Kitaev's QPE algorithm [70] gives an alternate derivation of Shor's algorithm. It reduces the order-finding problem to one involving phase estimation by essentially replacing the unitary $\hat{U}$ (figure 7) with a QPE subroutine.

Classical computers have to date factored semi-primes up to 829 bits long [76]. Each extra bit doubles the classical computational cost. However, quantum computers large enough to beat this are still some way into the future. This is due to the error correction overheads required to carry out the quantum computation accurately enough [77].

2.3.4. Grover's algorithm.

Grover's algorithm [14, 15] is a technique for searching unstructured data. The canonical example is finding the name corresponding to a given phone number. It can be used as a building block in other applications, such as database search and constraint satisfaction problems [78]. Suppose there are N items labelled by bit strings $x = 0$ to $N-1$, and there is a unique element x_0 such that $f(x) = 1$ if and only if $x = x_0$. The quantum algorithm uses n-qubit basis states $|x\rangle$ corresponding to the labels of the items, with $n = \lceil \log_2 N \rceil$.

Grover's algorithm uses an oracle, which in computer science is a mathematical device for providing part of the computation that is not included in the analysis. Consequently, Grover's algorithm is only of practical use in situations where we can perform the oracle function efficiently, i.e. $f(x)$ can be evaluated for a superposition of all possible inputs $|x\rangle$. This can happen when the algorithm is used as a subroutine, but is

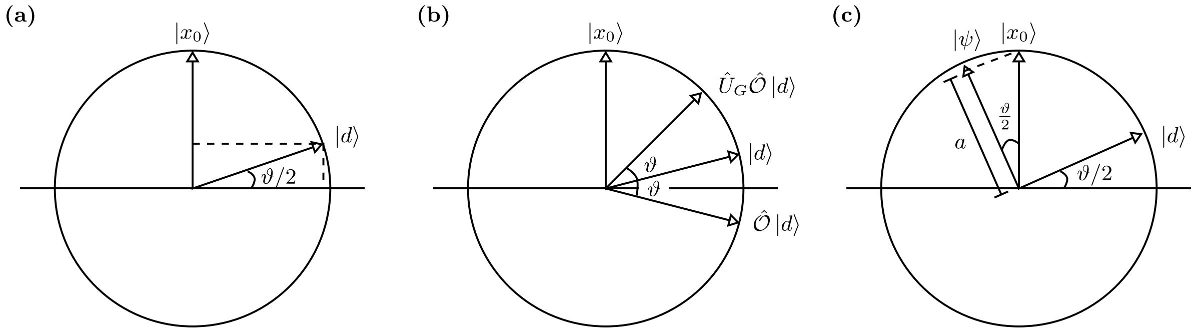


Figure 8. Grover's algorithm. (a) Initial state vector $|d\rangle$ with angle $\vartheta/2$. (b) Vector $|d\rangle$ after inversion and reflection. (c) Final state vector $|\psi\rangle$.

not generally possible when the data are classical. The oracle $\hat{O} = (-1)^{f(x)}$ has the effect

$$\hat{O}|x\rangle = \begin{cases} -|x_0\rangle, & x = x_0 \\ |x\rangle, & \text{otherwise.} \end{cases} \tag{16}$$

Figure 8 illustrates in two dimensions the amplitude amplification part of Grover's algorithm, through inversion about the mean. The initial state

$$|d\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle \tag{17}$$

is an equal superposition of all $|x\rangle$. For large N, note that $|d\rangle$ is almost orthogonal to the marked state $|x_0\rangle$. In figure 8(a), the small angle $\vartheta/2 \simeq 1/\sqrt{N}$. The oracle $\hat{O}$ acts on only $|x_0\rangle$ by inverting the sign of its amplitude. Geometrically, $\hat{O}$ reflects vector $|d\rangle$ about the horizontal axis. Then applying the Grover operator

$$\hat{U}_\mathrm{G} = 2|d\rangle\langle d| - \mathbb{1} \tag{18}$$

reflects $\hat{O}|d\rangle$ about the axis defined by $|d\rangle$. Figure 8(b) shows the action of $\hat{O}$ and $\hat{U}_\mathrm{G}$. Since ϑ is a small angle, this achievement is modest. Solving for $r\vartheta = \pi/2$ gives an approximate number of required iterations,

$$r = \left\lfloor \frac{\pi}{2\vartheta} \right\rfloor = \left\lfloor \frac{\pi}{4}\sqrt{N} \right\rfloor \tag{19}$$

to reach the final state within $\vartheta/2$ of $|x_0\rangle$. The probability of measuring $|x_0\rangle$ is then better than $1 - 1/N$. Since the number of iterations r in (19) scales as $\sqrt{N}$, Grover's algorithm generally provides a square root speed up over a classical search, which would need to check on average $N/2$ items to find x_0. There are alternative quantum algorithms that solve the search problem, such as quantum walks [79] (section 2.3.6). These are easily generalized to find multiple items [80] and are intuitive for graph-based search problems.

2.3.5. *Quantum amplitude amplification (QAA).*

Grover's algorithm was subsequently generalized to the framework of QAA [81]. We can understand QAA as an iterative process that starts with a uniform superposition of all states. Then the algorithm uses a generalization of the Grover operator to increase the probability amplitude of some target state while reducing all other probability amplitudes at each iteration. Amplitude amplification is a valuable subroutine that appears in many quantum algorithms. In the Harrow–Hassidim–Lloyd (HHL) algorithm for solving linear systems (section 2.3.7), the variable time QAA algorithm reduces the number of repetitions needed to obtain the correct solution, which reduces the algorithm run-time [82]. QAA also appears in applications that involve optimization [83, 84], determining graph connectivity [85], pattern matching [86], quantum counting [87], and crypto-key search [88].

2.3.6. *Quantum walks.*

In the last few decades, classical random walks and Markov chains have generated powerful new algorithms in computer science and mathematics [89]. They were thus a natural place to look for quantum equivalents. Early in the development of quantum algorithms both discrete-time [90, 91] and continuous-time quantum walks [92] were introduced with algorithmic applications in mind. The first proven speed ups for quantum walk algorithms came soon after, with a search algorithm [79] equivalent to Grover's algorithm [14] providing a quadratic speed up, and an algorithm for transport across a particular type of disordered graph [93] that provides an exponential speed up with respect to an oracle. There are many comprehensive reviews of quantum walks and their applications, including [94, 95].

Quantum walks are widely studied for their mathematical properties, and for providing simple models of physical phenomena. In an algorithmic context, they provide quantum speed up for sampling problems. The quantum walk dynamics provide both faster spreading [91] and faster mixing [90], but also localization. These are exploited in the search algorithm [79]. The fast mixing dynamics can also be exploited in a quantum annealing (section 2.4) context, see, for example, [96].

Quantum walks also provide models of universal quantum computing [97, 98], but they are not suitable for physical implementation. Instead these are mostly useful for complexity proofs. Quantum walks with multiple non-interacting walkers and quantum particle statistics include the boson sampling model [99], a computational model intermediate between classical and quantum computing. Quantum walks with multiple

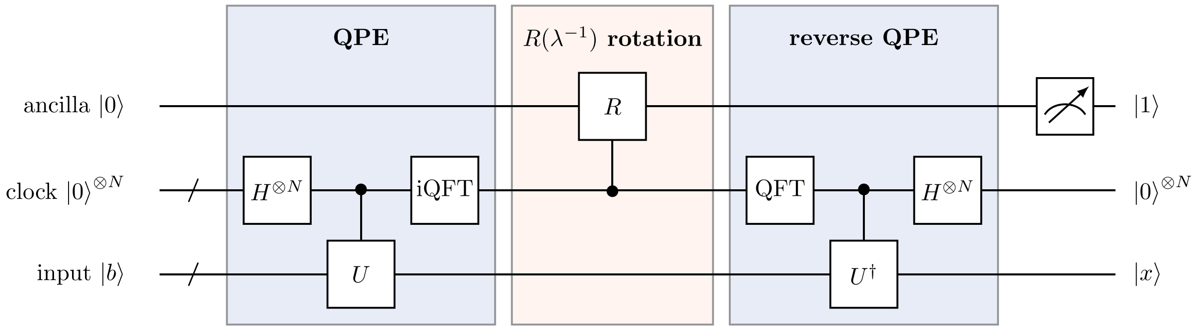


Figure 9. High level HHL circuit. Reproduced with permission from [104].

interacting walkers [100] are a full universal model of quantum computing. They are a special case of quantum cellular automata (see, for example, [101]) and can provide an alternative physical architecture for quantum computing, most suitable for neutral atoms in optical lattices [102].

2.3.7. Harrow-Hassidim-Lloyd (HHL) algorithm. In 2009, Harrow, Hassadim and Lloyd [103] presented their eponymous algorithm to solve one of the most basic classical problems in scientific computing. The HHL algorithm efficiently calculates the solution $\vec{x} = \mathbf{A}^{-1}\vec{b}$ of the linear system $\mathbf{A}\vec{x} = \vec{b}$ for a sparse, regular $N \times N$ Hermitian matrix $\mathbf{A}$. The hard part of this problem is to invert $\mathbf{A}$ and obtain $\mathbf{A}^{-1}$. To see how a quantum method might help, we present a high level description of the HHL algorithm. For further details and analysis, see [41, 104].

We start with the ancilla, clock, and input registers (figure 9). The HHL algorithm represents $\vec{b}$ as a quantum state in the computational basis

$$|\vec{b}\rangle = \sum_j b_j |j\rangle. \tag{20}$$

And stores it in the input register with the amplitude encoding method (section 5.2). Then using Hamiltonian simulation, we apply

$$\hat{U} = e^{i\hat{A}t} \tag{21}$$

to $|b\rangle$ for a superposition of times t determined by the clock register. This step corresponds to QPE which expresses $|b\rangle$ in terms of the eigenbasis $|u_j\rangle$ of $\mathbf{A}$ and finds the corresponding eigenvalues λ_j which are stored in the clock register. Now we can express the system state as

$$\sum_j \beta_j |u_j\rangle |\lambda_j\rangle \tag{22}$$

with $\beta_j = \langle u_j|\vec{b}\rangle$ and

$$|\vec{b}\rangle = \sum_j \beta_j |u_j\rangle. \tag{23}$$

Note that we do not know what the eigenstates $|u_j\rangle$ are.

In the next step, we perform a controlled rotation on the ancilla register with the clock register as the control. The procedure extracts the eigenvalues of $\mathbf{A}^{-1}$ by performing a σ_y rotation, conditioned on λ_j. It is equivalent to the linear map, $|\lambda_j\rangle \rightarrow \lambda_j^{-1}|\lambda_j\rangle$. This transforms the system to

$$\sum_j \beta_j |u_j\rangle |\lambda_j\rangle \left(\sqrt{1 - \frac{c^2}{\lambda_j^2}} |0\rangle + \frac{c}{\lambda_j} |1\rangle \right) \tag{24}$$

with normalization constant c.

The final step uses inverse QPE to uncompute the $|\lambda_j\rangle$ contained in the clock register. This leaves the remaining state as

$$\sum_j \beta_j |u_j\rangle |0\rangle \left(\sqrt{1 - \frac{c^2}{\lambda_j^2}} |0\rangle + \frac{c}{\lambda_j} |1\rangle \right). \tag{25}$$

Measuring and post-selecting on the ancilla qubit being $|1\rangle$ outputs the state

$$|x\rangle = A^{-1}|b\rangle \approx c \sum_j \frac{\beta_j}{\lambda_j} |u_j\rangle, \tag{26}$$

so we have the solution amplitude encoded in the input register as a quantum state. If the post-selection is unsuccessful and the ancilla measurement outputs $|0\rangle$, we have to rerun the algorithm.

There are many limitations of the HHL algorithm [58, 104]. Key examples include the requirement to be able to efficiently prepare the state $|b\rangle$ by somehow loading the vector components into the quantum computer. This may be easy if the state is an output from another quantum subroutine but in general this is not trivial. Additionally it must be possible to efficiently implement unitaries of the form $e^{i\hat{A}t}$ to carry out QPE. Moreover, the matrix $\mathbf{A}$ must be Hermitian with unit determinant, although this can be somewhat relaxed by virtue of the fact that any non-Hermitian matrix can be padded into Hermitian form and the determinant can be re-scaled. However, even if these requirements are met, it is not possible to efficiently recover the full solution $|x\rangle$. Instead, we may carry out further processing to compute the expectation value of some operator $\hat{M}$, such as $\langle x|\hat{M}|x\rangle$.

The importance of HHL for practical applications has led to many improvements on the original algorithm, such as using time variable amplitude amplification to increase the success probability during ancilla measurement [82]. This is a more general case of amplitude amplification (section 2.3.5), better suited to quantum algorithms where individual branches finish at different times. Individual components, such as Hamiltonian simulation in the QPE step (21), have also been improved. QPE remains a bottleneck by requiring $O(1/\epsilon)$ applications of a unitary operation to estimate its eigenvalues to precision ϵ. Hence, even if we use the fastest Hamiltonian simulation methods, the overall improvement would likely be modest with total complexity still being polynomial in $(1/\epsilon)$. One remedy involves directly applying the matrix inverse to bypass the phase estimation subroutine. This can be achieved by decomposing the inverse operator into a linear combination of unitaries (LCU) that are easier to apply, using Fourier or Chebyshev expansions [105]. This approach exponentially improves dependence on the precision parameter and has subsequently found good use in applications ranging from accelerating finite element methods (FEMs) [106] to approximating the hitting times of Markov chains [107].

Alternatively, HHL can be 'hybridized' into a quantum–classical method. Given the resource limitations of NISQ-era hardware, hybrid approaches will likely pave the way for meaningful applications in the near future. For example, an iterative-HHL approach [108] contains a classical iterative process to improve accuracy beyond the limit imposed by number of qubits in QPE. However, the convergence rate for more general cases remains uncertain. Other approaches include streamlining the algorithm, such as using a classical information feed forward step after the initial QPE to reduce the circuit depth for subsequent steps in HHL [109]. Of course, if the reduced circuit is not applicable, it would then require repeating QPE to determine an appropriate reduction. A hybrid approach can achieve comparable precision to conventional HHL with fewer qubits and multiple phase estimation modules [110]. It can also improve phase estimation itself to mitigate the long coherence times [111].

Solving dense matrices, particularly relevant in machine learning and kernel methods [112], has been tackled by using quantum singular value estimation to gain a polynomial advantage over traditional HHL approaches [113, 114]. There are many QML algorithms where HHL is an important subroutine, such as data classification using quantum support vector machines [115], quantum principal component analysis [116], or quantum linear discriminant analysis [117]; quantum ordinary linear regression (QOLR) [118, 119] and QOLR for prediction [120]; quantum ridge regression [121]; quantum recommendation systems [122]; quantum singular value thresholding [123]; and quantum Hopfield neural networks [124]. See section 2.5.4 for more on quantum neural networks (QNN).

In a classical context, it is usually the complete solution state that is the desired output from solving a system of linear equations. This could be the velocity at each point in a fluid simulation for example. Thus, it is not immediately clear how recovering the expectation value of some operator such as $\langle x|\hat{M}|x\rangle$ could be meaningful. Moreover, extracting the individual components of $|x\rangle$ is a costly process that can severely mitigate the inherent speedup offered by the HHL algorithm [58]. Thus, while HHL has proven useful as a subroutine for other quantum algorithms, direct application to classical problems requires detailed consideration of the bottlenecks associated with data transfer between classical and quantum realms [125] (see section 5). Although for some configurations it is possible to efficiently prepare the quantum state by exploiting an underlying functional form [126], this is not always possible and usually requires some knowledge of the problem *a priori*.

2.4. Quantum optimization

Combinatorial optimization problems are widely encountered across industry [53], academia [52], and medical science [127]. Typically, the problem is to find the best of an exponentially large set of solutions. Mathematically, a basic optimization problem consists of the objective (cost) function which is the output we want to maximize or minimize, the input variables, and optional constraints. Many optimization problems can be expressed as quadratic unconstrained binary optimization (QUBO) problems. These are encoded using an upper-triangular $N\times N$ real matrix $\mathbf{Q}$ and vector of binary variables $\vec{x}$. The aim is to minimize the cost function

$$f(\mathbf{x}) = \sum_j \mathbf{Q}_{jj}\vec{x}_j + \sum_{j<k}\mathbf{Q}_{jk}\vec{x}_j\vec{x}_k. \tag{27}$$

Many cost functions contain a large number of false local minima, making it difficult for classical algorithms to find the true global minimum.

Classical simulated annealing [128] is based on the idea that thermal fluctuations move the system out of local minima and toward the lowest potential energy state as temperature decreases. The simulated annealing algorithm simulates a random walker that travels through the search space or optimization landscape. The rate at which the temperature is reduced determines how fast the system evolves and how likely it is to avoid becoming stuck in local minima. Quantum annealing [129, 130] was conceived as an alternative that adds quantum tunnelling to the system evolution, allowing it to escape from local minima more easily. Figure 10 illustrates the quantum annealing process.

Conveniently, classical optimization problems can be efficiently mapped to finding the ground state of a classical Ising Hamiltonian [132, 133]. This leads naturally to a quantum version using the transverse field Ising Hamiltonian,

$$\hat{H}_{\text{Ising}} = \underbrace{A(t)\sum_j \hat{\sigma}^x_j}_{\hat{H}_0} + \underbrace{B(t)\sum_{j,k}\left(h_j\hat{\sigma}^z_j + J_{jk}\hat{\sigma}^z_j\hat{\sigma}^z_k\right)}_{\hat{H}_\text{p}} \tag{28}$$

with the Pauli x- and z-matrices $\hat{\sigma}^{x,z}$, symmetric interaction strength $J_{jk} = J_{kj}$ of qubit spins q_j and q_k, and on-site energy h_j. Note that the qubit spin basis states are $|\pm 1\rangle$. There is a straightforward translation from QUBO $\vec{x}$ to Ising $\vec{q}$, using $\vec{x}_j = (\vec{q}_j + 1)/2$. The resulting values of the $\{h_j, J_{jk}\}$ variables then encode the problem into the Ising Hamiltonian $\hat{H}_\text{p}$.

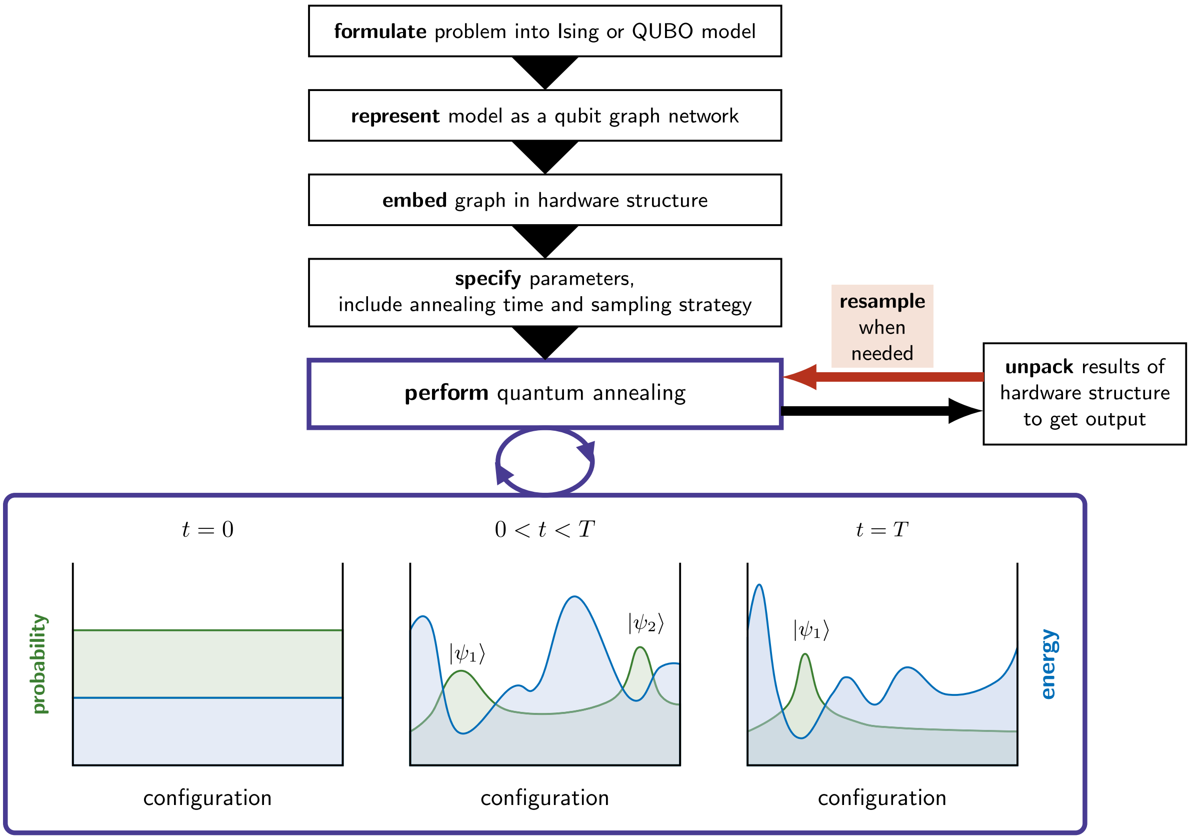

formulate problem into Ising or QUBO model
represent model as a qubit graph network
embed graph in hardware structure
specify parameters,
include annealing time and sampling strategy
perform quantum annealing
resample when needed
unpack results of hardware structure to get output
$t = 0$
$0 < t < T$
$t = T$
probability
energy
$|\psi_1\rangle$
$|\psi_2\rangle$
configuration
configuration
configuration

Figure 10. The full quantum annealing process from problem formulation to solution. The graphs illustrate how the initially uniform distribution evolves under the driver Hamiltonian to concentrate in the global minimum representing the solution. Reproduced from [131]. CC BY 4.0.

The transverse field in the x-direction $\hat{H}_0$ provides the dynamics that rotate the qubits from their initial state to the final target state, ideally the ground state of problem Hamiltonian $\hat{H}_\mathrm{p}$. The control functions $A(t)$ and $B(t)$ must be specified. Usually $A(t)$ varies from 1 to 0, and $B(t)$ from 0 to 1. The choice of how they vary determines the method used to find the ground state and hence the problem solution.

Adiabatic quantum computing (AQC), introduced by Farhi *et al* [134] in 2001, uses the quantum adiabatic theorem to guarantee that the system remains in the ground state. It assumes that the control functions $A(t)$ and $B(t)$ are varied slowly enough to keep the system in the instantaneous ground state. A useful property of AQC is that it is inherently robust against noise [135]. When used with more general Hamiltonians than in (28), it is computationally equivalent to gate-based quantum computers [136], making it a universal quantum computing paradigm [137].

While AQC provides a sound theoretical underpinning for quantum annealing, in reality, the adiabatic conditions imposed by AQC are rarely met. Meeting them would require long run times that are inefficient and impractical. At the other extreme, continuous-time quantum walk algorithms [92] (section 2.3.6) can locate ground states for certain problems [96] where it is viable to make many short repeats. Further diabatic methods, which use other mechanisms with shorter run times, are reviewed in [138]. Many of these, including the quantum approximate optimization algorithm (QAOA) or quantum alternating operator ansatz [139, 140] use classical optimization in the controls to produce an efficient quantum process. We describe QAOA in section 2.5.1 after introducing VQAs more generally. Exploiting the non-adiabatic and open system effects in quantum annealing is the topic of many recent efforts to derive efficient controls [141, 142]. For more details, Albash and Lidar [43] provide a comprehensive introduction to AQC and quantum annealing. Hardware currently available for quantum annealing includes superconducting systems, trapped ions, and Rydberg atoms (see [52] for an overview).

2.5. Variational quantum algorithms (VQAs)

VQAs distribute the job of solving a problem between a parameterized quantum circuit (PQC) and a conventional classical optimizer. Figure 11 shows schematically how it works; for a detailed recent review, see Cerezo *et al* [143]. The quantum circuit is defined by a set of parameters $\vec{\theta}$ that determine the quantum gates. We evaluate a classical cost function $C(\vec{\theta})$ that can be calculated from the measured outputs of the quantum circuit. Classical optimization then updates the parameters $\vec{\theta}$, guided by the cost function. Mathematically, the aim is to minimize a linear cost function which typically takes the form

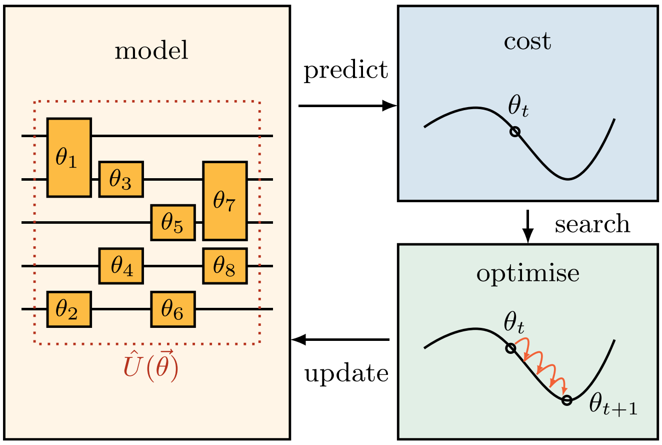


Figure 11. High level diagram of hybrid quantum–classical training algorithm for variational circuit. Quantum device (yellow) calculates terms of a cost function, then classical device calculates better circuit parameters $\vec{\theta}$ (blue, green). Repeat cycle until desired accuracy is achieved.

$$C(\vec{\theta}) = tr(\hat{U}(\vec{\theta})\rho\hat{U}^{\dagger}(\vec{\theta})\hat{H}) \tag{29}$$

with initial state ρ, trainable PQC $\hat{U}(\vec{\theta})$, and Hermitian operator $\hat{H}$. This general framework can be applied to a wide range of linear optimization problems and extended to solve nonlinear variational problems [144, 145]. Applications include our chosen example of electronic structure for quantum chemistry and materials science [146, 147], for which we present further details in section 3).

The past years have seen rapid developments in the variational approach, and it is clear there are beneficial properties for NISQ-era devices [143]. It is possible to use many separate repeats of a shallow (short) quantum circuit. This avoids the need for long coherence times. Moreover, the variational parameter tuning can compensate for several types of errors. Suitable error mitigation strategies have been developed for NISQ devices [148, 149] that can be integrated into VQA circuits.

In this section, we discuss the most relevant VQAs for computational science: the QAOA (section 2.5.1); variational quantum eigensolvers (VQE) (section 2.5.2); and simulation of quantum systems (section 2.5.3). We also briefly cover QNNs in section 2.5.4. Then we outline some known limitations common to all VQAs, due to noise and the barren plateau problem (section 2.5.5), along with pointers to current research to overcome these and extend VQAs in multiple directions.

2.5.1. *Quantum approximate optimization algorithm (QAOA).*

The QAOA was originally introduced to solve combinatorial optimization problems [139] on gate-based architectures. This was generalized as the quantum alternating operator ansatz [140]. Instead of applying the full Hamiltonian $\hat{H} = \hat{H}_{\mathrm{M}} + \hat{H}_{\mathrm{P}}$ continuously in time, the aim is to map some input state to the ground state of a problem Hamiltonian $\hat{H}_{\mathrm{P}}$ by sequentially applying the ansatz

$$\hat{U}(\gamma,\beta) = \prod_k e^{-i\beta_k\hat{H}_{\mathrm{M}}} e^{-i\gamma_k\hat{H}_{\mathrm{P}}}, \tag{30}$$

with mixer Hamiltonian $\hat{H}_{\mathrm{M}}$. The ansatz is applied in short discrete steps, that approximate the continuous time evolution, with parameters $\vec{\theta} = (\vec{\gamma},\vec{\beta})$ to be optimized. The full Hamiltonian is often the transverse Ising Hamiltonian (28), i.e. $\hat{H}_{\mathrm{M}} = \hat{H}_0$, and $\hat{H}_{\mathrm{P}} = \hat{H}_{\mathrm{p}}$, but the method can be applied more generally to any Hamiltonian with non-commuting parts.

In-depth studies of QAOA performance [150, 151] developed an efficient parameter-optimization procedure to find high-quality parameters. This becomes increasingly important as the depth and complexity of the QAOA circuit increase. The original QAOA proposal suggests a random selection of initial parameters within a range believed to be close to the optimal parameters. However, this method can often hinder the algorithm performance when the cost function landscape is rugged and contains numerous local minima or barren plateaus (section 2.5.5). Blekos *et al*'s review [152] outlines various methods that address QAOA optimization. For example, we can provide a good initial guess [150] by warm-starting initial parameters [153] or through parameters transfer [154], or by developing a more efficient optimization subroutine [155–158]. Other approaches involve exploiting the problem structure and symmetries to enhance the optimization process [159, 160].

Note that many proposed techniques rely on heuristics and empirical observations rather than theoretical guarantees. However, Brady *et al* [161] (building on the work of Yang *et al* [162]) put the method on a sound theoretical base by showing that QAOA converges to an optimal quantum annealing schedule in the limit of many stages. Gerblich *et al* [163] used the same methods to show that a sequence of quantum walks also approximates optimal quantum annealing in the limit of many stages, and shows better convergence for the same number of stages. Each stage requires its own parameters to be optimized, so it is helpful to minimize the number of stages to keep the parameter tuning costs down. Most studies focus on the few-stage regime where the theoretical results do not apply and are thus based on numerical simulations. It is important to count the optimization costs correctly to assess the full quantum cost, since each round of classical optimization requires at least one quantum run to evaluate the cost function. In a slightly different setting (quantum annealing approximated by Bézier curves), Schulz *et al* [142] show how ignoring optimization costs can distort the apparent scaling in numerical results, and they also show how to account for the optimization costs in the full analysis, recovering the expected scaling.

2.5.2. *Variational quantum eigensolvers (VQEs).*

VQEs were originally developed to find the ground state energy of molecules [147, 164, 165]. They have since been adapted to solve a wider range of problems, including optimization [143, 166]. The cost function is defined as the expectation value of the quantum state energy, $\langle\psi(\theta)|\hat{H}|\psi(\theta)\rangle$. The aim is to minimize the expectation value of Hamiltonian $\hat{H}$ over a trial state $|\psi(\theta)\rangle = \hat{U}(\theta)|\psi_0\rangle$ for ansatz $\hat{U}(\vec{\theta})$ and initial state $|\psi_0\rangle$.

The Rayleigh–Ritz principle states that the lowest eigenvalue of a Hermitian operator $\hat{H}$ is upper-bounded by the minimum expectation value found by varying the state $|\psi\rangle$. In other words, the statement

$$\langle\hat{H}\rangle(\vec{\theta}) \equiv \langle\psi(\theta)|\hat{H}|\psi(\theta)\rangle \geq E_0 \tag{31}$$

is valid with the equality holding if $|\psi(\theta)\rangle$ is the ground state $|E_0\rangle$ of $\hat{H}$. Hence, the optimized vector $\vec{\theta}$ that approximates the ground state (or eigenvector corresponding to the lowest eigenvalue) is the choice that minimizes $\langle\hat{H}\rangle$. Using this principle, we can break down the VQE methodology into three distinct steps, which are then repeated:

(i) Prepare a parameterized initial state as the starting ansatz $\hat{U}(\vec{\theta})|\psi_0\rangle$ using initial values of $\vec{\theta}$.
(ii) Estimate expectation value $\langle\hat{H}\rangle(\vec{\theta})$ in the quantum processor (this may require multiple runs using QPE).
(iii) Use a classical optimizer to find a new set of $\vec{\theta}$ values that decreases $\langle\hat{H}\rangle(\vec{\theta})$.
(iv) Repeat the above procedure to achieve convergence in $\langle\hat{H}\rangle(\vec{\theta})$. The final values of the parameters $\vec{\theta}$ at convergence define the desired state.

An application in quantum chemistry uses VQEs to generate PQCs which can be used to produce ansätze states to approximate molecular ground states [147]. These are an improvement over unitary coupled-cluster (UCC) circuits [167] based on the coupled-cluster method in computational chemistry, which are too deep to be easily implemented on NISQ quantum computers. See section 3.2.5 for more details and context. PQCs are composed of gates that are natural operations on the quantum hardware and hence more efficient than standard quantum circuits.

2.5.3. Variational quantum simulation (VQS). Simulating the dynamics of quantum systems is expected to be one of the first useful applications of quantum computers. Variational approaches make best use of limited quantum hardware by optimizing the parameters classically. There are two basic algorithms for VQS, specifically for solving real- and imaginary-time evolution of quantum systems. These can be described respectively by the time-dependent Schrödinger equation

$$\frac{\mathrm{d}|\psi(t)\rangle}{\mathrm{d}t} = -i\hat{H}|\psi(t)\rangle \tag{32}$$

with Hamiltonian $\hat{H}$ and state $|\psi(t)\rangle$, and normalized Wick-rotated Schrödinger equation

$$\frac{\mathrm{d}|\psi(\tau)\rangle}{\mathrm{d}\tau} = -(\hat{H} - \langle\hat{H}\rangle)|\psi(\tau)\rangle, \tag{33}$$

where $\tau = -it$, and $\langle\hat{H}\rangle = \langle\psi(\tau)|\hat{H}|\psi(\tau)\rangle$ preserves the norm of state $|\psi(\tau)\rangle$.

The direct approach to quantum simulation is to apply a unitary circuit $e^{-i\hat{H}t}$ to the initial state, using numerical integration. However, this increases the circuit depth polynomially with respect to the evolution time t [168] which is difficult to achieve in hardware with short coherence times. On the other hand, variational quantum simulations use an ansatz quantum circuit

$$|\phi(\vec{\theta}(t))\rangle = \hat{U}(\vec{\theta}(t))|\phi_0\rangle \tag{34}$$

applied to an initial state $|\phi_0\rangle$ to represent $|\psi(t)\rangle$. Therefore, we must choose the variational parameters to map the time evolution of the Schrödinger equation for $|\psi(t)\rangle$ on to the evolution of parameters $\vec{\theta}(t)$. The aim is to obtain a shorter circuit than would be needed for direct simulation. How the parameters evolve and simulate the time evolution depends on the variational principle used [169].

The variational method for generalized time evolution can also be used to perform matrix multiplication and to solve linear systems of equations [170]. It is an alternative method to those presented in [114, 171], or the HHL algorithm (section 2.3.7).

2.5.4. Quantum neural networks (QNNs). Artificial neural networks (NNs) [172] allow us to classify and cluster large datasets by training the parameters associated with the neural connections. This is analogous to the learning process in the brain. QNNs combine artificial neural networks with quantum principles [173]. NNs are highly nonlinear models, whereas QNNs obey the laws of quantum mechanics and contain linear operators acting on quantum states. QNNs can offload nonlinearities to classical computers, or use quantum kernels. The latter option encodes classical data into Hilbert space using a nonlinear transformation, the quantum feature map [174].

There are many different proposals for QNN architectures, such as the fully quantum, feed-forward neural networks [175], and convolutional neural networks [176]. Most QNNs follow similar steps [177]: initialize a neural network architecture; specify a learning task; implement a training algorithm; and simulate the learning task.

There are similarities between QNNs and kernel methods [174, 178], which are well-established machine learning techniques. Kernel methods map the features of a dataset into a high-dimensional space through function $\phi(\vec{x})$, then uses a kernel $K(x_1, x_2)$ to measure the distance between two data points in high-dimensional space. We observe the same behavior in QNNs: when the features are mapped on to a high-dimensional Hilbert space, we find the level of similarity between two points in the space by calculating the overlap of two encoded states.

For QNNs constructed using quantum circuits, the parameters that perform an accurate mapping are defined by a set of gate operations [179], corresponding to $V(\theta)$ (figure 12). There are typically three steps. First, encode the input data, which is usually classical, into a quantum state using a quantum feature map [180] ϕ, applied to the initialized qubits through unitary

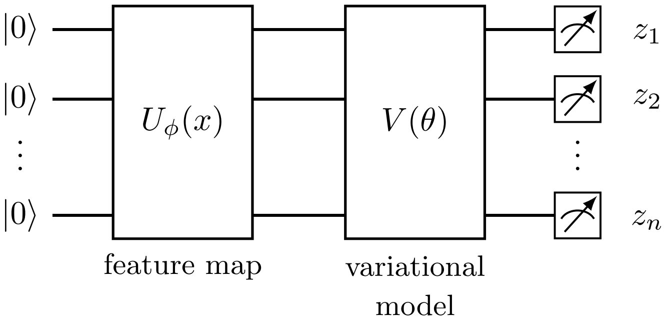


Figure 12. Example of quantum neural network using variational quantum circuit.

operation $U_\phi(x)$ (figure 12). The choice of feature map aims to enhance the QNN performance.

$$\hat{U}_\phi(x)|0\rangle^{\otimes n} = |\phi(x)\rangle. \tag{35}$$

Second, we use a variational model parameterized by vector $\vec{\theta}$ to transform the quantum state

$$|\psi(x,\vec{\theta})\rangle = V(\vec{\theta})|\phi(x)\rangle. \tag{36}$$

It contains parameterized gate operations that are optimized for a particular task, analogous to classical ML techniques [39]. The aim is to choose parameters $\vec{\theta}$ to minimize a loss function. Third, we measure an observable $\hat{M}$. Classical post-processing procedures on the measurement outcomes $\vec{z} = [z_1, z_2, \ldots, z_n]$ extract the output of the model,

$$f(\vec{z}) = \langle\psi(x,\vec{\theta})|\hat{M}|\psi(x,\vec{\theta})\rangle. \tag{37}$$

The main advantage of variational circuits actually arises from the feature map: when $\hat{U}_\phi(x)$ cannot be simulated efficiently on classical computers, the overall quantum speedup can be up to exponential.

Many types of QNNs have been proposed for different types of data analysis, such as quantum Boltzmann machines [181, 182] which are a natural fit for quantum annealing [183, 184] hardware, as are quantum support vector machines [185, 186]. Constructing QNN architectures is currently an active area of research. For a concise review of QML, see [187] and references therein.

2.5.5. VQA limitations and enhancements. VQAs are a very active area of current research, and face a range of problems (see [143, 187] for an overview). In general, there is a trade-off between the expressivity and trainability of the circuit [188]. Performance decreases significantly when qubit number and circuit depth increase. Using more quantum resources to implement the ansatz results in greater expressivity of the ansatz, which increases the likelihood of finding the correct solution [189–191]. However, large circuit depth means the classical optimizer would receive noisy and potentially unusable gradient information, which can lead to divergent optimization or barren plateaus [192, 193]. We illustrate these two common problems in figure 13.

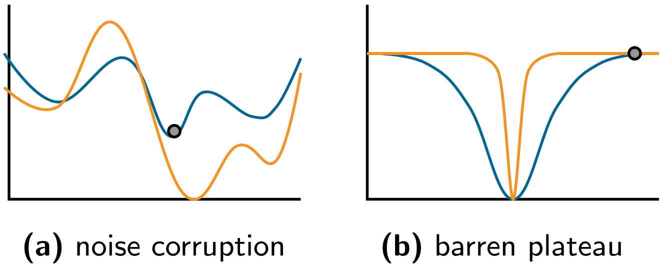


Figure 13. Ideal solution landscape (blue) corrupted by (a) noise and (b) barren plateau, both orange.

The barren plateau phenomenon occurs when the non-convex loss function landscape becomes exponentially flat (on average) as qubit number increases. At the same time, the valley containing the global minimum also shrinks exponentially with problem size, leading to a narrow gorge [194] (figure 13(b)). The cost function gradient is exponentially suppressed except in an exponentially small region. For cost functions in (29), barren plateaus can be caused by circuit properties $\hat{U}(\vec{\theta})$ such as its structure and depth [192, 193, 195], expressibility [188, 196] and entanglement power [197–199]. The effects on parameter optimization can potentially destroy any quantum speedup.

There have been considerable efforts to understand and mitigate the effects of barren plateaus, for a review, see Larocca *et al* [200]. These include building shallow circuits [193, 201, 202], developing gradient free algorithms [203], using error mitigation techniques [204, 205], and developing more sophisticated state preparation strategies when initializing the variational parameters [206–208]. Others have presented the mathematical foundations for a unified Lie algebraic theory to explain the sources of barren plateaus and quantify the ultimate expressiveness of parameterized circuits [209, 210]. There is also ongoing work to develop a more general barren plateau theory [211].

In addition, there have been considerable efforts in designing suitable ansätze. Quantum architecture search (QAS) dynamically designs optimal ansätze for VQAs [212–214]. Given a search space (possibly constrained by quantum hardware properties), performance criteria, and the underlying problem, QAS aims to automatically find an optimal quantum circuit with parameters $\vec{\theta}'$ that maximize the VQA performance. For VQEs, there are approaches which construct variational models constrained to a (smaller) symmetry-respecting solution space to preserve the problem Hamiltonian symmetries [215–217]. This results in smaller circuits more suitable for NISQ hardware. This has been extended to QML models [218–220].

Quantum optimal control (QOC) theory [221, 222] has also inspired work into finding connections with VQAs [223]. For example, the quantum-optimal-control-inspired ansatz [224] goes beyond symmetry-preserving methods by introducing a set of unitaries that break the symmetries of the problem Hamiltonian.

2.6. Quantum software

As well as quantum hardware, making use of quantum computers also needs new software at various levels of the quantum computing stack, see figure 4. This includes programming languages, compilers, operating systems, and low level classical control software. Quantum software start up companies are promising seamless integration into existing programming languages, such that you will not need to know you are using a quantum computer. HPC users will be well aware that this is hype. Good performance is only achieved with some consideration of how best to map an application to the hardware. For large scale HPC, this is increasingly becoming a professional team effort, requiring research software engineers specialized in tuning applications to the specific HPC hardware it is running on. This is especially true for using GPUs, and for task scheduling to ensure all available compute nodes are kept busy. Since adding QPUs to HPC increases the hardware complexity, we can expect to need the same level of professional team support to fully leverage future large scale quantum computers. Nonetheless, current quantum computers that are available for use in the cloud have not yet reached this level of complexity, and writing and executing test bed applications is well within reach of the average computational scientist or engineer. There are many tutorials available: each available platform provides its own, and there are guides and open source code for specific applications, see, for example, [225] for optimizing crystal structures on D-Wave Systems.

Currently available commercial quantum computers tend to use Python modules for the user-level programming language, and provide simulators, compilers and optimizers specific to their platform. In particular, Qiskit from IBM [66] is open source and has been adapted for other platforms such as ion traps and neutral atoms. As one of the first online services, IBM has the most comprehensive set of online resources, and also provide a visual drag and drop interface for creating quantum circuits, which is especially suitable for pedagogical uses. Programming gate by gate is very low level, about equivalent to assembler code for classical computers, despite the high level interface. This is indicative of the early stage of development of quantum hardware with only tens or hundreds of qubits, where circuit optimization by hand is still possible. Indeed, hand optimization will tend to beat the automated compilers for specific applications (the automated compilers are generic), and is necessary to obtain optimal performance from the limited capabilities of NISQ systems. On some platforms (including IBM), it is even possible to experiment at the hardware pulse control level to optimize quantum gate operations. Equivalently low level control is used for non-circuit model hardware, such as quantum annealers, where the programming consists of mapping the problem to the hardware settings (fields and couplers) and shaping the annealing schedule. Higher level tools to map the problem to the hardware graph across 5000 qubits are available [226, 227], but do not necessarily beat a hand-crafted assignment for specific problems.

There are are several software start ups specializing in operating systems and control software designed to operate equally well on multiple platforms and provide a similar interface regardless of the underlying hardware, even recommending the best choice if multiple platforms are available. While we are still some way from this being viable for using quantum computers, the automated lower level control techniques are showing promise when integrated correctly [228]. Meanwhile, those who are ready to be quantum computing pioneers will find ample support and opportunities online, while those who prefer to wait for the next advances can be assured that software at an appropriate level will be available, either commercially or open source, when the hardware is ready.

2.7. Performance benchmarks: quality, speed, scale

Benchmarking for high performance computers is well established, and essential for managing HPC facilities and optimizing their throughput. For example, the High Performance Linpack benchmark [229] solves a dense linear system to measure the throughput of a computing system. Running the same computational tasks on different HPC hardware can provide some level of side-by-side comparison of different facilities. However, the results from standard benchmarks cannot fully predict performance for a specific application, and bespoke tests are often run to tune performance on a per-application basis.

Benchmarks for quantum computers are under active development. Fair comparisons of quantum hardware are even more challenging than for classical hardware, given the diverse types of qubits under development. Hardware based on superconducting qubits [230, 231] is arguably the most popular choice [232, 233], followed by ion-trap-based systems [234–236]. Semiconductor-based spin qubits [237, 238] are less advanced but potentially offer more scalability in future. Photonic platforms [239, 240] require less stringent cooling regimes but other aspects of the engineering (e.g. single photon sources and photon loss) are challenging. Neutral atoms, especially using highly excited Rydberg states [241], are versatile and already showing impressive results for directly simulating quantum many-body systems [242, 243]. Further types of hardware are under active development, and it is not clear yet which will prove the most useful and scalable in the long term.

The criteria for good quantum computer hardware are fairly straightforward [63]. It must have as many high quality qubits as possible. Qubits must be individually controlled to generate complex (entangled) states. We must be able to apply a large number of sequential operations before the qubits lose coherence. The control and gate operations are characterized by the quality of single- and two-qubit gates. Scaling up to obtain good performance on large numbers of qubits requires error correction [48, 49] to be implemented in the hardware. This needs mid-circuit measurements to be possible, as well as measuring the qubit states at the end of the computation.

2.7.1. Low- and high-level benchmarks. Low-level benchmarks measure the performance of individual qubits, single- and two-qubit gate fidelities, state preparation and measurement operations. We can often directly measure the

$T_{1,2}$ decoherence times, gate speed or gate operation reliability (gate fidelity). However these characteristics do not provide an accurate benchmark of quantum processor performance. Using low-level benchmarks alone can label a slow high-fidelity quantum computer as equivalent to a fast low-fidelity quantum computer [244].

For more thorough testing of gate operations, there is randomized benchmarking [245, 246], cycle benchmarking [247], or gate set tomography [248]. Efficient methods to verify the output state are now available [249], when the expected state is known. Randomized benchmarking protocols typically characterize the quality of Clifford operations, which are classically simulable (see section 2.1). For non-Clifford operations, random circuit sampling and direct randomized benchmarking can be used [250, 251]. These methods may not be scalable to large numbers of qubits as it is difficult to classically simulate non-Clifford operations to check the test outcomes. A class of methods that aims to circumvent this issue is mirror benchmarking [252, 253]. The trivial ideal output of the circuits involved in these benchmarks makes the results easily verifiable and scalable to large numbers of qubits.

At a slightly higher level, specific quantum circuits can be used to test hardware performance for gate sequences. The circuits can be random, application-agnostic circuits [254, 255], or circuits run as part of certain algorithms [256–258]. Other types of gate sequences can be used including identity operations and entanglement generation circuits [259]. For non-gate-based hardware, a generalized form of randomized benchmarking can be used, for example in programmable analogue quantum simulators [260].

Qubit layout and connectivity constrain the implementation of gate sequences and hence affect algorithm efficiency. Some types of hardware (e.g. neutral atoms [241]) can easily be reconfigured between runs. Other platforms, such as superconducting or semiconductor qubits have connectivity baked in at the manufacturing stage. Algorithm- [261] or application-level [262, 263] benchmarks are thus important for a full understanding of hardware performance. Such benchmarks require knowing the ideal output in order to evaluate the solution quality. In some cases, analytical solutions are available for suitable problems. For example, the one-dimensional Fermi–Hubbard model is exactly solvable using the Bethe ansatz [264]. For a comprehensive assessment of performance, there are benchmarks comprising a suite of many different applications (e.g. [265–267]). Most include well-known algorithms such as the QFT (section 2.3.1), QPE (section 2.3.2), Grover's search algorithm (section 2.3.4), VQE (section 2.5.2), the Bernstein-Vazirani algorithm [46], and the hidden shift algorithm [268]. In general, device performance on simple problems or circuits may not scale [262]. It is also necessary to define suitable high-level success criteria, e.g. for quality [256, 269].

It is instructive to test at both low and high levels because extrapolating from low level performance to higher level is unreliable. Cross-talk errors can dominate and these are not captured by lower level tests using only one- or two-qubit gates. Even at a higher level, there may be large differences between performance on random and structured circuits [255, 262].

Determining whether an actual computation has run correctly requires verification. Classical verification methods are becoming increasingly sophisticated for larger, more complex problems (see, for example, [270]). Verifying quantum computations [51], especially for NISQ hardware, requires some extra tools, such the accreditation scheme of Ferracin *et al* [271] for gate-based computations, which has been experimentally implemented [272]. Accreditation makes use of similar tools to low-level benchmarking, like mirror circuits, to quantify the errors and hence assign a confidence level to the correctness of the output of the computation, which has been run alongside the test circuits. Accreditation has been extended beyond the circuit model to certain types of quantum simulation and quantum annealing [273].

2.7.2. Quantifying quantum device performance. Various combinations of benchmarks have been proposed as metrics to capture the performance of quantum computers in a single number or a few parameters. Most of these use fairly low level benchmarks. For example, quantum volume is a benchmark developed by IBM [254] to test hardware performance for gate sequences. Quantum volume is calculated by running randomized square circuits with different numbers of qubits. The quantum volume is given by the largest square circuit that can be run on the device which passes an acceptance criterion that quantifies the output quality. It has been generalized to more realistic non-square circuits [269, 274].

Metrics such as *circuit layer operations per second* (CLOPS) [275], layer fidelity [276], and qubit number have been proposed to collectively quantify the quality, speed, and scale of quantum computers. Single-number metrics are useful to make cross-device comparisons straightforward, but similar to classical metrics, they may not accurately describe quantum computer performance for all kinds of algorithms. In a non-gate model setting, McGeoch [277] has proposed guidelines for reporting and analyzing quantum annealer performance.

As well as accuracy, it is also important to measure how quickly the quantum processor can execute layers of a parameterized model circuit similar to those used to measure quantum volume. Increased quantum processor speed is critical to support near-term variational algorithms, which require thousands of iterations. Some benchmarks evaluate speed as a sub-score and combine it with other factors to determine a final metric [263]. There also exist standalone metrics for speed [275]. Lubinski *et al* [262] note that evaluating measures of speed for comparing different platforms is difficult. Timing information (when available at all) is reported differently by different hardware vendors, and gate speeds naturally vary by several orders of magnitude between hardware types (see section 5). Another time-related factor is consistency, or how the performance varies over time. This determines how often hardware must be recalibrated. Consistency can be captured by repeating the same tests, and is an important consideration for current hardware which requires significant downtime for regular recalibration.

The above methods (section 2.7.1) and metrics (section 2.7.2) are useful for benchmarking hardware dependent characteristics. On top of this, the compilers and software stack efficiency can significantly affect performance, in much the same way as classical HPC requires highly tuned software to optimize performance. Quantum software benchmarking is still at an early stage of development, but there are indications it will be critical for making quantum hardware useful, see for example, [228].

2.7.3. Quantum circuit simulation methods. In the current NISQ era, simulating quantum circuits is important for multiple purposes. Simulations of ideal quantum circuits allow quantum algorithm development and testing in a more reliable environment than quantum hardware currently provides, albeit limited to about 45 qubits in full generality due to the exponential size of Hilbert space in qubit number. Several benchmarking tools (e.g. [251, 278]) require knowledge of the ideal circuit outputs produced by such simulations, to verify the outputs of the quantum hardware. Further examples of tensor network-based benchmarking methods include the holoQUADS algorithm [279, 280] and qFlex [281].

Classical simulation also establishes a classical computational bar that quantum computers must pass to demonstrate a quantum advantage. It is important to carry out classical simulations as efficiently as possible, so that reports of quantum advantage are not exaggerated. One of the main tests used to claim a quantum advantage from a NISQ device involves sampling the bit-strings from a random quantum circuit (RQC) [282]. The result must be within some variational distance of the output distribution defined by the circuit [281, 283] to pass the test. Challenges to Google's supremacy claims [1] used tensor network simulation methods [284–286] to develop efficient classical simulations for imperfect hardware. Decoherence and other errors in the hardware means it can only generate entanglement over a limited range [287]. This allows sufficiently accurate classical simulations of larger numbers of qubits to be performed. The tensor network contraction method is one of the best classical methods for simulating RQCs for sizes close to the quantum advantage regime [288, 289]; for recent reviews and tutorials, see, e.g. [290–292].

2.7.4. Measuring quantum advantage. It is not straightforward to determine exactly when a quantum computer provides a practical advantage over a classical computer. The latter are benchmarked using FLOPS, which measure the number of floating-point operations per second. The equivalent speed benchmark for quantum processors is CLOPS (see section 2.7.2) which measures the number of CLOPS. It does not make sense to convert between the two metrics, since they do not correspond to the same unit of computation.

Demonstrating a quantum advantage at the level of whole algorithms also highlights the difference between theoretical computational complexity and practical implementation. Many QML algorithms make different assumptions from the classical algorithms they are compared with [58]. In some cases, it is possible to find 'dequantized' versions of QML algorithms [293]. These are fully classical algorithms that process classical data and perform only polynomially slower than their quantum counterparts. If a dequantized algorithm exists, then its quantum counterpart cannot give an exponential speedup on classical data. Many QML algorithms rely on pre-processed data via quantum encoding. Breakthroughs that claim a quantum speed-up often neglect the quantum encoding step in the algorithm runtime, while making comparisons to classical algorithms that operate on raw, unprocessed data. This means it is often possible to achieve equivalent speed-ups using only classical resources, if we perform the same pre-processing for classical algorithms. For example, the pre-processing in [294] allows us to apply 'dequantized' classical sampling techniques to the data, and achieve a significant speedup over previous classical methods. We discuss the issues around encoding classical data into quantum systems in more depth in section 5.2. On the other hand, dequantization shows how replacing quantum linear algebra algorithms with classical sampling techniques can potentially create a classical algorithm that runs exponentially faster than any other known classical algorithm [294]. Quantum-inspired classical algorithms that provide a significant improvement on current best methods are a useful benefit from quantum algorithm research, if they turn out to be practical to implement on classical HPC.

Comparing state-of-the-art quantum computers with the best classical algorithm for the same problem is not enough to establish a quantum advantage. For the RQC used in Google's claim of quantum supremacy [1], classical algorithms had not been optimized for the problem: it is not a useful problem and there was little research on it. Subsequent work brought down the classical runtime by some orders of magnitude [284–286]. Similar advances have occurred with boson sampling [295–297].

Current research predicts a narrow window in terms of the number of qubits where circuit model NISQ hardware can potentially outperform classical hardware for certain problems [298]. Larger numbers of qubits need longer gate sequences for the same algorithm. Errors inevitably build up and render the computation unreliable without active error correction. Recent proof-of-concept studies of QEC [299, 300] begin to show errors suppressed for extended gate sequences, heralding a new era of fault tolerant quantum computing.

3. Quantum simulation for quantum systems

Simulating quantum systems on classical computers has long been known for the exponential explosion problem, where the size of the quantum state space (Hilbert space) increases exponentially with the number of qubits. This limits the size of full classical simulations to a state space equivalent to around 45 qubits (2^{45} complex numbers). For quantum systems with a larger Hilbert space, this means the simulation is constrained to a subspace that may not be large enough to reveal the full

Table 3. Compare digital vs analogue quantum simulation.

	Analogue	Digital
Hamiltonian	Directly implement system Hamiltonian by experimentally building a controllable quantum system (simulator).	Express Hamiltonian in second quantized form, then construct Hamiltonian out of quantum logic gates.
System vs simulator	System may be completely different from simulator, e.g. using atoms in optical lattices to simulate electrons in a crystal.	Implements circuits containing quantum gates, therefore has same level of technical difficulty as quantum computer.
Physical model	Needs to match the simulator Hamiltonian within the available experimental controls.	In principle, any quantum system can be simulated using digital (universal) quantum simulation.
Hardware	Easier to implement for certain problems well matched to the simulator hardware. Likely to become practicable before universal quantum computers become available.	Requires high depth circuits (many quantum gates) therefore unlikely to be practical before fully error corrected quantum computers are available.

behavior. This can be overcome by using quantum simulation [11]: we use some controllable quantum system to study the model of another less controllable or accessible system. The method is especially relevant for studying quantum many-body problems [301] in condensed matter physics, atomic physics, and quantum chemistry. For authoritative reviews, see [16, 17, 302]. We first provide a brief overview of methods for quantum simulation, then, to provide a focus for a more in depth assessment, we provide a more detailed account of quantum computational chemistry and the prospects for quantum computers to advance this and related fields.

3.1. Quantum simulation methods

Quantum simulation can be implemented using quantum computers (digital quantum simulation), or with analogue devices (analogue quantum simulation) that are not necessarily as powerful as universal quantum computers and therefore are easier to build. An example of an analogue quantum simulator is the ultracold atomic systems often used to simulate condensed matter physics [303, 304] and quantum chemistry [305]. A comparison of the relative merits of the different methods is summarized in table 3.

The basic task underlying most quantum simulations involves time evolution under the system Hamiltonian. For Hamiltonian $\hat{H}$, the unitary operator $\hat{U}(t)$ evolves the dynamics for a time t,

$$\hat{U}(t) = \exp\left(-\mathrm{i}\int_0^t \hat{H}(t)\,\mathrm{d}t\right), \tag{38}$$

using units where $\hbar = 1$. For many physical systems, the system Hamiltonian $\hat{H}$ can be written as a sum of terms that describe local interactions,

$$\hat{H} = \sum_{j=1}^{n} \hat{H}_j. \tag{39}$$

In analogue quantum simulation, the system Hamiltonian $\hat{H}_{\mathrm{sys}}$ is directly mapped [306] on to the simulator Hamiltonian $\hat{H}_{\mathrm{sys}} \mapsto \hat{H}_{\mathrm{sim}}$, and the time evolution proceeds under the natural Hamiltonian evolution of the simulator.

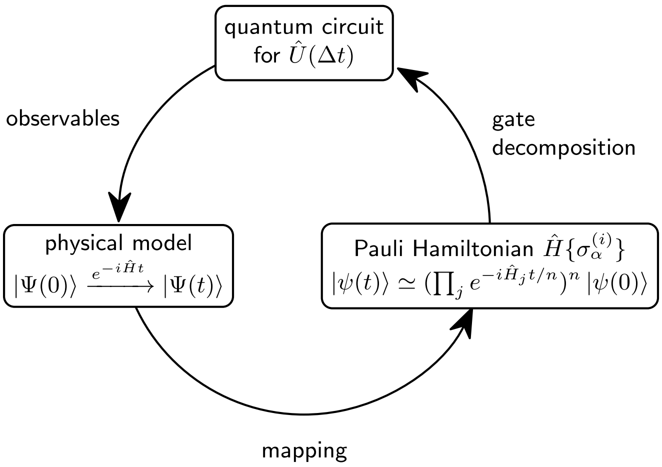

quantum circuit for $\hat{U}(\Delta t)$
observables
gate decomposition
physical model
$|\Psi(0)\rangle \xrightarrow{e^{-i\hat{H}t}} |\Psi(t)\rangle$
Pauli Hamiltonian $\hat{H}\{\sigma_\alpha^{(i)}\}$
$|\psi(t)\rangle \simeq (\prod_j e^{-i\hat{H}_j t/n})^n |\psi(0)\rangle$
mapping

Figure 14. Conceptual diagram of universal simulator on digital quantum computing device. [308] John Wiley & Sons. © 2019 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim.

In contrast, implementing $\hat{U}(t)$ on a digital quantum computer usually relies on numerical integration, i.e. breaking up the evolution time into small steps of duration Δt, taking care to preserve the order of non-commuting operators. The first-order Trotter–Suzuki formula gives

$$\hat{U}(\Delta t) = e^{-i\sum_j \hat{H}_j \Delta t} = \prod_j e^{-i\hat{H}_j \Delta t} + O((\Delta t)^2), \tag{40}$$

and higher order approximations can also be used to improve accuracy [307]. Digitizing Hamiltonian time evolution in this way is often referred to as 'Trotterization'. Acceptable precision comes at the expense of small Δt, which requires many quantum gates and hence extremely deep circuits, well beyond NISQ capabilities for useful problem sizes.

Figure 14 gives a conceptual illustration of how to perform quantum simulations on a digital quantum computer. A physical model describes the quantum state evolution of interest $\Psi(t)$; this evolution can be approximated to arbitrary precision by mapping the given model to a spin-type Hamiltonian which is easily encoded onto a qubit register. The sequence of unitary operations to implement the time evolution using (40) or higher order variants can be programmed as a quantum circuit, giving the approximated evolved state as output $\psi(t)$ starting from a given input state $|\psi(0)\rangle$. The mapping from the physical system to qubits can then, in principle, be reversed to obtain the desired physical output state. However, since the output is a quantum state, not a classical description of it, in practice various types of post-processing are usually required to compute properties of interest (e.g. see section 2.3.2).

Although time evolution can be readily realized using Trotter product formulas, accuracy requires many small time steps, leading to very deep circuits. Recent work has demonstrated that variational approaches can speed up the calculation of Trotter terms [309]. Moreover, tensor network methods (see section 2.7.2) can be used to generate quantum circuits that can be more efficient than Trotterization [310].

One of the main classical methods for simulating quantum systems beyond the constraints imposed by the exponential size of Hilbert space is quantum Monte Carlo (QMC) [311–313]. Instead of a full classical simulation, which would be beyond the reach of HPC, individual quantum trajectories are sampled to build up a statistical average of the properties of interest.

The method requires large numbers of samples to achieve reasonable accuracy, often on the order of millions of samples, which can take hours or even days on current HPC. As well as computational chemistry [313], QMC methods are widely used in materials science [311], nuclear physics [314], and condensed matter physics [315, 316].

In all fermionic quantum systems, QMC can suffer from the 'sign problem' where enforcing fermion exchange statistics leads to a lack of convergence in the QMC sampling [317]. This is where quantum simulators that can naturally represent fermionic properties are particularly promising [318, 319]. Quantum field theories are used to describe condensed matter many-body systems. Work on quantum simulation in this setting, e.g. quantum algorithms [320], and analogue [321] and digital simulators [322], has evolved into an experimental field that has attracted the attention of particle and nuclear physicists [323, 324]. In particular, after the first quantum degenerate Bose and Fermi gases were experimentally created [325, 326], these became a versatile analogue simulator to study the behavior of condensed matter systems [327, 328]. The quantum gas microscope is another significant advance, allowing research into quantum-degenerate gas phenomena where particle indistinguishability plays a key role [318, 319].

Quantum field theory is also the main model used in HEP. In 2016, the first full quantum simulation of a HEP experiment demonstrated the creation of electron–positron pairs from energy [329]. Gauge field theories are the underlying formalism describing interactions among elementary particles in the Standard Model, and can also be extended to physics beyond the Standard Model. Quantum simulators can be used to study gauge field theories and investigate non-perturbative dynamics at strong coupling. Rapid advances in atomic, optical, molecular, and solid-state platforms [17] have given us analogue quantum simulators to explore HEP models. Better HEP simulations may eventually help us understand difficult problems like quark confinement and the properties of dense nuclear matter, e.g. neutron stars. Bauer *et al* [324] provide a roadmap of quantum simulations for HEP.

In the NISQ era, while we wait for digital quantum hardware to develop, VQAs (section 2.5) have been shown to produce meaningful results with shallow circuits and without active error-correction procedures. There has been work on simulating field theories, including finding the lowest lying energy spectra of low-dimensional Abelian and non-Abelian lattice gauge theories [146, 330, 331]. It is important to note that many lattice gauge theory simulations on NISQ hardware require fully or partially removing redundant degrees of freedom or imposing symmetries. Simplifications like these are essential in the resource-limited NISQ era.

In the next sections, we provide a more detailed account of the potential of quantum computers to advance quantum simulation in quantum computational chemistry. Many of the methods are more widely applicable, and can provide insight for adjacent and related fields.

3.2. Quantum computational chemistry

One of the original motivations for the development of quantum computers was to be able to accurately simulate and characterize systems of interacting fermions [332, 333]. Obtaining accurate solutions of the electronic structure of many-body systems is a major challenge in computational chemistry. Research has been carried out to develop algorithms to solve the electronic structure problem using quantum computers [334–336]. Fault-tolerant quantum computers are predicted to solve the electronic structure problem for many-body systems [337] in polynomial time, for example, using the QPE algorithm [70, 73, 332] (section 2.3.2). It is possible that a quantum advantage exists even in the absence of a clear polynomial scaling. A detailed overview of the state-of-the-art methods developed in this field can be found in [337, 338]. In practice, the possibility of achieving an exponential quantum advantage for quantum chemistry problems remains controversial [339]. It is challenging for quantum simulations to achieve quantum advantage over the best classical computations, given the decades of work already done on classical methods for quantum systems. As classical algorithms continue to improve, quantum information processing has been useful in developing advanced 'quantum-inspired' classical techniques [340]. In this regard, we review new techniques for classical and quantum hardware from different perspectives. We cover a range of methods applicable to problems such as *ab initio* simulations, Hubbard models, spin Hamiltonians, interacting Fermi liquids, solid-state systems, and molecular systems.

Modern computational chemistry aims to simulate molecules and materials within 1 kcal mol^{-1} accuracy, which is required to predict chemical phenomena at room temperature. Despite the tremendous progress in quantum chemistry algorithms in the past century, such a level of precision can only be achieved for small systems due to the polynomial scaling (higher than O(x^4) with system size x) of advanced computational chemistry methods [341]. Hence, current research has a strong emphasis on developing hybrid quantum–classical and quantum algorithms; estimating quantum resources; and designing quantum hardware architectures for simulations of practical interest.

Another central quantity in statistical mechanics and materials science is the partition function. All thermodynamic quantities of interest can be derived from the partition function, which then help us understand the behavior of many-body systems. Calculating partition functions of physical systems is generally a #P-hard complexity class problem [342, 343]—both classical and quantum methods are unlikely to provide exact solutions. Quantum algorithmic attempts to estimate partition functions involve QPE [344] combined with a quantum version of the Monte Carlo Metropolis algorithm, and algorithms based on quantum walks (section 2.3.6) [345, 346]. Hybrid quantum–classical algorithms [347] and quantum-inspired algorithms [348] have also been proposed, with quantum algorithms potentially capable of performing better than classical ones.

Even with noisy hardware, the long-term impact of using quantum simulation for quantum chemistry and materials is promising. For example, Google's Sycamore quantum processor used a hybrid quantum–classical Monte Carlo method [1] to calculate the atomization energy of the strongly correlated square H_4 molecule. The results are competitive with those of the state-of-the-art classical methods. The algorithm relies on being able to efficiently prepare a good initial guess of the ground state on the quantum hardware, which is a non-trivial step.

Understanding the strengths and weaknesses of classical algorithms can indicate where we may see a quantum advantage. It also indicates where issues may arise in quantum simulation approaches in the near and long term. For example, Tubman *et al* [349] used classical simulations to provide resource estimates for crucially important but often neglected aspects of quantum chemistry simulations. These include efficient state preparation for quantum methods such as phase estimation. Sections 3.2.1–3.2.4 present an overview of techniques in computational chemistry along with difficulties encountered in modelling chemical systems. Sections 3.2.5 and 3.2.6 cover early applications of quantum computing within computational chemistry. Section 3.3 discusses two examples of chemical systems with industrial applications that are beyond the capabilities of classical computational chemistry methods.

3.2.1. Quantum chemistry's many-electron problem. The core objective of quantum chemistry is to solve the stationary Schrödinger equation for an interacting electronic system in the static external field of the nuclei, which describes isolated molecules in the Born–Oppenheimer approximation. The problem can be defined as follows. There are N electrons in the field of stationary M point charges (nuclei). Electrons interact only through electrostatic Coulomb terms and are collectively described by a wave function with $4N$ coordinates $\vec{x}_j = (\vec{r}_j, \vec{\omega}_j)$, i.e. $3N$ spatial coordinates $\vec{r}_j$ and N spin coordinates ω_j. The antisymmetric electronic wave functions are solutions of the time-independent Schrödinger equation

$$\hat{H}\Psi(\vec{r}) = E\Psi(\vec{r}) \tag{41}$$

where we describe the motion of the N electrons (j, k indices) in the field of the M nuclei (A index) with Hamiltonian [338]

$$\hat{H} = \underbrace{-\sum_j \frac{\nabla_j^2}{2}}_{=\hat{H}_\mathrm{K}} \underbrace{-\sum_{jA} \frac{z_A}{|\vec{r}_j - \vec{r}_A|}}_{=\hat{H}_\mathrm{e\text{–}n}} + \underbrace{\frac{1}{2}\sum_{k\neq j} \frac{1}{|\vec{r}_j - \vec{r}_k|}}_{=\hat{H}_\mathrm{e\text{–}e}} \tag{42}$$

in atomic units z. The Hamiltonian contains the kinetic term $\hat{H}_\mathrm{K}$, the electron–electron interaction $\hat{H}_\mathrm{e\text{–}e}$, and the electron–nuclei interaction $\hat{H}_\mathrm{e\text{–}n}$. When there are two or more electrons, we cannot solve the Schrödinger equation exactly. This is the fundamental challenge of computational chemistry.

There are two main families of approximations to solve this problem: *Hartree–Fock* (HF) [350] methods with refinements and *density functional theory* (DFT) [351, 352]. Each has its own opportunities for quantum enhancements. We summarize the main methods and their limitations before discussing potential quantum enhancements.

3.2.2. Hartree-Fock (HF) and the electron correlation problem. The HF (mean-field) method [353] describes the potential experienced by each electron resulting from the average field of all electrons. This means that the $\hat{H}_\mathrm{e\text{–}e}$ term in (42) is replaced by a mean-field potential to make the equation solvable. However, this approximation does not accurately describe the electron-electron correlation. Electron correlation accounts for the fact that electrons do not move independently in the average field of other electrons rather, their movement is influenced by the instantaneous positions and motions of other electrons. The particle exchange energy resulting from the anti-symmetric wave function (i.e. Pauli principle)

$$\Psi(\vec{x}_1,\ldots,\vec{x}_j,\vec{x}_k,\ldots,\vec{x}_N) = -\Psi(\vec{x}_1,\ldots,\vec{x}_k,\vec{x}_j,\ldots,\vec{x}_N), \tag{43}$$

means we cannot express the wave function as a separable product of single-particle wave functions,

$$\Psi(\vec{x}_1,\vec{x}_2,\ldots,\vec{x}_N) \neq \psi_1(\vec{x}_1)\psi_2(\vec{x}_2)\ldots\psi_N(\vec{x}_N) \tag{44}$$

where the spatial and spin parts are defined as $\psi_j(\vec{x}_j) = \phi_j(\vec{r}_j)\chi_j(\vec{\omega}_j)$. Instead, the HF method approximates the many-body wave function with a Slater determinant

$$\Psi(\vec{x}_1,\vec{x}_2,\ldots,\vec{x}_N) = \frac{1}{\sqrt{N!}}\begin{vmatrix} \psi_1(\vec{x}_1) & \psi_2(\vec{x}_1) & \ldots & \psi_N(\vec{x}_1) \\ \psi_1(\vec{x}_2) & \psi_2(\vec{x}_2) & \ldots & \psi_N(\vec{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \psi_1(\vec{x}_N) & \psi_2(\vec{x}_N) & \ldots & \psi_N(\vec{x}_N) \end{vmatrix}, \tag{45}$$

where the set of $\{\psi_j(\vec{x}_k)\}$ are chosen to minimize the ground state HF energy

$$E_0 = \langle\Psi_0|\hat{H}|\Psi_0\rangle. \tag{46}$$

The wave function depends on the Hamiltonian, and, in turn, the mean-field part of the Hamiltonian depends on the wave function. This means that the HF method must be solved iteratively using the self-consistent field (SCF) method. We start from an initial guess for the wave function, build an initial Hamiltonian, and then calculate the initial energy. We change the wave function, update the Hamiltonian, and calculate the energy until the difference between cycles is lower than a defined threshold. The SCF method relies on the variational principle, which states that the energy of any approximate wave function is higher than or equal to the exact energy. The final energy is the exact HF energy E_0. This value is always higher than the true ground-state energy because the formalism cannot fully account for electron correlation effects. Recovering the correlation energy is the main goal of post-HF methods.

For small systems, such as atoms, the unknown ψ functions in (45) can be solved numerically on a three-dimensional grid of points. Larger systems require grids that are too dense to reach convergence. Hence we express the ψ functions as a linear combination of a set of known functions, the basis set:

$$f(x) = \sum_j c_j \chi_j(x). \tag{47}$$

In practice, the basis set could be built from any type of function χ_j following two criteria: they should reproduce the behavior of the electron density (the highest density should be located at the atom positions, and they should monotonically decrease to zero at an infinite distance from the nuclei) and they should be easy to integrate. Among the most widely used basis sets, there are Gaussian and Slater-type basis sets for molecules [354] and plane waves for periodic systems [355].

Selecting the appropriate basis set is a critical consideration. It is a balance between achieving the desired accuracy and managing computational resources. Classical computer simulations often employ extensive basis sets, which can provide high levels of accuracy, including up to 30 functions per atom [356, 357]. The computational capacity of classical computers typically means that the basis set size is not the primary limitation on accuracy. In contrast, NISQ devices impose limitations on the basis set size due to hardware constraints [358], making the basis set size a more significant factor affecting accuracy as McArdle *et al* discuss in their comprehensive review [338].

3.2.3. Density functional theory (DFT). DFT [359] focuses on the electronic density

$$n(\vec{r}) = N \int \mathrm{d}^1 r_1 \int \mathrm{d}^2 r_2 \ldots \int \mathrm{d}^N r_N |\Psi(\vec{r}, \vec{r}_2, \ldots, \vec{r}_N)|^2 \tag{48}$$

rather than electron coordinates. DFT systematically maps the many-body problem to a single-body problem through the particle density $n(\vec{r})$ from which we calculate all other observables.

The Hohenberg–Kohn (HK) theorem is at the heart of DFT. It states that the system's ground-state energy is uniquely determined by the ground-state density $n_0(\vec{r})$. The ground-state wave function $\Psi_0(\vec{r}_1, \vec{r}_2, \ldots, \vec{r}_N)$ must both reproduce the ground-state density $n_0(\vec{r})$ and minimize the energy. Hence the aim is to minimize the sum of kinetic, interaction and potential energies

$$\begin{aligned} E_0\left[n\left(\vec{r}\right)\right] &= \min_{\Psi \to n} \langle \Psi | \hat{H}_K + \hat{H}_{\mathrm{e-e}} + \hat{H}_{\mathrm{e-n}} | \Psi \rangle & (49) \\ &= \underbrace{\min_{\Psi \to n} \langle \Psi | \hat{H}_K + \hat{H}_{\mathrm{e-e}} | \Psi \rangle}_{=:F[n]} + \int \mathrm{d}^3 r\, n\left(\vec{r}\right) v\left(\vec{r}\right) & (50) \end{aligned}$$

with HK density functional $F[n]$. We then minimize E_0 with a variational method using Lagrange multipliers to obtain the Kohn–Sham equations [360],

$$-\frac{\nabla^2}{2}\psi_j(\vec{r}) + v_{\mathrm{eff}}(\vec{r})\psi_j(\vec{r}) = \epsilon_j \psi_j(\vec{r}). \tag{51}$$

The effective potential $v_{\mathrm{eff}}(\vec{r})$ is defined as the sum of the external potential (the interaction with the nuclei) $v_{\mathrm{ext}}(\vec{r})$, the Coulomb interaction among the electrons $v_{\mathrm{H}}(\vec{r})$, and the exchange-correlation potential $v_{\mathrm{xc}}(\vec{r})$:

$$v_{\mathrm{eff}}(\vec{r}) = v_{\mathrm{ext}}(\vec{r}) + v_{\mathrm{H}}(\vec{r}) + v_{\mathrm{xc}}(\vec{r}). \tag{52}$$

The exchange-correlation potential takes into account nonclassical effects such as the Pauli principle and the electron correlation. This functional, which must be parameterized, determines the quality of the result. DFT is system-specific and does not reach the desired chemical accuracy. Similarly to the HF method, it does not take into account direct electron correlation, although this can be compensated for by the functional. Additionally, this method has the disadvantage of self-interaction, which occurs when each electron interacts with the electron density generated by all the electrons, including itself.

Because DFT is based on the three-dimensional distribution of the electron density, it is possible to directly calculate it on a three-dimensional grid around the atoms and bonds that represent the electron density at different points in space. However, in practical implementations, DFT frequently uses auxiliary functions to discretize the electron density on the grid. This approach ensures precise numerical integration, computational efficiency, management of exchange-correlation functionals, and allows us to apply linear scaling methods for larger systems. Consequently, DFT uses basis set functions similar to those in HF theory.

3.2.4. Post HF methods. Since neither HF nor DFT fully take into account the electron-electron correlation, the repulsive force between the electrons must be added *a posteriori*. Post-HF methods are more accurate than HF, but at greater computational cost. For example, the full configuration interaction (FCI) method considers all possible electron configurations (excitations) within the chosen basis set and gives the exact solution of the Schrödinger equation within a basis set expansion [361]. The number of all possible occupancy combinations scales as $O(N^M)$ with N electrons and M basis functions. It can only be performed on classical computers for small systems. Another example is the Møller–Plesset perturbation theory [362] which takes the core terms $\hat{H}_{\mathrm{K}} + \hat{H}_{\mathrm{e-n}}$ as an unperturbed Hamiltonian and treats the electron repulsion terms $\hat{H}_{\mathrm{e-e}}$ as a perturbation. Applying the Rayleigh–Schrödinger perturbation theory gives an approximate correction of the correlation energy to the HF energy. In comparison, coupled cluster techniques [363] apply an exponential cluster operator $\hat{T} = \sum_k \hat{T}_k$ containing k-fold excitation operators $\hat{T}_k$ to the HF wave function $|\Psi_0\rangle$. We obtain the exact solution $e^{\hat{T}}|\Psi_0\rangle$ through perturbation theory. These methods can be used for larger systems in their truncated form. For example, instead of the FCI approach, we could consider a single excitation to reduce the computational cost. However, this will also decrease the accuracy of the final energy. Another approach is to reduce the number of electrons and orbitals considered in the CI expansion by selecting a complete active space (CAS) where all excitations are considered. This approach relies on the premise that chemical reactions primarily involve higher-energy electrons. Although we may not obtain the exact total energy, energy differences (such as those between reactants and products) should account for the electron correlation.

The discussion above focuses on the simulation of molecules. The same concepts apply when simulating crystalline materials when the basis set consists of plane waves [355] or Bloch functions built from local orbitals [364]. Crystalline orbitals, which are the periodic equivalent of molecular orbitals, are delocalized over the cell and, therefore, are difficult to use in a CAS approach. One method to solve this problem is to use the Wannier single-particle electron basis [365]. A detailed analysis of how these functions can be used in the context of quantum algorithms can be found in [366].

3.2.5. Quantum algorithms for computational chemistry.

Quantum computing holds significant promise for enhancing two primary categories of chemistry challenges: post-HF methods and optimization/global minimum search tasks. Both categories deal with the challenges of combinatorial explosion. The former arises when encompassing all possible excitations, as seen in the FCI approach, while the latter emerges in scenarios where a multitude of configurations need to be evaluated, such as in determining the thermodynamic properties of complex materials or the optimum folded structure of particular proteins. Efforts have been made to map chemical problems to quantum computing-based simulations [337, 367]; understand the possible benefits of using quantum hardware [368]; and explore the possibility of QML [369]. It is worth noting that the FCI method, while being a significant improvement over HF or DFT for a given basis set, is still constrained in its accuracy by the choice of basis set.

An additional challenge when simulating electrons arises from their fermionic nature (half-integer spin). The need for their wavefunction to be antisymmetric introduces non-locality, affecting the mapping to qubits. The mapping process starts from the second quantized form of the wavefunction, where the computational basis state is represented by a Slater determinant in Fock space. The most widely used methods to map it to a qubit system are the Jordan–Wigner [370] and Bravyi–Kitaev [371] mappings [372]. Both methods ensure a one-to-one correspondence between electrons and qubits. Therefore, the non-locality of the wavefunction manifests itself in the operators, resulting in long strings of spin operators that increase gate depth and potentially degrade performance on NISQ devices. Although the Bravyi–Kitaev method aims to reduce the length of these operators, it is more challenging to implement and yields more complex quantum circuits, which may be harder to optimize. Recent and innovative mapping techniques, as discussed in [373, 374], achieve circuit lengths that scale logarithmically with the number of fermions in the system.

One promising research direction is to improve existing quantum methods for estimating electronic ground-state energies, such as the QPE [73] (section 2.3.2) and VQEs [147] (section 2.5) methods. The key difference between VQE and QPE-based approaches is that QPE requires encoding the full Hamiltonian into the unitary gates for the energy read-out method. Hence, QPE requires a large number of qubits and quantum gates [375, 376]. This complicated procedure is not needed in VQE. In addition, while QPE assumes the ansatz has a reasonable overlap with the exact ground state, VQE aims to approximate the ground state by variationally optimizing a set of parameters.

Ansätze widely used in quantum chemistry problems include the UCC ansatz [167, 377], for calculating the ground-state energy of a fermionic molecular Hamiltonian. In the UCC method, we expand a reference wavefunction on the basis of molecular orbitals mapped to the qubit basis in a quantum circuit. Then, we apply a parameterized unitary operator to the circuit, which produces a trial wavefunction. This procedure encodes complicated electronic states in a small number of qubits.

Variants of the UCC ansatz can reduce the circuit depth by compiling the fermionic operators more efficiently [378–380]. The variational Hamiltonian ansatz prepares a trial ground state for a given Hamiltonian by Trotterizing an adiabatic state preparation process [381, 382]. The variable structure ansatz optimizes the circuit structure itself by adding or removing unnecessary circuit elements [383–385]. Hybrid ansätze offload some of the complexity from the quantum ansatz to a classical device [386, 387]. They are especially useful for exploiting the classical simulability of the free-fermion dynamics.

The adaptive derivative-assembled pseudo-Trotter VQE (ADAPT-VQE) approach [383, 384, 388] is another promising way to construct accurate wavefunctions and accurately predict the ground state properties of chemical systems [389, 390]. ADAPT-VQE follows a dynamic strategy for selecting and refining the ansatz. It systematically expands the electronic wavefunction to include the operators that contribute significantly to the energy. This makes it well suited for strongly correlated electronic systems. ADAPT-VQE produces hardware-friendly circuits with few entangling gates, but the present technological limitations make it necessary to consider variants of ADAPT-VQE to enhance its performance.

Since quantum circuits can be simulated by tensor networks [391], it is also possible to combine tensor network techniques with a quantum ansatz [392, 393]. Alternatively, the deep VQE divides the system into smaller subsystems and sequentially solves each subsystem and the interaction between them [394].

So far, gate-based quantum hardware has been the technology of choice for performing post-HF simulations. On NISQ devices, FCI is suitable for small molecules such as H_2 and LiH [338], while, for larger chemical systems, an embedding scheme is generally preferred. Embedding techniques are well-established in classical computing for quantum chemical systems, see for example [395] for an overview of common approaches. The model system is typically divided into two regions, an active region that is treated at a high level of theory, using DFT or methods beyond DFT, embedded within an environment or inactive (frozen) region. Different schemes are available for embedding via systems of point charges, molecular mechanical models [395], electronic density [396], a density matrix [397] or Green's function-based method [398]. This approach has high potential to be used with quantum computing systems, where quantum algorithms are applied to the active region in the embedding model. For example, Gao *et al* [399] selected an orbital active space derived from an initial HF wavefunction to study the lithium superoxide dimer rearrangement. The method incorporates relevant orbitals present in reactants, products, and those involved in the transition state. In another instance, Gujarati *et al* [400] chose active space orbitals from DFT orbitals to study the water splitting reaction on magnesium surfaces.

Performing calculations of the electronic excited-state properties of molecules is another challenging area of research [401]. This is relevant when we want to simulate spectroscopic experiments, non-equilibrium dynamics, light-matter interactions and other excited-state based phenomena. Accurate simulations of these phenomena depend on precise calculations of the ground and excited states for the system, specifically the eigenvalues of the Hamiltonian operator. As discussed above, the VQE can achieve this with high precision [402]. Furthermore, the quantum subspace expansion method [403], which is an extension of the VQE, allows for the definition of a subspace of low-energy excited states once the ground state of the system is determined.

Quantum computers hold significant potential for the simulation of dynamic processes [404] such as chemical reactions, the study of transition states, and molecular dynamics. The most widely used approach relies on the Suzuki–Trotter decomposition [307, 405] to break down the time evolution into small steps that can be implemented on quantum computers. This approach requires deep circuits and is therefore limited by the size of current NISQ devices. A different approach [406], based on the VQE, results in a significant reduction in the gate count, but requires a large number of measurements, which are only possible for small systems.

3.2.6. Quantum optimization for chemical systems.

Quantum annealing (section 2.4) has been demonstrated to be a valuable tool for solving optimization problems in chemistry, biochemistry [407] and materials science [225]. Given a set of variables (for example, atom position in a solid solution lattice or bond angle in protein folding), the aim is to find the configurations of those variables that correspond to the lowest energy. The configuration space to explore is proportional to $\binom{n}{k}$ with n variables and k possible values that each variable can take. This configurational space rapidly becomes impossible to sample using classical methods for large values of n. Quantum annealing can offer an alternative means of finding the lowest energy configurations by exploiting quantum superposition and tunnelling (see Camino *et al*'s [225] tutorial for more details).

Quantum annealers have also been used to solve the electronic Hamiltonian eigenvalue-eigenvector problem [408] and to calculate excited electronic states [409] for biatomic and triatomic molecules. In industry, researchers at Volkswagen used D-Wave to calculate the ground state energies of H_2 and LiH as a proof-of-principle [410], where their mixed discrete-continuous optimization algorithm found the lowest eigenstate of the qubit coupled cluster method and used a quantum annealer to solve the discrete part of the problem [411]. Menten AI used D-Wave to address the rotamer optimization problem for protein design without major simplifications or a decrease in accuracy [412]. GlaxoSmithKline studied the mRNA codon optimization problem to help drug discovery [413] using quantum annealers. Fujitsu developed a quantum-inspired annealing machine to screen chemicals for materials discovery [414].

Optimization can also be performed on gate-based hardware when the problem is formulated as a QAOA problem (section 2.5.1). For example, the design of deuterated molecules for OLED applications has been studied using an approach that combines DFT-derived energies and VQE [415].

3.3. Important applications in industry

Quantum simulation will potentially solve many industrial problems and bring about significant societal and environmental benefits. We give two examples in this section. Researchers are using the Fermi–Hubbard model as an analogue quantum simulator for high-temperature superconductors (HTS), specifically copper oxide compounds. On the other hand, digital quantum simulation may help us understand how to produce ammonia fertilizers using less energy intensive processes.

3.3.1. High-temperature superconductivity.

Theoretical high-energy and condensed matter physics share common fundamental concepts such as symmetry breaking, renormalization group, and Feynman diagrams. Interesting phenomena occur in strongly correlated electronic systems where several physical interactions (spin, charge, lattice, and/or orbital) are

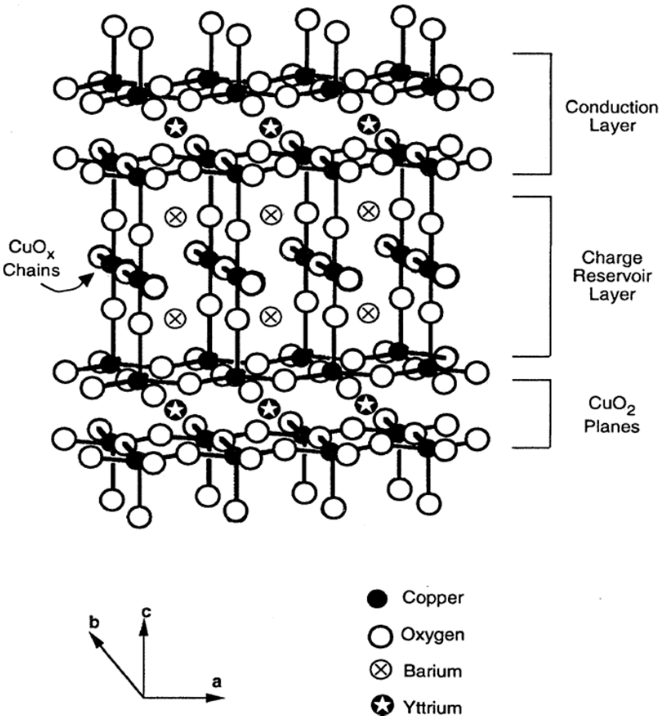


Figure 15. Crystal structure of cuprate superconductor $YBa_2Cu_3O_{6+x}$. Reprinted (figure) with permission from [424], Copyright (1994) by the American Physical Society.

simultaneously active. In particular, the discovery of HTS in the 1980s [416, 417] launched decades of diligent efforts to understand and use these compounds [418, 419].

The Bardeen–Cooper–Schrieffer (BCS) theory [420, 421] introduced the idea of electron–hole Cooper pairs to explain conventional low temperature superconductivity. However, the pairing mechanism in unconventional (high temperature) superconductors is still unknown. These materials do not conform to conventional BCS theory or its extensions. Conventional superconductors are used in many application areas such as MRI machines and high speed trains, but they can only operate at relatively low temperatures. This severely limits their wider use. One of the holy grails of materials science is to develop superconducting materials that operate at room temperature. This would revolutionize many technologies [419]. In particular, room-temperature superconductors would decrease the heat wasted from electronic devices and allow them to run more efficiently. On a larger scale, HTS can help achieve the International Energy Agency's roadmap to carbon-free economies [422] via nuclear fusion-generated electricity [423].

Researchers have focused on the two-dimensional Hubbard model as it is believed to capture the important behavior of HTS [424, 425], specifically the cuprate superconductors (a popular HTS) in the copper–oxygen planes ('CuO_2 planes' in figure 15). Despite the Hubbard model's apparent simplicity, its theoretical properties are far from fully understood. It is difficult to solve accurately as the model exhibits competing orders in its phase diagram where it is most relevant to cuprates. The Hubbard model is also widely used to benchmark numerical methods for strongly correlated systems [426]. Quantum efforts include experimental work building analogue quantum simulators of the model using quantum dots [427, 428] and ultracold atoms in optical lattices [429, 430]. Digital quantum approaches include VQE variants [217, 431, 432], while Google simulated the Fermi–Hubbard model using digital superconducting quantum processors [433].

A state-of-the-art circuit-model algorithm for simulating the two-dimensional Hubbard time dynamics on an 8×8 lattice requires roughly 10^7 Toffoli (CNOT) gates [434]. This includes the overhead for fault tolerance, given the gate fidelities of current and near-term hardware. A significant contribution to the gate count comes from the phase estimation procedure. These large gate counts are corroborated by Clinton *et al* [435], who estimate the resources to simulate the time dynamics on a 5×5 square lattice to $\leqslant 10\%$ accuracy require 50 qubits and 1243 586 standard two-qubit gates. This is an optimistic estimate that assumes the effects of decoherence and errors in the circuit can be neglected; the source of inaccuracy is the approximations in the gate sequences.

There is much work to be done before we have a complete microscopic theory of HTS [418, 419]. The combination of theory, simulation, materials synthesis and experiment has been crucial to progress in the last two decades. For example, the breakthrough discovery of hydrogen-rich superconductors (super-hydrides) [436] may not have happened for another century [437] had there not been significant advances in simulation and algorithms for chemical structure prediction [438]. Similarly, numerical work to understand the two-dimensional doped Hubbard model has relied on the lastest advances in Monte Carlo methods [439, 440]. Experiments with ultracold atoms in optical lattices have also proved promising [429, 441, 442] due to advances in quantum control with quantum microscopes and other techniques [443]. Despite the lack of a clear timeline on when we can expect a fundamental theory for HTS, researchers remain optimistic [437].

3.3.2. Fertilizers. Another strong motivation for quantum simulation is to develop new materials and processes to make significant environmental contributions. For example, the Haber–Bosch process for producing ammonia (NH_3) fertilizers is one of the world's most CO_2-intensive chemical processes, consuming up to 2% of global energy output [444] and 3%–5% of all natural gas generated globally [445].

The Haber–Bosch process turns nitrogen in the air into ammonia-based fertilizer for crops, which in turn provide food for about 40% of the world's population. The natural biological process, nitrogen fixation, is a much more efficient process: microorganisms that contain the biological enzyme *nitrogenase* convert atmospheric dinitrogen (N_2) to ammonia under ambient conditions. Despite almost a century of research, the reaction mechanism is still unexplained [446, 447]. Understanding how this enzyme works would be an important step towards replacing the Haber–Bosch process and creating less energy-intensive synthetic fertilizers,

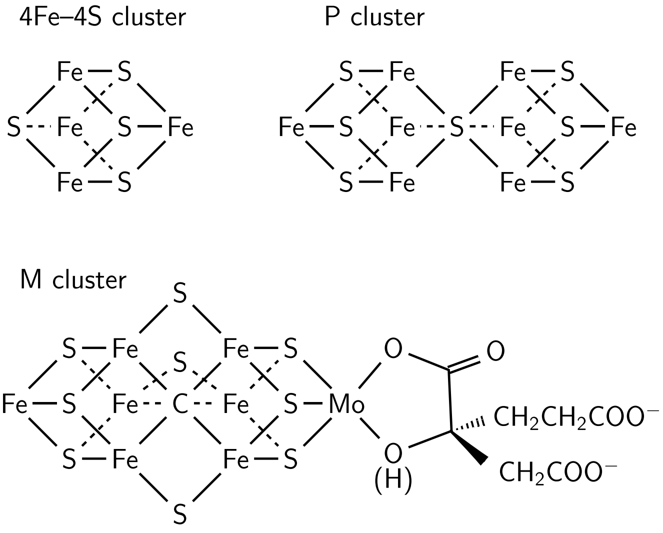


Figure 16. Nitrogenase enzymes convert atmospheric dinitrogen into ammonia. The process involves transferring multiple electrons and protons to dinitrogen and uses multiple metalloclusters found in the nitrogenase enzyme, including the 4Fe–4S cluster (top left), P cluster (top right) and Fe-Mo cofactor M cluster (bottom).

The transition metal compounds within the enzyme potentially hold the answers. These are the '4Fe–4S' cluster (containing iron and sulfur), 'P cluster' and the iron molybdenum cofactor 'M cluster' (FeMoco, containing iron, molybdenum, carbon, hydrogen, and oxygen) [448] (figure 16). The FeMoco active space contains 54 electrons in 108 spin orbitals. Proposed computational models of FeMoco are beyond the reach of current classical methods but are possible with small error-corrected quantum computers.

As an initial study to develop simulation methods, Reiher *et al* [64] used a Trotterization approach to simulate the active-space model of FeMoCo. Their algorithm had T-gate complexity scaling as approximately $O(N^2S/\epsilon^{3/2})$, with Hamiltonian sparsity S and error rate ϵ, and required over 10^{14} T gates. This corresponds to roughly 10^8 physical qubits if implemented in the surface code [449] with gates at 10^{-3} error rate. Their work focused on counting and reducing the required number of T gates. In practical error-correcting codes such as the surface code, these gates require significantly more time to implement than any other gate and require a large number of physical qubits for implementation. This work was followed by Low and Chuang's [168] qubitization approach. They reduced the gate cost estimates by several orders of magnitude to roughly four days on a fault-tolerant quantum computer equipped with 4×10^6 physical qubits [450].

Qubitization is particularly promising for reducing the computational costs of digital quantum simulation [376, 451], which largely depends on how the input Hamiltonians are accessed by quantum computers. While simulations typically use sparse matrices [452] or LCU [453, 454], Low and Chuang's standard-form encoding [168] is much more general and includes sparse matrices and LCU as special cases. Qubitization uses the fact that whenever the encoded Hamiltonian $\hat{H}$ contains an eigenvalue λ, the standard-form encoding operation contains a 2×2 block

$$\begin{bmatrix} \lambda & -\sqrt{1-\lambda^2} \\ \pm\sqrt{1-\lambda^2} & \pm\lambda \end{bmatrix}$$

on the diagonal, with respect to a basis determined jointly by $\hat{H}$ eigenstates and the encoding [168]. Different λ values produce different 2×2 blocks, which are analogous to single-qubit rotations or reflections, hence the name 'qubitization'. This spectral relation builds on earlier work such as Szegedy's quantum walk [455, 456] and quantum Merlin–Arthur amplification [457], but the formulation by Low and Chuang [168] is more suitable for quantum simulation.

Google and collaborators [458] recently attempted to calculate the energy states of the nitrogenase 4Fe–4S cluster on their Sycamore quantum computer. However, simulations containing over 300 gates were overwhelmed by noise. The authors concluded that quantum circuit based quantum computers may not provide much advantage over classical computers until they incorporate noise reduction and/or full QEC methods. These results are both exciting and daunting: quantum computational chemistry and materials science has made great progress over the last decades, but [458] also shows how much work is still ahead.

4. Quantum algorithms for classical simulation

High-performance computing is used for modelling large-scale systems in many socially and economically important areas, including weather forecasting, cosmology, plasmas, coastal engineering, real time traffic management, and many other applications. These models often have analytically intractable non-linear partial differential equations (NLPDEs) at their core, with a large number of variables in two or more dimensions. Classical numerical solvers for NLPDEs have been an active area of research for decades [459–461]. Finite difference, volume, and element methods; spectral Galerkin methods; and neural networks are commonly used. The latter are particularly suitable for (approximately) solving high-dimensional PDEs [462, 463]. Given the large amounts of HPC resources used to solve NLPDEs, improved algorithms are highly important and the subject of significant research investment. For example, NASA's Quantum Artificial Intelligence Laboratory aims to show how quantum algorithms may dramatically improve the agency's computational problem-solving ability [464–466]. The HHL algorithm [103] (section 2.3.7) can be used in regimes where it is sufficient to take a linear approximation of the NLPDEs. For many applications, it is crucial to model the nonlinear effects more accurately. In this section, we focus on one of the most common NLPDEs, the Navier–Stokes (NS) equation for computational fluid dynamics (CFD), but the quantum algorithms can be readily adapted to other NLPDEs.

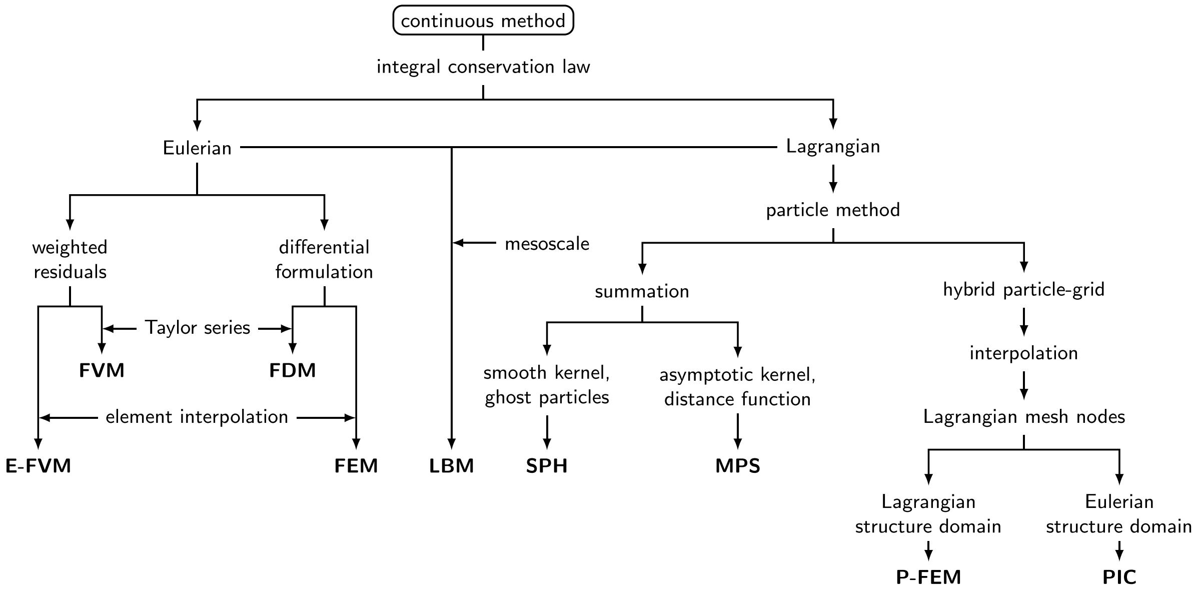


Figure 17. Examples of numerical methods for simulating CFD problems: finite volume method (FVM), element-based finite volume method (E-FVM), finite element method (FEM), finite difference method (FDM), particle-in-cell (PIC), moving-particle semi-implicit method (MPS), particle-based FEM (P-FEM), smooth particle hydrodynamics (SPH), and lattice Boltzmann (LBM).

4.1. Solving the Navier-Stokes (NS) equations

In CFD, the aim is to solve the NS equations which describe a flow of pure gas or liquid (single- or multi-phase flows)

$$\frac{\partial \vec{u}}{\partial t} = \underbrace{-\vec{u}\cdot\nabla\vec{u}}_{\text{advection}} - \nabla p + \underbrace{\frac{1}{\mathcal{R}}\nabla^2\vec{u}}_{\text{viscous diffusion}} \tag{53}$$

with continuity equation

$$\nabla\cdot\vec{u}\begin{cases} = 0, & \text{incompressible flow,} \\ \neq 0, & \text{compressible flow.}\end{cases} \tag{54}$$

Velocity $\vec{u}(\vec{r},t)$ and pressure $p(\vec{r},t)$ fields are functions of time t and spatial coordinates in continuous space $\vec{r}$. The Reynolds number $\mathcal{R}$ describes the ratio of inertial force to viscous force (viscous diffusion).

The NS equations consider the contribution of all forces acting on an infinitesimal element of volume and its surface. Given a certain mass of fluid in a region of space, two types of forces act on it: volume (forces outside the region) and surface (internal forces arising from fluid interactions via the boundary surfaces). The NS equations are a system of three balance equations (PDEs) of continuum mechanics which describe a linear viscous fluid. Under this umbrella, Stokes' law (kinematic balance) refers to the specific case of force on a moving sphere in fluid, and Fourier's law (energy balance) is a fundamental law of the material.

A straightforward approach to solving the NS equations is direct numerical simulation (DNS) [467]. DNS directly discretizes the NS equations and relies on using a mesh size that provides resolution at all scales of turbulent motion, including Kolmogorov length scales. Thus, there is no need for any subgrid modelling in order to capture turbulent flow dynamics, and an exhaustive description of the fluid is available throughout the domain. However, this understanding comes at high computational cost due to the requirement of a sufficiently fine mesh. DNS is too costly to use in most industrial problems and instead is generally relegated to more fundamental research. Hence, this fuels the need for alternatives, better suited to real world applications.

Recent progress in machine learning techniques includes methods to replace some of the expensive time evolution in fluid simulation. Physics-informed neural networks (PINNs) [468, 469] are trained to solve supervised learning tasks constrained by the PDEs' physical laws, while respecting the conservation laws in fluid mechanics [470, 471]. This allows us to develop a more general predictive model for different problems. See [472–474] for a pedagogical introduction.

Figure 17 summarizes some methods for solving the NS equations which form the basis of many commercial software packages [475]. These include mesh-based methods such as the FDMs, FVMs and FEMs. Mesh-based models divide a continuum domain into smaller subdomains where the subdomain size is varied depending on the level of detail required around that region. Despite their success in science and engineering problems, mesh-based methods encounter difficulties at free surfaces, deformed boundaries, moving interfaces, and extremely large deformation and crack propagation. It is time-consuming and expensive to generate quality meshes for complicated geometries. For example, in FDM, irregular or complex geometries usually require additional transformations (e.g. mesh-rezoning [476]) that are more expensive than directly solving the problem itself, and may introduce numerical inaccuracies.

In contrast to more established and mostly mesh-based codes, the smoothed particle hydrodynamics (SPH) method offers a viable alternative, especially for flows with a free

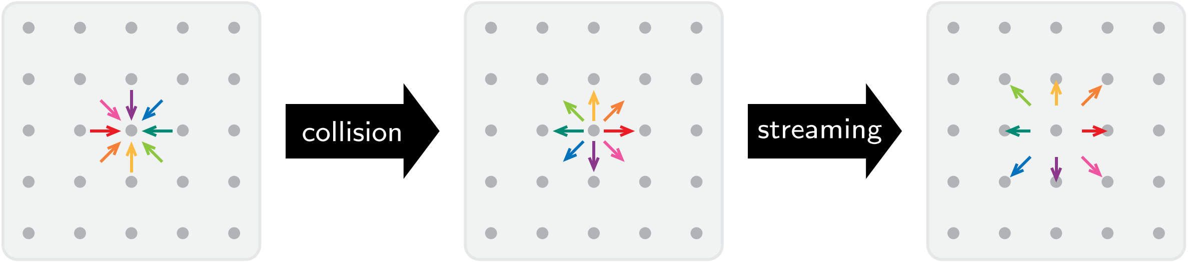


Figure 18. Schematics of two-dimensional *D2Q9* LB model showing nodes (grey), collision and streaming processes under temporal evolution. Colors represent direction vectors $\vec{e}_k$ where k corresponds to different directions in *D2Q9*.

surface [477]. SPH is a purely Lagrangian approach that discretizes a continuum using a set of point particles. It has become popular in the last few decades as a way to simulate a wide range of applications, e.g. in engineering [478] and astrophysics [479, 480]. Another relatively new approach, the lattice-Boltzmann (LB) method [481, 482], combines both mesh- and particle-based methods. LB is an advance on lattice gas automata that overcomes the convergence problems of directly using particles on grids. Both SPH and LB have interesting potential for quantum algorithms or quantum–classical hybrid enhancements. We discuss both in turn, first summarizing the classical method, then explaining the proposed quantum algorithms.

4.2. Lattice Boltzmann (LB) method

LB is a mesoscale approach based on the Boltzmann equation. It naturally bridges microscopic phenomena with the continuum macroscopic equations and accommodates a wide variety of boundary conditions [481, 482]. LB has good scalability for solving PDEs due to its algorithmic locality, hence is inherently suited for large-scale parallelization on HPC systems, including on GPUs [483–485].

The LB method considers a typical volume element of fluid that contains a collection of particles [486]. These are characterized by a particle velocity distribution function for each fluid component at each grid point. At discrete time steps, fluid particles can collide with each other as they move, possibly under applied forces. The particle collision behavior is designed such that their time-average motion is consistent with the NS equation. The fluid is treated as a group of particles that have only mass and no volume. Particles flow in several directions of the lattice and collide with the particles around them. LB uses the collective motion of microscopic particles to describe the macroscopic parameters such as velocity, pressure, and temperature according to kinetic theory.

We use the Lattice Bhatnagar–Gross–Krook notation to describe lattice structures: the *DnQb* classification indicates an n-dimensional space where each particle collides with b surrounding particles (including itself). For example, the *D2Q9* lattice in two-dimensional space contains a particle labelled c_0 which collides with the surrounding particles in eight directions (figure 18). Particle movement is described in two steps: collision and streaming. In the collision step, particles collide with the opposite particles along each axis, changing the velocity. In the streaming step, particles move in the velocity direction to the neighboring lattice. The evolution equation is

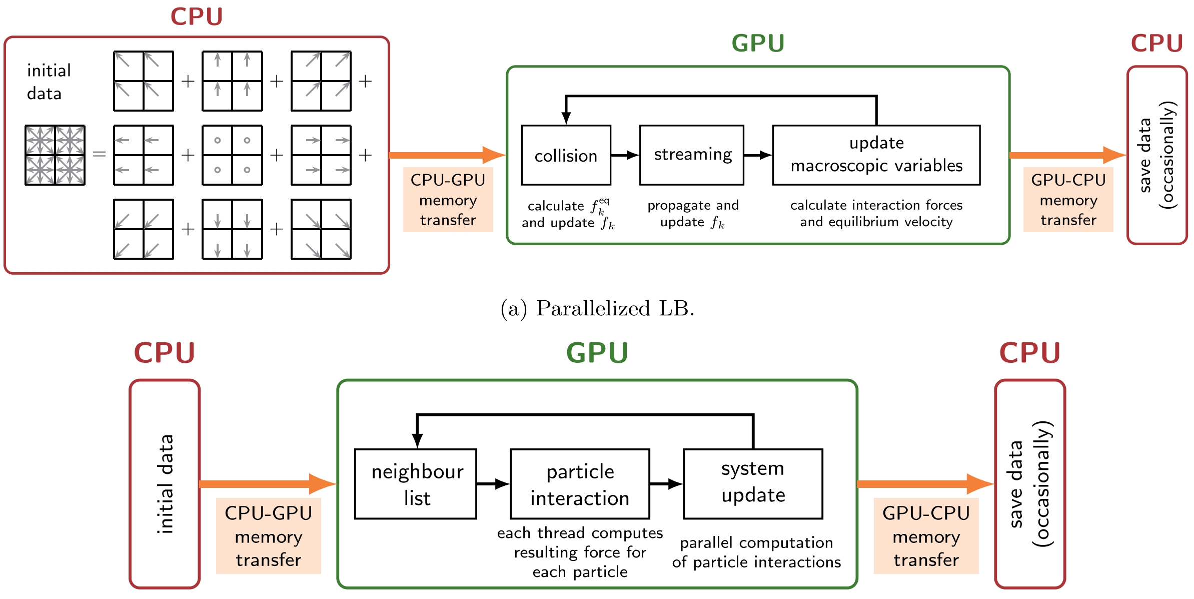
$$f_k(\vec{x}+\vec{e}_k, t+\Delta t) - f_k(\vec{x},t) = \Omega(f_k^{\text{eq}}(\vec{x},t) - f_k(\vec{x},t)) \qquad (55)$$

with collision term Ω, fluid distribution $f_k(\vec{x},t)$ at point $\vec{x}$ at time t, and equilibrium distribution f_k^{eq}. At the next timestep $t+\Delta t$, the fluid would be at point $\vec{x}+\vec{e}_k$. The direction vector is $\vec{e}_k$ where k represents the nine different directions in *D2Q9*. Each node is connected to its nearest neighbors with different colored vectors, denoted as $\vec{e}_k$. (Not shown in the figure is one $\vec{e}_k$ pointing to the node itself.) Every f_k moves along its $\vec{e}_k$ vector to its neighbor and replaces the neighbor's distribution $f_{k'}$, except the one pointing to the node itself. In practice, the *D2Q9* lattice requires up to several million nodes to generate an accurate flow field.

A common theme in LB applications is their suitability for parallel computing [487, 488]. This is primarily due to the intrinsic locality of the method. For each time step, both the collision and streaming operators only require communication between neighboring cells at most. Recent research has focused on parallelizing and optimizing the computation using heterogeneous acceleration devices such as GPUs. The idea is to divide the LB lattice and group the distributions f_k into arrays according to their velocity vectors $\mathbf{e}_k$ (figure 19(a)). The first iteration starts with the distributions as inputs to calculate the density and velocity outputs. These quantities allow us to find the equilibrium distributions. New distributions are then computed from the input distributions and the equilibrium distributions according to the collision and the streaming operations. Finally, the boundary and outflow conditions are used to update the distributions. The updated distributions are then used as inputs for the next simulation step. For further details, see [489]. Example open-source software platforms include OpenLB [490] and Palabos [491].

4.3. Quantum lattice Boltzmann (QLB)

Early exploration of quantum methods for CFD used quantum lattice gas models [492] and 'type II quantum computers' [493]. Both contain several quantum nodes connected by classical channels that carry bits instead of qubits [494], while the latter are mathematically equivalent to a classical LB formulation [495]. These lattice models are susceptible to noise, non-isotropic advection and violation of Galilean invariance. Recently, a revised quantum algorithm for lattice gas



(a) Parallelized LB.

(b) Parallelized SPH.

Figure 19. Sketch of parallelized LB and SPH numerical processes with CPU and GPU.

automata [496] managed to eliminate the need for repeated measurements at each time step. However, the size of the required quantum register scales linearly with the number of lattice sites. This makes it difficult to scale to realistic use cases.

There is a direct correspondence between the LB streaming step and quantum walks [497]. Todorova and Steijl [498] build on the latter and present the first fully quantum (as opposed to hybrid classical-quantum) LB method. The authors consider the simplified case of a collisionless Boltzmann system which reduces to

$$\frac{\partial F(\vec{x},\vec{c};t)}{\partial t} + \vec{c}\frac{\partial F(\vec{x},\vec{c};t)}{\partial \vec{x}} = 0 \tag{56}$$

with single-particle distribution function $F(\vec{x},\vec{c};t)$ defined in physical $\vec{x}$ and velocity $\vec{c}$ spaces. Classically, the discrete velocity method discretizes the solution state space into potentially three spatial and flow dimensions in (56), leading to a six dimensional solution space which can be computationally expensive. Therefore, a key stepping stone for any appreciable quantum advantage is to represent this higher dimension space as a quantum state using a limited number of qubits. As shown by Todorova and Steijl [498], this can provide an exponential reduction in memory, particularly for the extension to fluid mixtures. Classically, where doubling the number of fluid components would require doubling the memory, this was shown to be achievable with one extra qubit. Note also that the assumption of a collisionless system essentially decouples (56) for each fluid, a condition which is not true for real fluids.

A primary obstacle to quantum LB algorithms is the need to take measurements during the simulation. This results in costly initialization routines when re-preparing the quantum state. Todorova and Steijl [498] address this issue by proposing that the purpose of their QLB is different to conventional CFD. The idea is to forego acquiring the complete flow field, i.e. the algorithm does not obtain a complete picture of the classical state. Instead, it efficiently obtains specific information such as particle number densities and concentrations for multi-fluid configurations. Thus, the algorithm allows an uninterrupted temporal iteration with measurement postponed until the end, at the cost of reducing the obtainable information. Similar approaches from quantum simulation and quantum chemistry are discussed in [333, 499] or techniques to obtain a single amplitude in [81].

Including the collision operator in LB adds a nonlinear complexity, but is essential for using the method as a general PDE solver. Being able to choose a particular collision operator provides flexibility in which equation is actually being solved at the continuum level. One approach relies on Carleman linearization where an equivalent linear system in a higher dimensional space essentially replaces the nonlinear system. Recent work [500] explored this in the context of nearly incompressible LB to find a quantum formulation for the collision term. Building on their findings, the authors propose a fully unitary streaming and collision operator [501]. However, their collision step has a sub-optimal dependence on the number of time steps. This makes it unclear whether it is possible to achieve a practical quantum advantage, particularly at large Reynolds numbers.

Alternatively, Budinski [502] formulates a fully quantum algorithm for the complete LB, i.e. including non-unitary collision operators using the standard form encoding approach [168] and a quantum walk for the propagation step. Having initially applied this to an advection-diffusion equation, it was later extended to the streamfunction-vorticity formulation of the NS [503]. However, because the collision operator is retained in the equilibrium distribution function, the resulting nonlinearity restricts the algorithm to a single time-step before the state must be reinitialized. It has been shown [504] that the conventional amplitude and basis encodings do not allow unitary operators for both streaming and collision. This means future work needs to focus on alternative encodings, or, on ways to implement nonunitary operations as part of a quantum algorithm.

4.4. Smoothed particle hydrodynamics (SPH)

SPH is a particle-based method with no mesh or grid. The particles move according to classical mechanics, and the fluid velocity is obtained as an average over the particle distributions. Like LB, it is a more general method than just for solving Navier Stokes equations. Mathematically, the SPH method expresses a function in terms of its values at locations of the virtual SPH particles. The integral interpolant of any function $A(\mathrm{r})$ is an integral $A(\mathbf{r}) = \int_\Gamma A(\mathbf{r'})W(\mathbf{r}-\mathbf{r'},h)\mathrm{d}\mathbf{r'}$ over the entire space Γ for any point $\mathbf{r}$ in space and smoothing kernel W with width h. This can be approximated by a summation interpolant,

$$A_S(\vec{r}) = \sum_i m_i \frac{A(\vec{r}_i)}{\rho_i} W(\vec{r}-\vec{r}_i,h) \tag{57}$$

that sums over the set of all SPH particles $\{i\}$. Each particle i has mass m_i, position $\vec{r}_i$, density ρ_i, and velocity $\vec{v}_i$. Hence, we can construct a differentiable interpolant of a function from its values at the particle level (interpolation points) by using a differentiable kernel [477].

The general SPH workflow contains three main stages that are repeated at each time step. First, the algorithm creates a neighbor list. Then it calculates the particle interactions for momentum and continuity conservation equations. This involves calculating the smoothing length and SPH kernels. The final step is time integration (or system update). The SPH particle distribution follows the mass density of the fluid while their evolution relies on a weighted interpolation over nearest neighboring particles. This has several implications. The SPH kernel smoothes the physical properties within the range of interpolation (figure 20). This is characterized by the smoothing length h which also determines the local spatial resolution. The identity of the neighboring particles change as the simulated system evolves. There is no computationally efficient method to predict which particles will be neighbors over time, hence we must identify the neighbors at each timestep.

Parallelization strategies for SPH are strikingly different from those of mesh-based methods. The computational domain is divided into a grid of cells where each SPH particle is assigned a cell. We build a list of its neighbors by searching for particles only in nearby cells. Dynamical neighbor lists require specialized methods for data packing and communicating. Particles migrating between adjacent domains can cause difficulties on memory management especially in distributed-memory architectures, as can large variations in particle density and domain size. The total number of particles can narrow the choice of hardware, as memory space is one of the main limitations of shared memory architectures. Using both shared and distributed memories, most SPH parallelization schemes are scalable [505, 506]. However, if communication latencies increase, scalability decreases rapidly [507].

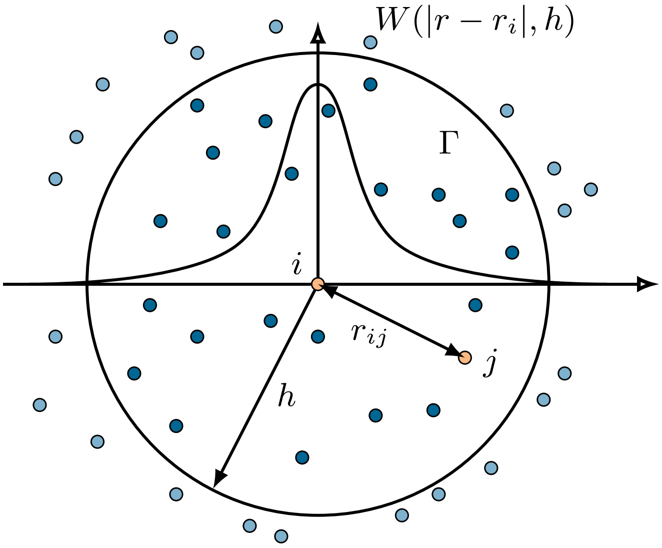


Figure 20. Schematics of two-dimensional SPH model with SPH particle i, its neighbor j, kernel W, smoothing length h and compact support domain Γ. Here, h is equal to radius of support domain (support length). Full support length can typically be between $2h$ and $4h$.

Diverse SPH applications demonstrate the versatility of SPH methods in multiple fields. SPHysics is an SPH code tailored for simulating free surface flows, which are challenging for other methods but effectively handled by SPH. DualSPHysics [508] runs on a hybrid architecture of CPUs and GPU accelerators for enhanced performance. The astrophysics community also extensively uses SPH codes, as evidenced by the popularity of SWIFT [509] and GADGET [510]. These tools have significantly contributed to advancements in cosmological and astrophysical research.

Figures 19(a) and (b) compare the typical workflow for combined CPU-GPU implementation of LB and SPH, illustrating the commonalities in the processes. We break down the algorithm into parts that are suitable for different types of classical hardware. This illustrates how to incorporate accelerators into large scale simulations. Extending this to incorporate quantum computers used as accelerators can hopefully provide further efficiencies.

4.5. Quantum SPH algorithm

Recent reviews [511, 512] discuss the computational bottlenecks and grand challenges we face before SPH becomes more widely used for practical problems (e.g. in engineering). The SPH method has traditionally been considered computationally expensive [511] due to two major factors: a large number of SPH particles are needed for good solution accuracy and time steps must be small enough to obey empirical stability criteria. One strategy to address these bottlenecks involves using quantum subroutines [513]. Below we list examples of how to substitute classical SPH procedures with quantum algorithms.

Rewriting the SPH interpolant (57) as an inner product of the form

$$\sum_i m_i \frac{A(\vec{r}_i)}{\rho_i} W(\vec{r}-\vec{r}_i,h) \to \langle mA(\vec{r})/\rho|W(\vec{r},h)\rangle \qquad (58)$$

requires efficiently loading the classical data (floating point numbers) into quantum processors using quantum encoding techniques. There are several possible methods to calculate the inner product: the swap test [514] or one of its variations [515]; the Bell-basis algorithm [516] which is efficient on NISQ devices; quantum mean estimation and support vector machines [278]; or quantum counting algorithm [87, 517].

The summation in (57) means that the algorithm requires a search algorithm to find all the neighboring particles inside the compact support domain Γ (figure 20). This is another major bottleneck that we can, in principle, address using Grover's search algorithm [14] implemented with a quantum walk (sections 2.3.4 and 2.3.6). This can be an effective search method when combined with existing SPH neighbor-list approaches, e.g. cell-linked or Verlet lists [518].

The timestepping procedure is subject to empirical stability criteria like the Courant–Friedrichs–Lewy convergence condition [519]. Physically, in the widely-used weakly-compressible [520] and incompressible [521] forms of SPH, the timestep is also limited by the speed of sound and maximum velocity respectively. This leads to timesteps of the order of $\leqslant 10^{-5}$ s [511], or typically one million time steps to simulate one second of physical time. Most three-dimensional applications require 10–100 million SPH particles. High computational costs have motivated research into timestepping procedures such as the Runge–Kutta–Chebyshev scheme [522]. This is where quantum algorithms may potentially provide an even greater speed up, since we do not need to read out the data every time step. A quantum algorithm that can evolve for many time steps between measurements has the best chance of providing real advantages.

4.6. Other quantum algorithms for fluids simulations

There are quantum algorithms that directly perform numerical integration for linear differential equations and provide some quantum advantages [523, 524]. Efforts to develop a quantum solver for the Poisson equation have applied VQAs (section 2.5) [525, 526] and HHL framework (section 2.3.7) [527, 528]. However, specialized quantum algorithms are less well developed, and methods for NLPDEs remain open [529]. NLPDE problems are some of the most computationally demanding calculations in CFD. The nonlinearities may be due to the physics (one or more of the following: convection, diffusion, forcing, turbulence source terms, reacting flows) or the numerics. Compared to linear problems, there have been few advances in developing quantum algorithms for NLPDEs due to difficulties in expressing nonlinearities with unitary gates. Up until recently, the only notable work was an HHL-based method that uses post-selection to implement the nonlinearity [530]. Unfortunately, it is an exponential-time algorithm that also needs an exponential number of copies of the initial state.

This calls for another way to linearize the nonlinear equations (e.g. with Carleman linearization [500, 531]). One approach is to discretize the NLPDEs then directly apply a quantum solver to the resulting nonlinear ordinary differential equation (ODE). This approach was adopted in [532, 533] using an ODE solver that had gone largely unnoticed until recently [534]. The method comes with stringent requirements on the underlying flow field and its rate of change. Despite this, a regime exists where there may be a quadratic speedup over classical random methods, and an exponential speedup over classical deterministic methods. Replacing the ODE solver with alternative formulations may also provide some benefits within the overall method.

While most discussions of quantum computing focus on when quantum computers will be able to consistently beat their classical counterparts, the reality is more likely to involve hybrid quantum–classical devices. For example, where possible, solve the non-linear parts of the calculation on conventional HPC, then send the linear part to a quantum processing unit which uses the HHL algorithm. However, if the classical part then requires consistent feedback from the quantum part, the cost of measurement and reloading input states may significantly reduce any quantum advantage. Some recent works include using a quantum algorithm as a predictor-corrector to accelerate the classical components of the algorithm without necessitating a complete solution readout [125]. Alternatively, the authors in [126, 535] discuss a framework for simulating simplified fluid dynamic configurations described by linear PDEs, using quantum linear equation solvers in conjunction with classical protocols. Instead of simply measuring out the complete velocity field, the authors propose a novel quantum postprocessing protocol for efficiently computing *in situ* nonlinear functions of the resulting flow field, namely the average viscous dissipation rate.

An advantage of PINNs for complex flow simulations is their robustness to parameter changes. For instance, in classical CFD altering the geometrical boundaries typically requires restarting the simulation in order to determine the new flow profile. However, by applying transfer learning methods [536], a PINN model trained on a slightly different geometry can be adapted for the new configuration without a complete restart. For PINNs to provide reliable outputs, it is imperative to have high expressibility and this is already known to be a key strength of quantum computers [180, 537]. Recent work [538] in the field includes a hybrid quantum PINN based on a quantum depth infused layer implemented using a VQA and coupled with transfer learning. The authors reported a 21% reduction in loss compared with its purely classical counterpart. Quantum PINNS have also shown promise in solving the PDEs of atmospheric simulation for weather and climate modelling [539].

More general VQA methods (see section 2.5) can solve NLPDEs on NISQ devices [540]. Lubasch *et al* [144] applied their method to the one-dimensional heat equation and Burger's equation. It is possible to solve the

Schrödinger–Poisson equation by combining this method with a subroutine to train a state to represent a nonlinear potential [541]. The reaction-diffusion equations can be solved similarly [542]. Further work presented a general version of this quantum method to treat a large class of nonlinearities and inhomogeneous terms [543], making the method applicable to most NLPDEs and multidimensional PDEs which are first-order in time, such as the NS equations. It can also be used to adapt many explicit and semi-implicit numerical schemes, such as the fourth-order Runge–Kutta algorithm.

The vortex-in-cell method is an example of how the QFT can be used to build a Poisson solver [544]. However, the main drawback is the assumption that multiplication with the wavenumbers can be done efficiently on classical hardware. There is also the added complexity of accommodating non-periodic boundary conditions with any Fourier-based CFD method, meaning its translation into a quantum setting may not be straightforward.

While there is an encouraging diversity of different quantum approaches, many existing demonstrations of quantum algorithms for CFD are limited to relatively simple flow cases. The effectiveness of these algorithms at scaling with the Reynolds number is either yet to be established or appears limited. While configurations involving low Reynolds numbers are valuable in areas like geophysics and biophysics, the study of turbulence at high Reynolds numbers is crucial for many disciplines. In this regime, a quantum-inspired algorithm using Schmidt decomposition combined with tensor network methods can successfully account for the interscale correlations in turbulence [545]. Even if useful quantum simulations of turbulence at scale are some way in the future, quantum-inspired algorithms like this can provide benefits to current classical simulations.

5. Interfacing with HPC

Compared to classical computers, quantum hardware operates on different intrinsic timescales (clock speeds), and uses different methods for data encoding—some of which are necessary to achieve a quantum advantage. This makes interfacing between classical and quantum hardware highly nontrivial. Despite the significant practical hurdles, using quantum computers as accelerators for HPC is the logical way to maximize overall computing power, and test beds offering coupled QPU and CPU configurations are being set up [546, 547]. In this section, we briefly review these interface issues, and the progress required to overcome them.

5.1. Clock speed mismatch

Every computation on a classical computer processor corresponds to a sequence of layers of logic gates. The time taken to execute one layer is the *clock cycle*. Modern CPUs contain several billion transistors operating at around 3–4 GHz or 1/3 nanosecond per cycle. When quantifying the speed of quantum computers, we can break down quantum algorithms into a sequence of quantum gate layers. However we cannot define the *quantum clock speed* as an inverse of the quantum gate layer time because of two major variables:

Table 4. Quantum hardware platforms ordered by decreasing gate speed (conservative estimates).

Platform	Two-qubit gate speed	Measurement time
Superconducting qubits [548]	10 MHz	660 ns
Photons (boson sampling) [549]	6 MHz	36 μs
Silicon spin qubits [550]	750 kHz	1.3 μs
Trapped ions [551, 552]	6 kHz	100 μs
Rydberg arrays [553]	170 Hz	6 ms
Photons (fusion-based) [554]	10 Hz	100 ms

(i) Gate choice. Quantum algorithms can be decomposed into elemental gates, but the choice of gates is flexible. Hence, the quantum gate duration can vary, even for different gate layers in the same hardware.

(ii) Quantum error correction. Quantum computers make many more errors than classical CPUs. Large-scale quantum computers will likely have a runtime dominated by QEC overheads. Each QEC method consists of different routines with different runtimes, that can be dominated by mid-circuit measurements, rather than gate times.

Table 4 shows typical gate operation speeds and measurement times for different hardware platforms. Note that these are physical gate speeds that error correction overheads will significantly modify. Slower, but higher quality, trapped ions or Rydberg atoms will need less error correction than more noisy, but faster, superconducting qubits. Fusion-based photonic platforms are slow because they have high overheads to generate the heralded resource states required for fully universal quantum computing [239, 555]. In general, state preparation can be done ‘offline’ in parallel with other operations, to avoid this step slowing down the overall compute time. This works well for trapped ions or atoms [556], but for photonic fusion gates the overheads are extreme and dominate the scaling. Photons can also be used for boson sampling [295, 557], which does not require post-selection and can therefore operate much faster. Madsen *et al* [549] performed boson sampling in just 36 μs via a 216-mode squeezed photon state. Raw photon generation and detection runs at 6 MHz, putting this platform up near the top of the table for processor speed, although it cannot perform universal quantum computing.

Classical CPU clock speeds have remained around 3 GHz over the last decade, which is significantly faster than any current quantum hardware. The steady increase in processor speeds that occurred during several decades of Moore’s law scaling [5] (figure 1) is not possible for quantum hardware platforms. Individual atoms, ions or electrons have their own intrinsic physical frequencies which cannot be made smaller. We therefore need to develop programming models that can leverage processors with vastly different effective clock speeds. A hybrid algorithm may need to queue 100 or 10^4 operations on the classical processor for every operation on the quantum processor. This should be viable for some of the large, multi-scale simulations that dominate today's HPC use, where many parts of the computation can be done classically while the quantum processor solves key components that are expensive for classical computation.

5.2. Quantum data encoding

Theoretical frameworks that address the interface between classical memory and quantum devices are critical for implementing quantum algorithms. In classical computing, floating point numbers are ubiquitous for numerical data in scientific applications. However, they are not the natural way to encode data into qubits, and the choice of data encoding can have a big impact on quantum algorithm efficiency. In this section, we summarise the available data encoding approaches for quantum computing, and the challenges they present for interfacing with classical HPC.

Basis (digital) encoding. Basis encoding is the simplest encoding. It corresponds to binary encoding in classical bits. Multi-qubit basis states are often written in a single 'ket' as $\{|0\ldots00\rangle, |0\ldots01\rangle, \ldots|1\ldots11\rangle\}$ or $|j\rangle$ where it is understood that the integer j corresponds in binary to the qubit values. The bitstring $b_{n-1}\ldots b_0$ is thus encoded by the $|b_{n-1}\ldots b_0\rangle$ state. Single computational basis states are easy to prepare from the all zero state, by applying bit flips to the appropriate qubits.

Angle encoding. Angle encoding extends basis state encoding to use the ability of qubits to be in any superposition of zero and one. A convenient parameterization of this using angles ϑ is

$$\cos\vartheta|0\rangle + \sin\vartheta|1\rangle.$$

An n-qubit register can thus encode n angle variables $\{\vartheta_k\}$. This is efficient to prepare if arbitrary single qubit rotation gates are available. We can extend this by encoding a second angle φ in the phase,

$$\cos\vartheta|0\rangle + e^{i\varphi}\sin\vartheta|1\rangle,$$

and noting that qubits correspond to unit vectors on a Bloch sphere (figure 3). However, it is not possible to read out the value of the angles using a single measurement on a single copy of the qubit. While the state preparation is easy, readout is only straightforward for results encoded as basis states.

Amplitude (analogue) encoding. Amplitude encoding loads a vector of real numbers $\vec{X} = (x_0, x_1 \ldots, x_{N-1})$, with $x_k \in [0,1]$, into the amplitudes of the quantum state, so they are stored in superposition

$$\vec{X} \to \sum_{k=0}^{N-1} x_k|k\rangle.$$

The overall normalization factor is stored separately if it is not equal to one. An n-qubit register stores up to $N = 2^n$ real values, so this is an efficient encoding. However, to manipulate individual amplitudes, the number of gates grows exponentially with qubit number. Preparing an arbitrary state is thus an expensive operation [558–560]. Refinements include 'approximate amplitude encoding' [561, 562] which reduces the preparation cost to $O(\mathrm{poly}(n))$ gates for n qubits at a cost of reduced accuracy. It is also possible to encode data more efficiently for wavepacket dynamical simulations in quantum chemistry [563, 564].

QRAM encoding. Quantum random access memory (QRAM) is a general-purpose architecture for quantum oracles (see [565, 566] for excellent introduction). It is a generalization of classical RAM [567, 568]: given an address k as input, the RAM returns memory contents j_k. Analogously, QRAM takes a superposition of addresses $|\psi_{\mathrm{in}}\rangle$ and returns an entangled state $|\psi_{\mathrm{out}}\rangle$ where each address is correlated with the corresponding memory element:

$$|\psi_{\mathrm{in}}\rangle = \sum_{k=0}^{N-1}|k\rangle^A|0\rangle^B$$
$$\xrightarrow{\mathrm{QRAM}} \sum_{k=0}^{N-1}|k\rangle^A|j_k\rangle^B = |\psi_{\mathrm{out}}\rangle \qquad (59)$$

for memory size N. Input and output registers are denoted by A and B respectively. Note we may alternatively denote the memory size by 2^n, where register A contains $n \equiv \log N$ qubits. The principle behind QRAM is that, if a query state is a quantum superposition over all addresses, then the circuit responds by performing a memory access operation over all addresses simultaneously. If we imagine the memory laid out in space, a QRAM access must transfer some information to each location in memory for it to correctly perform a superposition of memory accesses. QRAM performs the operation in (59) in $O(\log N)$ time at the cost of $O(N)$ ancillary qubits. This makes QRAM especially appealing for quantum algorithms that require $O(\log N)$ query times to claim exponential speedup. However, it is still uncertain whether QRAM can be used to achieve quantum speedups, in principle or in practice [58, 569]. QRAM is sensitive to decoherence [570] and is subject to an overhead associated with error correction of ancillary qubits [571]. Recent efforts have led to developments in highly noise-resilient QRAM using bucket-brigade QRAM architecture [566], and this continues to be an active open area of research.

Floating point encodings. Classical digital computers encode floating-point numbers using the IEEE 754 international standard, established in 1985 by the Institute of Electrical and Electronics Engineers (IEEE) [572]. Work is currently underway to develop the IEEE P7130 Standard for Quantum Computing Definitions [573]. Basis state encoding is one option for floating point numbers. It uses a mantissa and an exponent in direct analogy to the binary representation of classical floating point numbers. This is not the only (or the best) method, depending on the application. For example, quantum algorithms that rely on amplitude encoding for their efficiency gains cannot use basis state encoding. Seidel *et al*

[574] explore an alternative way to design and handle various types of arithmetic operations. They formulate arithmetic ring operations as semiboolean polynomials, and extend their encoding from intergers to floating point numbers, with significant efficiency gains over basis state encodings.

Block encodings. Block encodings of matrices play an important role in quantum algorithms derived from the quantum singular value transformation [575–577], such as Hamiltonian simulation (section 3) and QML [578]. Block encoding involves embedding a (scaled) non-unitary operator $\hat{A} = \sum_j \alpha_j |w_k\rangle\langle v_k|$ as the leading principal block in a larger unitary matrix,

$$\hat{U} = \begin{bmatrix} \hat{A}/\alpha & * \\ * & * \end{bmatrix} \Leftrightarrow \hat{A} = \hat{\Pi}'\hat{U}\hat{\Pi}, \tag{60}$$

with arbitrary matrix elements $*$. The projectors $\hat{\Pi}' = \sum_j |w_k\rangle\langle w_k|$ and $\hat{\Pi} = \sum_j |v_k\rangle\langle v_k|$ locate $\hat{A}$ within $\hat{U}$: respectively, $\hat{\Pi}$ and $\hat{\Pi}'$ selects the columns and rows in which $\hat{A}$ is encoded. See [577, 579] for a pedagogical introduction.

Hamiltonian encoding. Hamiltonian encoding is suitable for quantum optimization. The problem constraints are encoded in a QUBO or the fields and couplings of an Ising Hamiltonian (section 2.4). Loading the classical data on to a quantum device involves setting the values of the couplers and fields through classical control lines. Although straightforward to implement, the required precision increases with the problem size. Additionally, not every problem can be easily mapped on to the Ising Hamiltonian parameters.

One-hot and domain-wall encoding. One-hot and domain-wall encodings are unary encodings originally designed for higher dimensional variables in quantum optimization [580, 581]. One-hot encoding sets a single bit in a position that determines the value of the variable. Domain-wall encoding is based on a topological defect in an Ising spin chain. One side of the chain is in a spin-down state, whereas the other is spin-up. The domain-wall (topological defect) is where the system jumps from spin-up to -down. The location of the domain wall represents the value of the variable. The number of qubits in the spin chain determines the number of available values. Domain wall encoding is efficient in NISQ devices for variables with a small fixed range of values [582].

5.3. Data interconnections

As HPC users know well, exchanging data between different processing units is where many bottlenecks occur in large scale computations. The physical interconnections between quantum and classical hardware are already the focus of significant engineering development, since they are necessary to control the quantum hardware. A significant issue is transferring data and instructions from room temperature to the cryogenic conditions required for operating many types of quantum hardware. Electronic and electromagnetic signals carry unwanted heat into the cryogenic environment that must be minimized and removed to maintain ideal operating conditions. This introduces another constraint for optimizing code across both platforms: minimizing data exchange to reduce cooling costs. It takes time to run the procedures to re-cool cryogenic systems, and like calibration, this is time when the quantum processors are not available for computation.

Slower quantum 'clock speeds' (section 5.1) are accompanied by even slower timescales for measurements to read out the qubit state and convert into classical data signals. Table 4 lists typical measurement times for different hardware. These factors all need to be taken into account when designing algorithms, and significantly affect the wall-clock time required [583].

The most important next step for integrating quantum computers with HPC is making test bed hardware widely available [546, 547]. Exploring their combined capabilities can guide the engineering required to integrate them more efficiently at both the hardware and software levels.

6. Outlook

Looking back over the past two decades, it is already possible to identify a quantum equivalent of Moore's Law through the steady improvements in the qubit number and quality (figure 21). This is most evident in platforms that have been under development for the longest, such as superconducting qubits and ion traps. Early quantum computers are available now through cloud computing services, such as Amazon Braket, enabling researchers to develop and test of proof-of-concept algorithms. Cloud accessible quantum computing was pioneered by IBM, who released their first cloud accessible five-qubit superconducting quantum computer in 2016 [585]. IBM recently announced their newest quantum chip, which provides over a thousand physical qubits [586], an impressive development trajectory. However, further progress requires error correction [48, 49] to reduce the accumulated errors that occur during computation, and hence enable long enough computations for useful applications. This will produce high quality qubits, but each of IBM's thousand-qubit chips will contain only a few of these logical qubits, which will then need to be connected through sophisticated quantum network interfaces [556].

Ion trap based systems have made similar progress, prioritizing qubit quality to achieve the highest fidelity quantum gates of any current platform [587], in an academic research lab setting. IonQ's latest systems have up to 32 qubits, with 64 qubit systems due in 2025. This is enough qubits for simulating quantum systems beyond the reach of classical HPC. Again, significantly larger systems will require error correction. Newer entrants, such as Rydberg [588, 589] (neutral atoms), are catching up fast in performance and flexibility. Despite the current leading position of superconducting qubits and ion traps, it is by no means clear they will turn out to be the main type of quantum hardware in the longer term. Also firmly in the running are photonic and silicon or other semiconductor-based systems. An ecosystem that supports several types of hardware for different applications is both

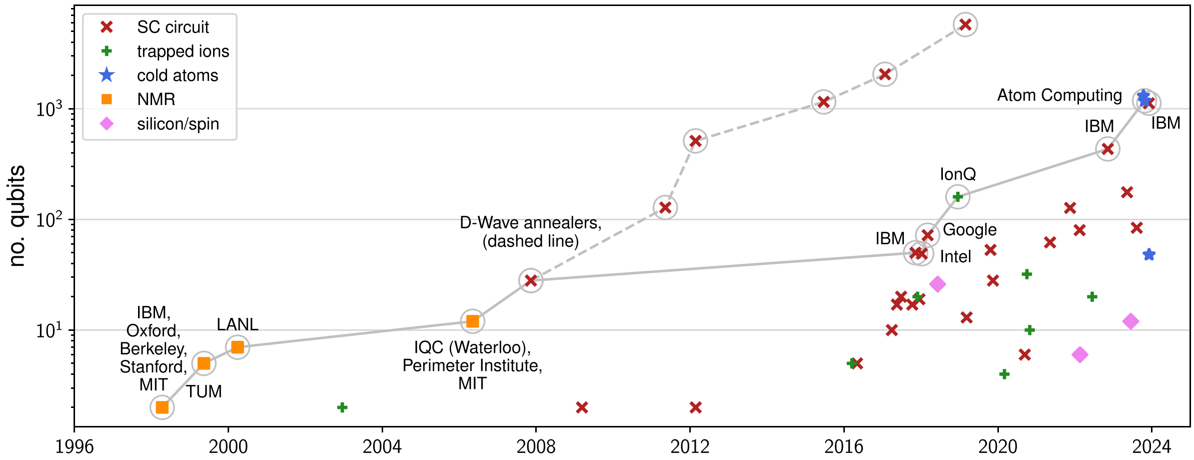


Figure 21. Growth in number of qubits per device from 1998 to 2024. Showing SC circuits (red ×), trapped ions (green +), cold atoms (blue stars), NMR (orange squares) and silicon/spin (pink diamonds) platforms. Results from selected teams annotated with team name and circled in grey. Compiled by R Au-Yeung from Statista [584] and press releases.

plausible and promising. Despite the significant engineering challenges, there is growing confidence that the hardware will deliver on the anticipated timescales. Companies are forecasting fully error corrected quantum processors on the scale of hundreds or thousands of qubits by 2030, and are investing the resources necessary to achieve this.

However, standalone quantum hardware will support only niche applications. Some of these are significant: simulating many-body quantum systems (see section 3) is a natural problem for quantum hardware that can potentially deliver many new scientific results with valuable impact in a wide range of commercial sectors. Nonetheless, to leverage the full potential of quantum computing, interfacing with classical HPC is essential, and already underway [546, 547]. As discussed in section 5, there are major challenges here that require new science and engineering to be developed. The most important are related to the different data encodings that are natural for quantum computers, making it highly non-trivial to transfer data between quantum and classical processors. This is compounded by clock speed mismatches that, depending on the type of quantum hardware, can be up to six orders of magnitude. While data conversion and transfer are largely an engineering challenge, we do not yet have good programming models for how to handle asynchronous computing on these timescales. Commercial promises of seamless software integration are plausible for test bed applications, but do not yet have an assured path for scaling up.

In addition, quantum computing is being developed in a context of significant changes in classical computing. Physical limits have been reached for scaling up conventional CMOS CPUs: the energy used by a typical large data centre or HPC facility is equivalent to a small town. More compute power requires new hardware that is less power-hungry for the same amount of computation. GPUs have provided this for the past decade, but they have also shown us that redesigning codes and algorithms to use different types of hardware is extremely challenging and time-consuming for large scale applications [4]. Quantum processors will be significantly more challenging still to deploy alongside classical HPC at scale. The investment of time and resources necessary to leverage quantum computing is a significant barrier to widespread adoption as quantum computers become more available. Applications developed and tested in an academic setting are important for opening up areas with commercial potential.

AI is also accelerating the pace of change in classical computing. In recent years, AI has come under scrutiny over its high energy consumption [590, 591] and carbon emissions. For example, the carbon footprint of training a large natural language processing (NLP) model equals roughly 300 tons of carbon dioxide emissions [592], or 125 round-trip flights between New York City and Beijing [590]. This increases the pressure to develop energy-efficient hardware optimized for the massive data processing required for AI applications. This will significantly reshape the application areas where quantum computing can deliver real benefits. While AI may overtake quantum in predicting solutions to scientific computing applications, quantum may in turn provide advantages [39, 187] for other types of AI applications, as quantum hardware becomes more capable.

Despite the remaining hurdles on the path to useful quantum computing, this is an exciting time to be doing computational science and engineering. There are multiple potential routes to significantly larger simulations and the scientific and technical breakthroughs they will produce. Achieving these goals requires interdisciplinary collaborations to identify where and how quantum computing can provide the most benefits. There are multiple projects aiming to accelerate this process. Using the UK for examples, we have the

- National Quantum Computing Centre's SPARQ program and hackathons https://www.nqcc.ac.uk/engage/sparq-programme/,
- Quantum Software Lab in Edinburgh https://www.quantumsoftwarelab.com/,
- Collaborative Computational Project on Quantum Computing https://ccp-qc.ac.uk/,

- ExCALIBUR cross-cutting project QEVEC https://excalibur.ac.uk/projects/qevec/,
- multiple projects funded by the Software for Quantum Computing call https://gow.jpgrc.ukri.org/NGBOViewPanel.aspx?PanelId = 1-FIN46S,
- and the Quantum Technology Hub for Quantum Computing and Simulation https://www.qcshub.org/, and its recently announced successor, the Quantum Technology Hub for Quantum Computing via Integrated and Interconnected Implementations, due to start in December 2024.

The Royal Academy of Engineering 2024 Quantum Infrastructure Review outlines a range of recommendations for developing the UK's quantum tech sector over the next decade [593]. This is motivated by the need to transition from a successful program of fundamental research to commercialization and deployment in real-world applications. As a result, industry will require a workforce that is knowledgeable about quantum mechanics [594]. However, there are fears that unless we enhance accessibility to quantum education [595, 596] and persuade students that it is a realistic career path [597, 598], current inequities will persist into the future quantum workforce. Initiatives such as QWorld https://qworld.net/ allows researchers from any country with internet access to participate in quantum application development. Since many countries do not have their own quantum computing hardware development programs, this enlarges the overall effort and ensures that a wide range of relevant problems are tackled.

In the global race to develop quantum computers, there has been growing debate on its ethical use [599, 600], especially given the discussion on potential military applications [601]. Even as governments move to restrict technology access for national security reasons, it is essential for researchers to have cloud access to compute resources, to ensure the benefits are available to all.

Data availability statement

No new data were created or analyzed in this study.

Acknowledgments

RA, BC, OR, and VK are supported by UK Research and Innovation (UKRI) Grant EP/W00772X/2 (QEVEC). RA, OR, and VK are supported by EP/Y004566/1 (QuANDiE). RA, and VK are supported by EP/T001062/1 (QCS Hub). VK is supported by EP/T026715/2 (CCP-QC). The authors thank John Buckeridge, Nick Chancellor, Animesh Datta, Steph Foulds and Steve Lind for useful discussions and proofreading the manuscript.

ORCID iDs

R Au-Yeung https://orcid.org/0000-0002-0082-5382
B Camino https://orcid.org/0000-0002-9569-2219
O Rathore https://orcid.org/0009-0002-2087-6715
V Kendon https://orcid.org/0000-0002-6551-3056

References

[1] Arute F *et al* 2019 Quantum supremacy using a programmable superconducting processor *Nature* **574** 505–10

[2] Li Y, Gan L, Chen M, Chen Y, Lu H, Lu C, Pan J, Fu H and Yang G 2022 Benchmarking 50-photon Gaussian boson sampling on the Sunway TaihuLight *IEEE Trans. Parallel Distrib. Syst.* **33** 1357–72

[3] Mann A 2020 Nascent exascale supercomputers offer promise, present challenges *Proc. Natl Acad. Sci.* **117** 22623–5

[4] Betcke T, Bungartz H-J, Wells G and Woodley S 2022 ExCALIBUR–U.K.'s preparation for the arrival of the next generation of HPC *Comput. Sci. Eng.* **24** 5–7

[5] Moore G E 1998 Cramming more components onto integrated circuits *Proc. IEEE* **86** 82–85

[6] Khan H N, Hounshell D A and Fuchs E R H 2018 Science and research policy at the end of Moore's law *Nat. Electron.* **1** 14–21

[7] Leiserson C E, Thompson N C, Emer J S, Kuszmaul B C, Lampson B W, Sanchez D and Schardl T B 2020 There's plenty of room at the top: what will drive computer performance after Moore's law? *Science* **368** eaam9744

[8] Rupp K 2022 Microprocessor trend data (available at: https://github.com/karlrupp/microprocessor-trend-data)

[9] Strohmaier E, Dongarra J, Simon H and Meuer M 2023 Top500 (available at: https://top500.org/) (Accessed 1 August 2023, updated every six months)

[10] Strohmaier E, Dongarra J, Simon H and Meuer M 2023 Green500 (available at: www.top500.org/lists/green500/) (Accessed 3 December 2023, updated every six months)

[11] Feynman R P 1982 Simulating physics with computers *Int. J. Theor. Phys.* **21** 467–88

[12] Deutsch D 1985 Quantum theory, the Church-Turing principle and the universal quantum computer *Proc. R. Soc.* A **400** 97–117

[13] Shor P W 1994 Polynomial time algorithms for discrete logarithms and factoring on a quantum computer *Algorithmic Number Theory* (*Lecture Notes in Computer Science* vol 877) ed L M Adleman and M-D Huang (Springer) p 289

[14] Grover L K 1996 A fast quantum mechanical algorithm for database search *Proc. 28th STOC* (ACM) pp 212–9

[15] Grover L K 1997 Quantum mechanics helps in searching for a needle in a haystack *Phys. Rev. Lett.* **79** 325–8

[16] Georgescu I M, Ashhab S and Nori F 2014 Quantum simulation *Rev. Mod. Phys.* **86** 153–85

[17] Altman E *et al* 2021 Quantum simulators: architectures and opportunities *PRX Quantum* **2** 017003

[18] Department for Science, Innovation and Technology 2023 National quantum strategy *Technical Report* (UK Government) (available at: www.gov.uk/government/publications/national-quantum-strategy)

[19] Acín A *et al* 2018 The quantum technologies roadmap: a European community view *New J. Phys.* **20** 080201

[20] Monroe C, Raymer M G and Taylor J 2019 The U.S. national quantum initiative: from act to action *Science* **364** 440–2

[21] Department of Industry, Science and Resources 2023 National quantum strategy *Technical Report* Australian Government (available at: www.industry.gov.au/publications/national-quantum-strategy)

[22] UNESCO 2024 International year of quantum science and technology (available at: https://quantum2025.org) (Accessed 15 June 2024)

[23] MacQuarrie E R, Simon C, Simmons S and Maine E 2020 The emerging commercial landscape of quantum computing *Nat. Rev. Phys.* **2** 596–8

[24] Gibney E 2019 Quantum gold rush: the private funding pouring into quantum start-ups *Nature* **574** 22–24

[25] Bova F, Goldfarb A and Melko R G 2021 Commercial applications of quantum computing *EPJ Quantum Technol.* **8** 2

[26] Preskill J 2018 Quantum computing in the NISQ era and beyond *Quantum* **2** 79

[27] Bharti K *et al* 2022 Noisy intermediate-scale quantum algorithms *Rev. Mod. Phys.* **94** 015004

[28] IQM 2023 Technology (available at: https://meetiqm.com/technology/) (Accessed 1 December 2023)

[29] Atos 2020 Atos joins forces with start-up Pasqal to accelerate high performance computing using quantum neutral atom technology (available at: https://atos.net/en/2020/press-release_2020_11_04/atos-joins-forces-with-start-up-pasqal-to-accelerate-high-performance-computing-using-quantum-neutral-atom-technology) (Accessed 1 December 2023)

[30] Atos 2021 Atos confirms role as global leader in quantum hybridization technologies at its 8th Quantum Advisory Board (available at: https://atos.net/en/2021/press-release_2021_12_03/atos-confirms-role-as-global-leader-in-quantum-hybridization-technologies-at-its-8th-quantum-advisory-board) (Accessed 1 August 2024)

[31] HPCQS 2021 Towards a world-class supercomputing ecosystem—HPCQS pioneers federated quantum-super-computing in Europe (available at: https://web.archive.org/web/20211201181208/www.hpcqs.eu/) (Accessed 1 August 2024)

[32] IQM 2024 Germany launches its first hybrid quantum computer at Leibniz Supercomputing Centre (available at: www.meetiqm.com/newsroom/press-releases/germany-launches-its-first-hybrid-quantum-computer) (Accessed 1 August 2024)

[33] McCaskey A 2022 The road to the hybrid quantum-HPC data center starts here (available at: https://blogs.nvidia.com/blog/quantum-computing-hpc-isc2022) (Accessed 1 December 2023)

[34] Zapata AI 2023 Build and deploy industrial generative AI applications on Orquestra (available at: https://zapata.ai/platform-orquestra/) (Accessed 1 December 2023)

[35] Herr D, Duke C, Godel M, Natraj A, Makowska A, Williams R, Perkins R and Visser C 2019 The impact of EPSRC's investments in high performance computing infrastructure: final report *Technical Report* (London Economics) (available at: www.ukri.org/publications/impact-of-epsrcs-investments-in-hpc-infrastructure/)

[36] Department for Science, Innovation and Technology 2022 Independent review of the future of compute: final report and recommendations *Technical Report* (UK Government) (available at: www.gov.uk/government/publications/future-of-compute-review/the-future-of-compute-report-of-the-review-of-independent-panel-of-experts)

[37] European Commission 2023 The European high performance computing joint undertaking (available at: https://digital-strategy.ec.europa.eu/en/policies/high-performance-computing-joint-undertaking) (Accessed 1 December 2023)

[38] UK Research and Innovation 2024 Digital research infrastructure (available at: www.ukri.org/what-we-do/creating-world-class-research-and-innovation-infrastructure/digital-research-infrastructure/) (Accessed 1 August 2024)

[39] Dunjko V and Briegel H J 2018 Machine learning & artificial intelligence in the quantum domain: a review of recent progress *Rep. Prog. Phys.* **81** 074001

[40] Nielsen M A and Chuang I L 2010 *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge University Press) (https://doi.org/10.1017/CBO9780511976667)

[41] Abhijith J *et al* 2022 Quantum algorithm implementations for beginners *ACM Trans. Quantum Comput.* **3** 18

[42] Montanaro A 2016 Quantum algorithms: an overview *npj Quantum Inf.* **2** 15023

[43] Albash T and Lidar D A 2018 Adiabatic quantum computation *Rev. Mod. Phys.* **90** 015002

[44] Childs A M and van Dam W 2010 Quantum algorithms for algebraic problems *Rev. Mod. Phys.* **82** 1–52

[45] Jordan S 2021 Quantum algorithm zoo (available at: https://quantumalgorithmzoo.org) (Accessed 1 August 2022)

[46] Bernstein E and Vazirani U 1993 Quantum complexity theory *Proc. 25th STOC* (ACM) pp 11–20

[47] Vazirani U 2002 A survey of quantum complexity theory *Quantum Computation: A Grand Mathematical Challenge for the Twenty-First Century and the Millennium* (*Proceedings of Symposia in Applied Mathematics* vol 58) (American Mathematical Society) pp 193–220

[48] Terhal B M 2015 Quantum error correction for quantum memories *Rev. Mod. Phys.* **87** 307–46

[49] Roffe J 2019 Quantum error correction: an introductory guide *Contemp. Phys.* **60** 226–45

[50] Eisert J, Hangleiter D, Walk N, Roth I, Markham D, Parekh R, Chabaud U and Kashefi E 2020 Quantum certification and benchmarking *Nat. Rev. Phys.* **2** 382–90

[51] Carrasco J, Elben A, Kokail C, Kraus B and Zoller P 2021 Theoretical and experimental perspectives of quantum verification *PRX Quantum* **2** 010102

[52] Hauke P, Katzgraber H G, Lechner W, Nishimori H and Oliver W D 2020 Perspectives of quantum annealing: methods and implementations *Rep. Prog. Phys.* **83** 054401

[53] Yarkoni S, Raponi E, Bäck T and Schmitt S 2022 Quantum annealing for industry applications: introduction and review *Rep. Prog. Phys.* **85** 104001

[54] Wu Y *et al* 2021 Strong quantum computational advantage using a superconducting quantum processor *Phys. Rev. Lett.* **127** 180501

[55] Kim Y *et al* 2023 Evidence for the utility of quantum computing before fault tolerance *Nature* **618** 500–5

[56] Leymann F and Barzen J 2020 The bitter truth about gate-based quantum algorithms in the NISQ era *Quantum Sci. Technol.* **5** 044007

[57] Coveney P V and Highfield R R 2020 From digital hype to analogue reality: universal simulation beyond the quantum and exascale eras *J. Comput. Sci.* **46** 101093

[58] Aaronson S 2015 Read the fine print *Nat. Phys.* **11** 291–3

[59] Chancellor N, Cumming R and Thomas T 2020 Toward a standardized methodology for constructing quantum computing use cases (arXiv:2006.05846)

[60] Perini S and Ciarletta F 2021 Untangling the HPC innovation dilemma through quantum computing *Technical Report* (IQM and Atos) (available at: www.meetiqm.com/technology/iqm-atos-state-of-quantum-research-2021/)

[61] Wootters W and Zurek W 1982 A single quantum cannot be cloned *Nature* **299** 802–3

[62] van Meter R and Horsman D 2013 A blueprint for building a quantum computer *Commun. ACM* **56** 84–93

[63] DiVincenzo D P 2000 The physical implementation of quantum computation *Fortschr. Phys.* **48** 771–83

[64] Reiher M, Wiebe N, Svore K M, Wecker D and Troyer M 2017 Elucidating reaction mechanisms on quantum computers *Proc. Natl. Acad. Sci.* **114** 7555–60

[65] Portugal R 2022 Basic quantum algorithms (arXiv: 2201.10574)

[66] Qiskit 2023 Qiskit is the open-source toolkit for useful quantum (available at: www.ibm.com/quantum/qiskit) (Accessed 1 October 2023)

[67] Camps D, Van Beeumen R and Yang C 2020 Quantum Fourier transform revisited *Numer. Linear Algebra Appl.* **28** e2331

[68] Griffiths R B and Niu C-S 1996 Semiclassical Fourier transform for quantum computation *Phys. Rev. Lett.* **76** 3228–31

[69] Browne D 2007 Efficient classical simulation of the semi-classical quantum Fourier transform *New J. Phys.* **9** 146

[70] Kitaev A Y 1996 Quantum measurements and the Abelian stabilizer problem *Electron. Colloquium Comput. Complex.* **3** 22

[71] Ahmadi H and Chiang C-F 2012 Quantum phase estimation with arbitrary constant-precision phase shift operators *Quantum Inf. Comput.* **12** 864–75

[72] Ni H, Li H and Ying L 2023 On low-depth algorithms for quantum phase estimation *Quantum* **7** 2265

[73] Aspuru-Guzik A, Dutoi A D, Love P J and Head-Gordon M 2005 Simulated quantum computation of molecular energies *Science* **309** 1704–7

[74] Ekert A and Jozsa R 1996 Quantum computation and Shor's factoring algorithm *Rev. Mod. Phys.* **68** 733–53

[75] Rivest R L, Shamir A and Adleman L 1978 A method for obtaining digital signatures and public-key cryptosystems *Commun. ACM* **21** 120–6

[76] Zimmermann P 2019 Factorization of RSA-250 (available at: https://sympa.inria.fr/sympa/arc/cado-nfs/2020-02/msg00001.html) Previous records: Kleinjung, Thomas (18 February 2010). "Factorization of a 768-bit RSA modulus". Thomé, Emmanuel (2 December 2019). "795-bit factoring and discrete logarithms". cado-nfs-discuss (Mailing list)

[77] van Meter R and Devitt S J 2016 The path to scalable distributed quantum computing *IEEE Comput.* **49** 31–42

[78] Ambainis A and Špalek R 2006 Quantum algorithms for matching and network flows *STACS 2006* ed B Durand and W Thomas (Springer) pp 172–83

[79] Shenvi N, Kempe J and Birgitta Whaley K 2003 Quantum random-walk search algorithm *Phys. Rev.* A **67** 052307

[80] Childs A and Eisenberg J M 2005 Quantum algorithms for subset finding *Quantum Inf. Comput.* **5** 593–604

[81] Brassard G, Høyer P, Mosca M and Tapp A 2002 Quantum amplitude amplification and estimation *Quantum Computation and Information* (*Contemporary Mathematics* vol 305) (American Mathematical Society) pp 53–74

[82] Ambainis A 2012 Variable time amplitude amplification and quantum algorithms for linear algebra problems *29th Int. STACS 2012* (*LIPIcs* vol 14) ed C Dürr and T Wilke (Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik) pp 636–47

[83] Hogg T and Portnov D 2000 Quantum optimization *Inf. Sci.* **128** 181–97

[84] Dürr C and Høyer P 1996 A quantum algorithm for finding the minimum (arXiv:quant-ph/9607014)

[85] Dürr C, Heiligman M, Høyer P and Mhalla M 2004 Quantum query complexity of some graph problems *ICALP 2004* (Springer) pp 481–93

[86] Ramesh H and Vinay V 2003 String matching in $\tilde{O}(\sqrt{n}+\sqrt{m})$ quantum time *J. Discrete Algorithms* **1** 103–10

[87] Brassard G, Høyer P and Tapp A 1998 Quantum counting *Automata, Languages and Programming* (*LNCS* vol 1443) ed K G Larsen, S Skyum and G Winskel (Springer) pp 820–31

[88] Davenport J H and Pring B 2020 Improvements to quantum search techniques for block-ciphers, with applications to AES *Selected Areas in Cryptography: 27th Int. Conf.* (Springer) pp 360–84

[89] Motwani R and Raghavan P 1995 *Randomized Algorithms* (Cambridge University Press) (https://doi.org/10.1017/CBO9780511814075)

[90] Aharonov D, Ambainis A, Kempe J and Vazirani U 2001 Quantum walks on graphs *Proc. 33rd ACM STOC* (ACM) pp 50–59

[91] Ambainis A, Bach E, Nayak A, Vishwanath A and Watrous J 2001 One-dimensional quantum walks *Proc. 33rd ACM STOC* (ACM) pp 60–69

[92] Farhi E and Gutmann S 1998 Quantum computation and decision trees *Phys. Rev.* A **58** 915–28

[93] Childs A M, Cleve R, Deotto E, Farhi E, Gutmann S and Spielman D A 2003 Exponential algorithmic speedup by a quantum walk *Proc. 35th STOC* (ACM) pp 59–68

[94] Venegas-Andraca S E 2012 Quantum walks: a comprehensive review *Quantum Inf. Process.* **11** 1015–106

[95] Kadian K, Garhwal S and Kumar A 2021 Quantum walk and its application domains: a systematic review *Comput. Sci. Rev.* **41** 100419

[96] Callison A, Chancellor N, Mintert F and Kendon V 2019 Finding spin glass ground states using quantum walks *New J. Phys.* **21** 123022

[97] Childs A M 2009 Universal computation by quantum walk *Phys. Rev. Lett.* **102** 180501

[98] Lovett N B, Cooper S, Everitt M, Trevers M and Kendon V 2010 Universal quantum computation using the discrete-time quantum walk *Phys. Rev.* A **81** 042330

[99] Aaronson S and Arkhipov A 2011 The computational complexity of linear optics *Proc. 43rd STOC* (ACM) pp 333–42

[100] Childs A M, Gosset D and Webb Z 2013 Universal computation by multiparticle quantum walk *Science* **339** 791–4

[101] Schumacher B and Werner R F 2004 Reversible quantum cellular automata (arXiv:quant-ph/0405174)

[102] Karski M, Forster L, Choi J-M, Steffen A, Alt W, Meschede D and Widera A 2009 Quantum walk in position space with single optically trapped atoms *Science* **325** 174–7

[103] Harrow A W, Hassidim A and Lloyd S 2009 Quantum algorithm for linear systems of equations *Phys. Rev. Lett.* **103** 150502

[104] Dervovic D, Herbster M, Mountney P, Severini S, Usher N and Wossnig L 2018 Quantum linear systems algorithms: a primer (arXiv:1802.08227)

[105] Childs A M, Kothari R and Somma R D 2017 Quantum algorithm for systems of linear equations with exponentially improved dependence on precision *SIAM J. Comput.* **46** 1920–50

[106] Montanaro A and Pallister S 2016 Quantum algorithms and the finite element method *Phys. Rev.* A **93** 032324

[107] Chowdhury A N and Somma R D 2017 Quantum algorithms for Gibbs sampling and hitting-time estimation *Quantum Inf. Comput.* **17** 41–64

[108] Saito Y, Lee X, Cai D and Asai N 2021 An iterative improvement method for HHL algorithm for solving linear system of equations (arXiv:2108.07744)

[109] Lee Y, Joo J and Lee S 2019 Hybrid quantum linear equation algorithm and its experimental test on IBM Quantum Experience *Sci. Rep.* **9** 4778

[110] Gao F, Wu G, Guo S, Dai W and Shuang F 2023 Solving DC power flow problems using quantum and hybrid algorithms *Appl. Soft. Comput.* **137** 110147

[111] Angara P P, Stege U, Müller H A and Bozzo-Rey M 2020 Hybrid quantum-classical problem solving in the NISQ era *CASCON '20* (IBM) pp 247–52

[112] Wilson A G, Dann C and Nickisch H 2015 Thoughts on massively scalable Gaussian processes (arXiv: 1511.01870)

[113] Wossnig L, Zhao Z and Prakash A 2018 Quantum linear system algorithm for dense matrices *Phys. Rev. Lett.* **120** 050502

[114] Bravo-Prieto C, García-Martín D and Latorre J I 2020 Quantum singular value decomposer *Phys. Rev.* A **101** 062310

[115] Rebentrost P, Mohseni M and Lloyd S 2014 Quantum support vector machine for big data classification *Phys. Rev. Lett.* **113** 130503

[116] Lloyd S, Mohseni M and Rebentrost P 2014 Quantum principal component analysis *Nat. Phys.* **10** 631–3

[117] Cong I and Duan L 2016 Quantum discriminant analysis for dimensionality reduction and classification *New J. Phys.* **18** 073011

[118] Wiebe N, Braun D and Lloyd S 2012 Quantum algorithm for data fitting *Phys. Rev. Lett.* **109** 050505

[119] Wang G 2017 Quantum algorithm for linear regression *Phys. Rev.* A **96** 012335

[120] Schuld M, Sinayskiy I and Petruccione F 2016 Prediction by linear regression on a quantum computer *Phys. Rev.* A **94** 022342

[121] Yu C-H, Gao F and Wen Q-Y 2021 An improved quantum algorithm for ridge regression *IEEE Trans. Knowl. Data Eng.* **33** 858–66

[122] Kerenidis I and Prakash A 2017 Quantum recommendation systems *8th ITCS 2017* (*LIPIcs* vol 67) ed C H Papadimitriou (Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik) pp 49:1–49:21

[123] Duan B, Yuan J, Liu Y and Li D 2018 Efficient quantum circuit for singular-value thresholding *Phys. Rev.* A **98** 012308

[124] Rebentrost P, Bromley T R, Weedbrook C and Lloyd S 2018 Quantum Hopfield neural network *Phys. Rev.* A **98** 042308

[125] Rathore O, Basden A, Chancellor N and Kusumaatmaja H 2024 Integrating quantum algorithms into classical frameworks: a predictor-corrector approach using HHL (arXiv:2406.19996)

[126] Bharadwaj S S and Sreenivasan K R 2023 Hybrid quantum algorithms for flow problems *Proc. Natl. Acad. Sci.* **120** e2311014120

[127] Emani P S *et al* 2021 Quantum computing at the frontiers of biological sciences *Nat. Methods* **18** 701–9

[128] Kirkpatrick S, Gelatt C D and Vecchi M P 1983 Optimization by simulated annealing *Science* **220** 671–80

[129] Finnila A B, Gomez M A, Sebenik C, Stenson C and Doll J D 1994 Quantum annealing: a new method for minimizing multidimensional functions *Chem. Phys. Lett.* **219** 343

[130] Kadowaki T and Nishimori H 1998 Quantum annealing in the transverse Ising model *Phys. Rev.* E **58** 5355–63

[131] Au-Yeung R, Chancellor N and Halffmann P 2023 NP-hard but no longer hard to solve? Using quantum computing to tackle optimization problems *Front. Quantum Sci. Technol.* **2** 1128576

[132] Lucas A 2014 Ising formulations of many NP problems *Front. Phys.* **2** 5

[133] Choi V 2011 Different adiabatic quantum optimization algorithms for the NP-complete exact cover and 3SAT problems *Quantum Inf. Comput.* **11** 638–48

[134] Farhi E, Goldstone J, Gutmann S, Lapan J, Lundgren A and Preda D 2001 A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem *Science* **292** 472–5

[135] Childs A M, Farhi E and Preskill J 2001 Robustness of adiabatic quantum computation *Phys. Rev.* A **65** 012322

[136] Aharonov D, van Dam W, Kempe J, Landau Z, Lloyd S and Regev O 2004 Adiabatic quantum computation is equivalent to standard quantum computation *45th IEEE SFCS* (IEEE) pp 42–51

[137] Aharonov D, van Dam W, Kempe J, Landau Z, Lloyd S and Regev O 2008 Adiabatic quantum computation is equivalent to standard quantum computation *SIAM Rev.* **50** 755–87

[138] Crosson E J and Lidar D A 2021 Prospects for quantum enhancement with diabatic quantum annealing *Nat. Rev. Phys.* **3** 466–89

[139] Farhi E, Goldstone J and Gutmann S 2014 A quantum approximate optimization algorithm (arXiv:1411.4028)

[140] Hadfield S, Wang Z, O'Gorman B, Rieffel E G, Venturelli D and Biswas R 2019 From the quantum approximate optimization algorithm to a quantum alternating operator ansatz *Algorithms* **12** 34

[141] Banks R J *et al* 2024 Continuous-time quantum walks for MAX-CUT are hot *Quantum* **8** 1254

[142] Schulz S, Willsch D and Michielsen K 2024 Guided quantum walk *Phys. Rev. Res.* **6** 013312

[143] Cerezo M *et al* 2021 Variational quantum algorithms *Nat. Rev. Phys.* **3** 625–44

[144] Lubasch M, Joo J, Moinier P, Kiffner M and Jaksch D 2020 Variational quantum algorithms for nonlinear problems *Phys. Rev.* A **101** 010301

[145] Kyriienko O, Paine A E and Elfving V E 2021 Solving nonlinear differential equations with differentiable quantum circuits *Phys. Rev.* A **103** 052416

[146] Kokail C *et al* 2019 Self-verifying variational quantum simulation of lattice models *Nature* **569** 355–60

[147] Kandala A, Mezzacapo A, Temme K, Takita M, Brink M, Chow J M and Gambetta J M 2017 Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets *Nature* **549** 242–6

[148] Botelho L, Glos A, Kundu A, Miszczak J A, Salehi O and Zimborás Z 2022 Error mitigation for variational quantum algorithms through mid-circuit measurements *Phys. Rev.* A **105** 022441

[149] Cai Z, Babbush R, Benjamin S C, Endo S, Huggins W J, Li Y, McClean J R and O'Brien T E 2023 Quantum error mitigation *Rev. Mod. Phys.* **95** 045005

[150] Zhou L, Wang S-T, Choi S, Pichler H and Lukin M D 2020 Quantum approximate optimization algorithm: performance, mechanism and implementation on near-term devices *Phys. Rev.* X **10** 021067

[151] Wang Z, Rubin N C, Dominy J M and Rieffel E G 2020 *XY* mixers: analytical and numerical results for the quantum alternating operator ansatz *Phys. Rev.* A **101** 012320

[152] Blekos K, Brand D, Ceschini A, Chou C-H, Li R-H, Pandya K and Summer A 2024 A review on quantum approximate optimization algorithm and its variants *Phys. Rep.* **1068** 1–66

[153] Jain N, Coyle B, Kashefi E and Kumar N 2022 Graph neural network initialisation of quantum approximate optimisation *Quantum* **6** 861

[154] Galda A, Liu X, Lykov D, Alexeev Y and Safro I 2021 Transferability of optimal QAOA parameters between random graphs *2021 QCE* (IEEE) pp 171–80

[155] Moussa C, Wang H, Bäck T and Dunjko V 2022 Unsupervised strategies for identifying optimal parameters in quantum approximate optimization algorithm *EPJ Quantum Technol.* **9** 11

[156] Cheng L, Chen Y-Q, Zhang S-X and Zhang S 2024 Quantum approximate optimization via learning-based adaptive optimization *Commun. Phys.* **7** 83

[157] Alam M, Ash-Saki A and Ghosh S 2020 Accelerating quantum approximate optimization algorithm using machine learning *2020 DATE* (IEEE) pp 686–9

[158] Dong Y, Meng X, Lin L, Kosut R and Whaley K B 2020 Robust control optimization for quantum approximate optimization algorithms *IFAC-PapersOnLine* **53** 242–9

[159] Shaydulin R, Hadfield S, Hogg T and Safro I 2021 Classical symmetries and the quantum approximate optimization algorithm *Quantum Inf. Process.* **20** 359

[160] Shaydulin R and Wild S M 2021 Exploiting symmetry reduces the cost of training QAOA *IEEE Trans. Quantum Eng.* **2** 1–0

[161] Brady L T, Baldwin C L, Bapat A, Kharkov Y and Gorshkov A V 2021 Optimal protocols in quantum annealing and quantum approximate optimization algorithm problems *Phys. Rev. Lett.* **126** 070505

[162] Yang Z-C, Rahmani A, Shabani A, Neven H and Chamon C 2017 Optimizing variational quantum algorithms using Pontryagin's minimum principle *Phys. Rev.* X **7** 021027

[163] Gerblich L, Dasanjh T, Wong H Q X, Ross D, Novo L, Chancellor N and Kendon V 2024 Advantages of multistage quantum walks over QAOA (arXiv:2407.06663)

[164] McClean J R, Romero J, Babbush R and Aspuru-Guzik A 2016 The theory of variational hybrid quantum-classical algorithms *New J. Phys.* **18** 023023

[165] Tilly J *et al* 2022 The variational quantum eigensolver: a review of methods and best practices *Phys. Rep.* **986** 1–128

[166] Amaro D, Modica C, Rosenkranz M, Fiorentini M, Benedetti M and Lubasch M 2022 Filtering variational quantum algorithms for combinatorial optimization *Quantum Sci. Technol.* **7** 015021

[167] Lee J, Huggins W J, Head-Gordon M and Whaley K B 2019 Generalized unitary coupled cluster wave functions for quantum computation *J. Chem. Theory Comput.* **13** 311–24

[168] Low G H and Chuang I L 2019 Hamiltonian simulation by qubitization *Quantum* **3** 163

[169] Yuan X, Endo S, Zhao Q, Li Y and Benjamin S C 2019 Theory of variational quantum simulation *Quantum* **3** 191

[170] Endo S, Sun J, Li Y, Benjamin S C and Yuan X 2020 Variational quantum simulation of general processes *Phys. Rev. Lett.* **125** 010501

[171] Xu X, Sun J, Endo S, Li Y, Benjamin S C and Yuan X 2021 Variational algorithms for linear algebra *Sci. Bull.* **66** 2181–8

[172] LeCun Y, Bengio Y and Hinton G 2015 Deep learning *Nature* **521** 436–44

[173] Abbas A, Sutter D, Zoufal C, Lucchi A, Figalli A and Woerner S 2021 The power of quantum neural networks *Nat. Comput. Sci.* **1** 403–9

[174] Schuld M and Killoran N 2019 Quantum machine learning in feature Hilbert spaces *Phys. Rev. Lett.* **122** 040504

[175] Farhi E and Neven H 2018 Classification with quantum neural networks on near term processors (arXiv:1802.06002)

[176] Cong I, Choi S and Lukin M D 2019 Quantum convolutional neural networks *Nat. Phys.* **15** 1273–8

[177] Beer K, Bondarenk D, Farrelly T, Osborne T J, Salzmann R, Scheiermann D and Wolf R 2020 Training deep quantum neural networks *Nat. Commun.* **11** 808

[178] Schuld M and Petruccione F 2021 Quantum Models as Kernel Methods, *Machine Learning with Quantum Computers* (Springer) ch 6, pp 217–45

[179] Schuld M, Bocharov A, Svore K M and Wiebe N 2020 Circuit-centric quantum classifiers *Phys. Rev.* A **101** 032308

[180] Schuld M, Sweke R and Meyer J J 2021 Effect of data encoding on the expressive power of variational quantum-machine-learning models *Phys. Rev.* A **103** 032430

[181] Amin M H, Andriyash E, Rolfe J, Kulchytskyy B and Melko R 2018 Quantum Boltzmann machine *Phys. Rev.* X **8** 021050

[182] Kieferová M and Wiebe N 2017 Tomography and generative training with quantum Boltzmann machines *Phys. Rev.* A **96** 062327

[183] Benedetti M, Realpe-Gómez J, Biswas R and Perdomo-Ortiz A 2016 Estimation of effective temperatures in quantum annealers for sampling applications: a case study with possible applications in deep learning *Phys. Rev.* A **94** 022308

[184] Xu G and Oates W S 2021 Adaptive hyperparameter updating for training restricted Boltzmann machines on quantum annealers *Sci. Rep.* **11** 2727

[185] Delilbasic A, Le Saux B, Riedel M, Michielsen K and Cavallaro G 2024 A single-step multiclass SVM based on quantum annealing for remote sensing data classification *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **17** 1434–45

[186] Willsch D, Willsch M, De Raedt H and Michielsen K 2020 Support vector machines on the D-Wave quantum annealer *Comput. Phys. Commun.* **248** 107006

[187] Cerezo M, Verdon G, Huang H-Y, Cincio L and Coles P J 2022 Challenges and opportunities in quantum machine learning *Nat. Comput. Sci.* **2** 567–76

[188] Holmes Z, Sharma K, Cerezo M and Coles P J 2022 Connecting ansatz expressibility to gradient magnitudes and barren plateaus *PRX Quantum* **3** 010313

[189] Benedetti M, Lloyd E, Sack S and Fiorentini M 2019 Parameterized quantum circuits as machine learning models *Quantum Sci. Technol.* **4** 043001

[190] Du Y, Hsieh M-H, Liu T and Tao D 2020 Expressive power of parametrized quantum circuits *Phys. Rev. Res.* **2** 033125

[191] Du Y, Tu Z, Yuan X and Tao D 2022 Efficient measure for the expressivity of variational quantum algorithms *Phys. Rev. Lett.* **128** 080506

[192] McClean J R, Boixo S, Smelyanskiy V N, Babbush R and Neven H 2018 Barren plateaus in quantum neural network training landscapes *Nat. Commun.* **9** 4812

[193] Wang S, Fontana E, Cerezo M, Sharma K, Sone A, Cincio L and Coles P J 2021 Noise-induced barren plateaus in variational quantum algorithms *Nat. Commun.* **12** 6961

[194] Arrasmith A, Holmes Z, Cerezo M and Coles P J 2022 Equivalence of quantum barren plateaus to cost concentration and narrow gorges *Quantum Sci. Technol.* **7** 045015

[195] Larocca M, Czarnik P, Sharma K, Muraleedharan G, Coles P J and Cerezo M 2022 Diagnosing barren plateaus with tools from quantum optimal control *Quantum* **6** 824

[196] Holmes Z, Arrasmith A, Yan B, Coles P J, Albrecht A and Sornborger A T 2021 Barren plateaus preclude learning scramblers *Phys. Rev. Lett.* **126** 190501

[197] Marrero C O, Kieferová M and Wiebe N 2021 Entanglement-induced barren plateaus *PRX Quantum* **2** 040316

[198] Patti T L, Najafi K, Gao X and Yelin S F 2021 Entanglement devised barren plateau mitigation *Phys. Rev. Res.* **3** 033090

[199] Sharma K, Cerezo M, Cincio L and Coles P J 2022 Trainability of dissipative perceptron-based quantum neural networks *Phys. Rev. Lett.* **128** 180505

[200] Larocca M, Thanasilp S, Wang S, Sharma K, Biamonte J, Coles P J, Cincio L, McClean J R, Holmes Z and Cerezo M 2024 A review of barren plateaus in variational quantum computing (arXiv:2405.00781)

[201] Zhang H-K, Liu S and Zhang S-X 2024 Absence of barren plateaus in finite local-depth circuits with long-range entanglement *Phys. Rev. Lett.* **132** 150603

[202] Cerezo M, Sone A, Volkoff T, Cincio L and Coles P J 2021 Cost function dependent barren plateaus in shallow parametrized quantum circuits *Nat. Commun.* **12** 1791

[203] Koczor B and Benjamin S C 2022 Quantum analytic descent *Phys. Rev. Res.* **4** 023017

[204] Wang S, Czarnik P, Arrasmith A, Cerezo M, Cincio L and Coles P J 2024 Can error mitigation improve trainability of noisy variational quantum algorithms? *Quantum* **8** 1287

[205] Endo S, Cai Z, Benjamin S C and Yuan X 2021 Hybrid quantum-classical algorithms and quantum error mitigation *J. Phys. Soc. Japan* **90** 032001

[206] Grant E, Wossnig L, Ostaszewski M and Benedetti M 2019 An initialization strategy for addressing barren plateaus in parametrized quantum circuits *Quantum* **3** 214

[207] Liu H-Y, Sun T-P, Wu Y-C, Han Y-J and Guo G-P 2023 Mitigating barren plateaus with transfer-learning-inspired parameter initializations *New J. Phys.* **25** 013039

[208] Sack S H, Medina R A, Michailidis A A, Kueng R and Serbyn M 2022 Avoiding barren plateaus using classical shadows *PRX Quantum* **3** 020365

[209] Ragone M, Bakalov B N, Sauvage F, Kemper A F, Marrero C O, Larocca M and Cerezo M 2024 A Lie algebraic theory of barren plateaus for deep parameterized quantum circuits *Nat. Commun.* **15** 7172

[210] Fontana E, Herman D, Chakrabarti S, Kumar N, Yalovetzky R, Heredge J, Sureshbabu S H and Pistoia M 2024 Characterizing barren plateaus in quantum ansätze with the adjoint representation *Nat. Commun.* **15** 7171

[211] Diaz N L, García-Martín D, Kazi S, Larocca M and Cerezo M 2023 Showcasing a barren plateau theory beyond the dynamical Lie algebra (arXiv:2310.11505)

[212] Zhang S-X, Hsieh C-Y, Zhang S and Yao H 2022 Differentiable quantum architecture search *Quantum Sci. Technol.* **7** 045023

[213] Du Y, Huang T, You S, Hsieh M-H and Tao D 2022 Quantum circuit architecture search for variational quantum algorithms *npj Quantum Inf.* **8** 62

[214] Martyniuk D, Jung J and Paschke A 2024 Quantum architecture search: a survey (arXiv:2406.06210)

[215] Sagastizabal R *et al* 2019 Experimental error mitigation via symmetry verification in a variational quantum eigensolver *Phys. Rev.* A **100** 010302

[216] Ganzhorn M *et al* 2019 Gate-efficient simulation of molecular eigenstates on a quantum computer *Phys. Rev. Appl.* **11** 044092

[217] Cade C, Mineh L, Montanaro A and Stanisic S 2020 Strategies for solving the Fermi-Hubbard model on near-term quantum computers *Phys. Rev.* B **102** 235122

[218] Larocca M, Sauvage F, Sbahi F M, Verdon G, Coles P J and Cerezo M 2022 Group-invariant quantum machine learning *PRX Quantum* **3** 030341

[219] Meyer J J, Mularski M, Gil-Fuster E, Mele A A, Arzani F, Wilms A and Eisert J 2023 Exploiting symmetry in variational quantum machine learning *PRX Quantum* **4** 010328

[220] Nguyen Q T, Schatzki L, Braccia P, Ragone M, Coles P J, Sauvage F, Larocca M and Cerezo M 2024 Theory for equivariant quantum neural networks *PRX Quantum* **5** 020328

[221] Koch C P *et al* 2022 Quantum optimal control in quantum technologies. Strategic report on current status, visions and goals for research in Europe *EPJ Quantum Technol.* **9** 19

[222] Ansel Q, Dionis E, Arrouas F, Peaudecerf B, Guérin S, Guéry-Odelin D and Sugny D 2024 Introduction to theoretical and experimental aspects of quantum optimal control *J. Phys. B: At. Mol. Opt. Phys.* **57** 133001

[223] Magann A B, Arenz C, Grace M D, Ho T-S, Kosut R L, McClean J R, Rabitz H A and Sarovar M 2021 From pulses to circuits and back again: a quantum optimal control perspective on variational quantum algorithms *PRX Quantum* **2** 010101

[224] Choquette A, Di Paolo A, Barkoutsos P K, Sénéchal D, Tavernelli I and Blais A 2021 Quantum-optimal-control-inspired ansatz for variational quantum algorithms *Phys. Rev. Res.* **3** 023092

[225] Camino B, Buckeridge J, Warburton P A, Kendon V and Woodley S M 2023 Quantum computing and materials science: a practical guide to applying quantum annealing to the configurational analysis of materials *J. Appl. Phys.* **133** 221102

[226] D-Wave 2024 D-Wave systems: Dimod (available at: https://docs.ocean.dwavesys.com/en/stable/docs_dimod/) (Accessed 1 August 2024)

[227] D-Wave 2024 D-Wave systems: Ocean (available at: https://docs.ocean.dwavesys.com/) (Accessed 1 August 2024)

[228] Mundada P S *et al* 2023 Experimental benchmarking of an automated deterministic error-suppression workflow for quantum algorithms *Phys. Rev. Appl.* **20** 024034

[229] Petitet A, Whaley R C, Dongarra J and Cleary A 2018 HPL—a portable implementation of the high-performance Linpack benchmark for distributed-memory computers (available at: https://netlib.org/benchmark/hpl/)

[230] Krantz P, Kjaergaard M, Yan F, Orlando T P, Gustavsson S and Oliver W D 2019 A quantum engineer's guide to superconducting qubits *Appl. Phys. Rev.* **6** 021318

[231] Blais A, Grimsmo A L, Girvin S M and Wallraff A 2021 Circuit quantum electrodynamics *Rev. Mod. Phys.* **93** 025005

[232] Bravyi S, Dial O, Gambetta J M, Gil D and Nazarioa Z 2022 The future of quantum computing with superconducting qubits *J. Appl. Phys.* **132** 160902

[233] Blais A, Girvin S M and Oliver W D 2020 Quantum information processing and quantum optics with circuit quantum electrodynamics *Nat. Phys.* **16** 247–56

[234] Bruzewicz C D, Chiaverini J, McConnell R and Sage J M 2019 Trapped-ion quantum computing: progress and challenges *Appl. Phys. Rev.* **6** 021314

[235] Malinowski M, Allcock D T C and Ballance C J 2023 How to wire a 1000-qubit trapped-ion quantum computer *PRX Quantum* **4** 040313

[236] Moses S A *et al* 2023 A race-track trapped-ion quantum processor *Phys. Rev.* X **13** 041052

[237] Chatterjee A, Stevenson P, De Franceschi S, Morello A, de Leon N P and Kuemmeth F 2021 Semiconductor qubits in practice *Nat. Rev. Phys.* **3** 157–77

[238] Burkard G, Ladd T D, Pan A, Nichol J M and Petta J R 2023 Semiconductor spin qubits *Rev. Mod. Phys.* **95** 025003

[239] Pelucchi E *et al* 2022 The potential and global outlook of integrated photonics for quantum technologies *Nat. Rev. Phys.* **4** 194–208

[240] Moody G *et al* 2022 2022 roadmap on integrated quantum photonics *J. Phys. Photonics* **4** 012501

[241] Adams C S, Pritchard J D and Shaffer J P 2020 Rydberg atom quantum technologies *J. Phys. B: At. Mol. Opt. Phys.* **53** 012002

[242] Surace F M, Mazza P P, Giudici G, Lerose A, Gambassi A and Dalmonte M 2020 Lattice gauge theories and string dynamics in Rydberg atom quantum simulators *Phys. Rev.* X **10** 021041

[243] Ebadi S *et al* 2021 Quantum phases of matter on a 256-atom programmable quantum simulator *Nature* **595** 227–32

[244] Campbell S and Deffner S 2017 Trade-off between speed and cost in shortcuts to adiabaticity *Phys. Rev. Lett.* **118** 100601

[245] Knill E, Leibfried D, Reichle R, Britton J, Blakestad R B, Jost J D, Langer C, Ozeri R, Seidelin S and Wineland D J 2008 Randomized benchmarking of quantum gates *Phys. Rev.* A **77** 012307

[246] Magesan E, Gambetta J M and Emerson J 2011 Scalable and robust randomized benchmarking of quantum processes *Phys. Rev. Lett.* **106** 180504

[247] Erhard A, Wallman J J, Postler L, Meth M, Stricker R, Martinez E A, Schindler P, Monz T, Emerson J and Blatt R 2019 Characterizing large-scale quantum computers via cycle benchmarking *Nat. Commun.* **10** 5347

[248] Nielsen E, Gamble J K, Rudinger K, Scholten T, Young K and Blume-Kohout R 2021 Gate set tomography *Quantum* **5** 557

[249] Huang H-Y, Preskill J and Soleimanifar M 2024 Certifying almost all quantum states with few single-qubit measurements (arXiv:2404.07281)

[250] Liu Y, Otten M, Bassirianjahromi R, Jiang L and Fefferman B 2022 Benchmarking near-term quantum computers via random circuit sampling (arXiv:2105.05232)

[251] Proctor T J, Carignan-Dugas A, Rudinger K, Nielsen E, Blume-Kohout R and Young K 2019 Direct randomized benchmarking for multiqubit devices *Phys. Rev. Lett.* **123** 030503

[252] Proctor T, Seritan S, Rudinger K, Nielsen E, Blume-Kohout R and Young K 2022 Scalable randomized benchmarking of quantum computers using mirror circuits *Phys. Rev. Lett.* **129** 150502

[253] Mayer K *et al* 2021 Theory of mirror benchmarking and demonstration on a quantum computer (arXiv:2108.10431)

[254] Cross A W, Bishop L S, Sheldon S, Nation P D and Gambetta J M 2019 Validating quantum computers using randomized model circuits *Phys. Rev.* A **100** 032328

[255] Proctor T, Rudinger K, Young K, Nielsen E and Blume-Kohout R 2022 Measuring the capabilities of quantum computers *Nat. Phys.* **18** 75–79

[256] Mills D, Sivarajah S, Scholten T L and Duncan R 2021 Application-motivated, holistic benchmarking of a full quantum computing stack *Quantum* **5** 415

[257] Quetschlich N, Burgholzer L and Wille R 2023 MQT Bench: benchmarking software and design automation tools for quantum computing *Quantum* **7** 1062

[258] Li A, Stein S, Krishnamoorthy S and Ang J 2022 QASMBench: a low-level QASM benchmark suite for NISQ evaluation and simulation *ACM Trans. Quantum Comput.* **4** 1–26

[259] Michielsen K, Nocon M, Willsch D, Jin F, Lippert T and De Raedt H 2017 Benchmarking gate-based quantum computers *Comput. Phys. Commun.* **220** 44–55

[260] Derbyshire E, Yago Malo J, Daley A J, Kashefi E and Wallden P 2020 Randomized benchmarking in the analogue setting *Quantum Sci. Technol.* **5** 034001

[261] Georgopoulos K, Emary C and Zuliani P 2021 Quantum computer benchmarking via quantum algorithms (arXiv:2112.09457)

[262] Lubinski T, Johri S, Varosy P, Coleman J, Zhao L, Necaise J, Baldwin C H, Mayer K and Proctor T 2023 Application-oriented performance benchmarks for quantum computing *IEEE Trans. Quantum Eng.* **4** 1–32

[263] Donkers H, Mesman K, Al-Ars Z and Möller M 2022 QPack scores: quantitative performance metrics for application-oriented quantum computer benchmarking (arXiv:2205.12142)

[264] Dallaire-Demers P L, Stęchły M, Gonthier J F, Bashige N T, Romero J and Cao Y 2020 An application benchmark for fermionic quantum simulations (arXiv:2003.01862)

[265] Cornelissen A, Bausch J and Gilyén A 2021 Scalable benchmarks for gate-based quantum computers (arXiv:2104.10698)

[266] Tomesh T, Gokhale P, Omole V, Ravi G S, Smith K N, Viszlai J, Wu X C, Hardavellas N, Martonosi M R and Chong F T 2022 SupermarQ: a scalable quantum benchmark suite *2022 IEEE HPCA* pp 587–603

[267] Murali P, Linke N M, Martonosi M, Abhari A J, Nguyen N H and Alderete C H 2019 Full-stack, real-system quantum computer studies: architectural comparisons and design insights *ISCA 2019* (ACM) pp 527–40

[268] van Dam W, Hallgren S and Ip L 2006 Quantum algorithms for some hidden shift problems *SIAM J. Comput.* **36** 763–78

[269] Blume-Kohout R and Young K C 2020 A volumetric framework for quantum computer benchmarks *Quantum* **4** 362

[270] Groen D, Arabnejad H, Suleimenova D, Edeling W, Raffin E, Xue Y, Bronik K, Monnier N and Coveney P V 2022 FabSim3: an automation toolkit for verified simulations using high performance computing *Comput. Phys. Commun.* **283** 108596

[271] Ferracin S, Kapourniotis T and Datta A 2019 Accrediting outputs of noisy intermediate-scale quantum computing devices *New J. Phys.* **21** 113038

[272] Ferracin S, Merkel S T, McKay D and Datta A 2021 Experimental accreditation of outputs of noisy quantum computers *Phys. Rev.* A **104** 042603

[273] Jackson A, Kapourniotis T and Datta A 2024 Accreditation of analogue quantum simulators *Proc. Natl Acad. Sci.* **121** e2309627121

[274] Miller K, Broomfield C, Cox A, Kinast J and Rodenburg B 2022 An improved volumetric metric for quantum computers via more representative quantum circuit shapes (arXiv:2207.02315)

[275] Wack A, Paik H, Javadi-Abhari A, Jurcevic P, Faro I, Gambetta J M and Johnson B R 2021 Quality, speed, and scale: three key attributes to measure the performance of near-term quantum computers (arXiv:2110.14108)

[276] McKay D C, Hincks I, Pritchett E J, Carroll M, Govia L C G and Merkel S T 2023 Benchmarking quantum processor performance at scale (arXiv:2311.05933)

[277] McGeoch C C 2019 Principles and guidelines for quantum performance analysis *QTOP 2019* ed S Feld and C Linnhoff-Popien (Springer) pp 36–48

[278] Liu W and Yin H-W 2022 A quantum scheme of state overlap based on quantum mean estimation and support vector machine *Physica* A **606** 128117

[279] Foss-Feig M, Hayes D, Dreiling J M, Figgatt C, Gaebler J P, Moses S A, Pino J M and Potter A C 2021 Holographic quantum algorithms for simulating correlated spin systems *Phys. Rev. Res.* **3** 033002

[280] Foss-Feig M *et al* 2022 Entanglement from tensor networks on a trapped-ion quantum computer *Phys. Rev. Lett.* **128** 150504

[281] Villalonga B, Boixo S, Nelson B, Henze C, Rieffel E, Biswas R and Mandrà S 2019 A flexible high-performance simulator for verifying and benchmarking quantum circuits implemented on real hardware *npj Quantum Inf.* **5** 86

[282] Boixo S, Isakov S V, Smelyanskiy V N, Babbush R, Ding N, Jiang Z, Bremner M J, Martinis J M and Neven H 2018 Characterizing quantum supremacy in near-term devices *Nat. Phys.* **14** 595–600

[283] Bouland A, Fefferman B, Nirkhe C and Vazirani U 2019 On the complexity and verification of quantum random circuit sampling *Nat. Phys.* **15** 159–63

[284] Pan F and Zhang P 2022 Simulation of quantum circuits using the big-batch tensor network method *Phys. Rev. Lett.* **128** 030501

[285] Pan F, Chen K and Zhang P 2022 Solving the sampling problem of the Sycamore quantum circuits *Phys. Rev. Lett.* **129** 090502

[286] Zhang S-X *et al* 2023 TensorCircuit: a quantum software framework for the NISQ era *Quantum* **7** 912

[287] Zhou Y, Stoudenmire E M and Waintal X 2020 What limits the simulation of quantum computers? *Phys. Rev.* X **10** 041038

[288] Wahl T B and Strelchuk S 2023 Simulating quantum circuits using efficient tensor network contraction algorithms with subexponential upper bound *Phys. Rev. Lett.* **131** 180601

[289] Huang C *et al* 2021 Efficient parallelization of tensor network contraction for simulating quantum computation *Nat. Comput. Sci.* **1** 578–87

[290] Orús R 2014 A practical introduction to tensor networks: matrix product states and projected entangled pair states *Ann. Phys., NY* **349** 117–58

[291] Orús R 2019 Tensor networks for complex quantum systems *Nat. Rev. Phys.* **1** 538–50

[292] Ran S-J, Tirrito E, Peng C, Chen X, Tagliacozzo L, Su G and Lewenstein M 2021 *Tensor Network Contractions: Methods and Applications to Quantum Many-Body Systems* (Springer) (https://doi.org/10.1007/978-3-030-34489-4)

[293] Tang E 2022 Dequantizing algorithms to understand quantum advantage in machine learning *Nat. Rev. Phys.* **4** 692–3

[294] Tang E 2019 A quantum-inspired classical algorithm for recommendation systems *Proc. 51st ACM SIGACT STOC* (ACM) pp 217–28

[295] Zhong H-S *et al* 2020 Quantum computational advantage using photons *Science* **370** 1460–3

[296] Bulmer J F F *et al* 2022 The boundary for quantum advantage in Gaussian boson sampling *Sci. Adv.* **8** eabl9236

[297] Oh C, Liu M, Alexeev Y, Fefferman B and Jiang L 2024 Classical algorithm for simulating experimental Gaussian boson sampling *Nat. Phys.* **20** 1461–8

[298] Bouland A, Fefferman B, Landau Z and Liu Y 2022 Noise and the frontier of quantum supremacy *IEEE 62nd FOCS* pp 1308–17

[299] Ryan-Anderson C *et al* 2021 Realization of real-time fault-tolerant quantum error correction *Phys. Rev.* X **11** 041058

[300] Sivak V V *et al* 2023 Real-time quantum error correction beyond break-even *Nature* **616** 50–55

[301] Weimer H, Kshetrimayum A and Orús R 2021 Simulation methods for open quantum many-body systems *Rev. Mod. Phys.* **93** 015008

[302] Daley A J, Bloch I, Kokail C, Flannigan S, Pearson N, Troyer M and Zoller P 2022 Practical quantum advantage in quantum simulation *Nature* **607** 667–76

[303] Trotzky S, Chen Y-A, Flesch A, McCulloch I P, Schollwöck U, Eisert J and Bloch I 2012 Probing the relaxation towards equilibrium in an isolated strongly correlated one-dimensional Bose gas *Nat. Phys.* **8** 325–30

[304] Choi J-Y, Hild S, Zeiher J, Schauß P, Rubio-Abadal A, Yefsah T, Khemani V, Huse D A, Bloch I and Gross C 2016 Exploring the many-body localization transition in two dimensions *Science* **352** 1547–52

[305] Argüello-Luengo J, González-Tudela A, Shi T, Zoller P and Cirac J I 2019 Analogue quantum chemistry simulation *Nature* **574** 215–8

[306] Somaroo S, Tseng C H, Havel T F, Laflamme R and Cory D G 1999 Quantum simulations on a quantum computer *Phys. Rev. Lett.* **82** 5381–4

[307] Hatano N and Suzuki M 2005 Finding exponential product formulas of higher orders *Quantum Annealing and Other Optimization Methods* ed A Das and B K Chakrabarti (Springer) ch 2, pp 37–68

[308] Tacchino F, Chiesa A, Carretta S and Gerace D 2020 Quantum computers as universal quantum simulators: state-of-the-art and perspectives *Adv. Quantum Technol.* **3** 1900052

[309] Benedetti M, Fiorentini M and Lubasch M 2021 Hardware-efficient variational quantum algorithms for time evolution *Phys. Rev. Res.* **3** 033083

[310] McKeever C and Lubasch M 2023 Classically optimized Hamiltonian simulation *Phys. Rev. Res.* **5** 023146

[311] Foulkes W M C, Mitas L, Needs R J and Rajagopal G 2001 Quantum Monte Carlo simulations of solids *Rev. Mod. Phys.* **73** 33–83

[312] Becca F and Sorella S 2017 *Quantum Monte Carlo Approaches for Correlated Systems* (Cambridge University Press) (https://doi.org/10.1017/9781316417041)

[313] Mazzola G 2024 Quantum computing for chemistry and physics applications from a Monte Carlo perspective *J. Chem. Phys.* **160** 010901

[314] Carlson J, Gandolfi S, Pederiva F, Pieper S C, Schiavilla R, Schmidt K E and Wiringa R B 2015 Quantum Monte Carlo methods for nuclear physics *Rev. Mod. Phys.* **87** 1067–118

[315] Cohen G, Gull E, Reichman D R and Millis A J 2015 Taming the dynamical sign problem in real-time evolution of quantum many-body problems *Phys. Rev. Lett.* **115** 266802

[316] Li Z-X and Yao H 2019 Sign-problem-free fermionic quantum Monte Carlo: developments and applications *Annu. Rev. Condens. Matter Phys.* **10** 337–56

[317] Alexandru A, Başar G, Bedaque P F and Warrington N C 2022 Complex paths around the sign problem *Rev. Mod. Phys.* **94** 015006

[318] Bakr W S, Gillen J I, Peng A, Fölling S and Greiner M 2009 A quantum gas microscope for detecting single atoms in a Hubbard-regime optical lattice *Nature* **462** 74–77

[319] Cheuk L W, Nichols M A, Okan M, Gersdorf T, Ramasesh V V, Bakr W S, Lompe T and Zwierlein M W 2015 Quantum-gas microscope for fermionic atoms *Phys. Rev. Lett.* **114** 193001

[320] Byrnes T and Yamamoto Y 2006 Simulating lattice gauge theories on a quantum computer *Phys. Rev.* A **73** 022328

[321] Zohar E and Reznik B 2011 High-temperature superconductivity *Phys. Rev. Lett.* **107** 275301

[322] Büchler H P, Hermele M, Huber S D, Fisher M P A and Zoller P 2005 Atomic quantum simulator for lattice gauge theories and ring exchange models *Phys. Rev. Lett.* **95** 040402

[323] Cloët I C *et al* 2019 Opportunities for nuclear physics & quantum information science (arXiv:1903.05453)

[324] Bauer C W *et al* 2023 Quantum simulation for high-energy physics *PRX Quantum* **4** 027001

[325] Anderson M H, Ensher J R, Matthews M R, Wieman C E and Cornell E A 1995 Observation of Bose-Einstein condensation in a dilute atomic vapor *Science* **269** 198–201

[326] DeMarco B and Jin D S 1999 Onset of Fermi degeneracy in a trapped atomic gas *Science* **285** 1703–6

[327] Schäfer F, Fukuhara T, Sugawa S, Takasu Y and Takahashi Y 2020 Tools for quantum simulation with ultracold atoms in optical lattices *Nat. Rev. Phys.* **2** 411–25

[328] Daley A J 2023 Twenty-five years of analogue quantum simulation *Nat. Rev. Phys.* **5** 702–3

[329] Martinez E A *et al* 2016 Real-time dynamics of lattice gauge theories with a few-qubit quantum computer *Nature* **534** 516–9

[330] Atas Y Y, Zhang J, Lewis R, Jahanpour A, Haase J F and Muschik C A 2021 SU(2) hadrons on a quantum computer via a variational approach *Nat. Commun.* **12** 6499

[331] Irmejs R, Bañuls M C and Cirac J I 2023 Quantum simulation of $\mathbb{Z}_2$ lattice gauge theory with minimal resources *Phys. Rev.* D **108** 074503

[332] Abrams D S and Lloyd S 1997 Simulation of many-body Fermi systems on a universal quantum computer *Phys. Rev. Lett.* **79** 2586

[333] Ortiz G, Gubernatis J E, Knill E and Laflamme R 2001 Quantum algorithms for fermionic simulations *Phys. Rev.* A **64** 022319

[334] Lin L and Tong Y 2022 Heisenberg-limited ground-state energy estimation for early fault-tolerant quantum computers *PRX Quantum* **3** 010318

[335] Huggins W J, O'Gorman B A, Rubin N C, Reichman D R, Babbush R and Lee J 2022 Unbiasing fermionic quantum Monte Carlo with a quantum computer *Nature* **603** 416–20

[336] Somma R D 2019 Quantum eigenvalue estimation via time series analysis *New J. Phys.* **21** 123025

[337] Cao Y *et al* 2019 Quantum chemistry in the age of quantum computing *Chem. Rev.* **119** 10856

[338] McArdle S, Endo S, Aspuru-Guzik A, Benjamin S C and Yuan X 2020 Quantum computational chemistry *Rev. Mod. Phys.* **92** 015003

[339] Lee S *et al* 2023 Evaluating the evidence for exponential quantum advantage in ground-state quantum chemistry *Nat. Commun.* **14** 1952

[340] Chen J, Cheng H-P and Freericks J K 2021 Quantum-inspired algorithm for the factorized form of unitary coupled cluster theory *J. Chem. Theory Comput.* **17** 841–7

[341] Grimme S and Schreiner P R 2018 Computational chemistry: the fate of current methods and future challenges *Angew. Chem., Int. Ed.* **57** 4170–6

[342] Lidar D 2004 On the quantum computational complexity of the Ising spin glass partition function and of knot invariants *New J. Phys.* **6** 167

[343] Jaeger F, Vertigan D L and Welsh D J A 1990 On the computational complexity of the Jones and Tutte polynomials *Math. Proc. Camb. Phil. Soc.* **108** 35–53

[344] Temme K, Osborne T J, Vollbrecht K G, Poulin D and Verstraete F 2011 Quantum metropolis sampling *Nature* **471** 87–90

[345] Wocjan P, Chiang C-F, Nagaj D and Abeyesinghe A 2009 Quantum algorithm for approximating partition functions *Phys. Rev.* A **80** 022340

[346] Arunachalam S, Havlicek V, Nannicini G, Temme K and Wocjan P 2022 Simpler (classical) and faster (quantum) algorithms for Gibbs partition functions *Quantum* **6** 789

[347] Wu Y and Wang J B 2021 Estimating Gibbs partition function with quantum Clifford sampling *Quantum Sci. Technol.* **7** 025006

[348] Jackson A, Kapourniotis T and Datta A 2023 Partition-function estimation: quantum and quantum-inspired algorithms *Phys. Rev.* A **107** 012421

[349] Tubman N M *et al* 2018 Postponing the orthogonality catastrophe: efficient state preparation for electronic structure simulations on quantum devices (arXiv: 1809.05523)

[350] Slater J C 1951 A simplification of the Hartree-Fock method *Phys. Rev.* **81** 385

[351] Hohenberg P and Kohn W 1964 Inhomogeneous electron gas *Phys. Rev.* **136** B864

[352] Kohn W and Sham L J 1965 Self-consistent equations including exchange and correlation effects *Phys. Rev.* **140** A1133

[353] Echenique P and Alonso J L 2007 A mathematical and computational review of Hartree-Fock SCF methods in quantum chemistry *Mol. Phys.* **105** 3057–98

[354] Molecular Sciences Software Institute (MolSSI) and Pacific Northwest National Lab/Environmental Molecular Sciences Laboratory (PNNL/EMSL) 2023 Basis set exchange (available at: www.basissetexchange.org/) (Accessed 1 September 2023)

[355] Singh D J and Nordström L 2006 *Planewaves, Pseudopotentials and the LAPW Method* (Springer) (https://doi.org/10.1007/978-0-387-29684-5)

[356] Peterson K A 2007 Gaussian basis sets exhibiting systematic convergence to the complete basis set limit *Annu. Rep. Comput. Chem.* **3** 195–206

[357] Jensen S R, Fløa T, Jonsson D, Monstad R S, Ruud K and Frediani L 2016 Magnetic properties with multiwavelets and DFT: the complete basis set limit achieved *Phys. Chem. Chem. Phys.* **18** 21145–61

[358] Chien R W and Klassen J 2022 Optimizing fermionic encodings for both Hamiltonian and hardware (arXiv:2210.05652)

[359] Capelle K 2006 A bird's-eye view of density-functional theory *Braz. J. Phys.* **36** 1318–43

[360] Baseden K A and Tye J W 2014 Introduction to density functional theory: calculations by hand on the helium atom *J. Chem. Educ.* **91** 2116–23

[361] Eriksen J J and Gauss J 2021 Incremental treatments of the full configuration interaction problem *Wiley Interdiscip. Rev.-Comput. Mol. Sci.* **11** e1525

[362] Møller C and Plesset M S 1934 Note on an approximation treatment for many-electron systems *Phys. Rev.* **46** 618–22

[363] Čížek J 1966 On the correlation problem in atomic and molecular systems. Calculation of wavefunction components in Ursell-type expansions using quantum-field theoretical methods *J. Chem. Phys.* **45** 4256–66

[364] Hoffmann R 1987 How chemistry and physics meet in the solid state *Angew. Chem., Int. Ed. Engl.* **26** 846–78

[365] Wannier G H 1937 The structure of electronic excitation levels in insulating crystals *Phys. Rev.* **52** 191–7

[366] Clinton L, Cubitt T, Flynn B, Gambetta F M, Klassen J, Montanaro A, Piddock S, Santos R A and Sheridan E 2024 Towards near-term quantum simulation of materials *Nat. Commun.* **15** 211

[367] Kassal I, Whitfield J D, Perdomo-Ortiz A, Yung M-H and Aspuru-Guzik A 2011 Simulating chemistry using quantum computers *Annu. Rev. Phys. Chem.* **62** 185–207

[368] Bauer B, Bravyi S, Motta M and Chan G K-L 2020 Quantum algorithms for quantum chemistry and quantum materials science *Chem. Rev.* **120** 12685–717

[369] Bernal D E, Ajagekar A, Harwood S M, Stober S T, Trenev D and You F 2022 Perspectives of quantum computing for chemical engineering *AIChE J.* **68** e17651

[370] Jordan P and Wigner E 1928 Über das Paulische äquivalenzverbot *Z. Phys.* **47** 631–51

[371] Bravyi S B and Kitaev A Y 2002 Fermionic quantum computation *Ann. Phys., NY* **298** 210–26

[372] Tranter A, Love P J, Mintert F and Coveney P V 2018 A comparison of the Bravyi–Kitaev and Jordan–Wigner transformations for the quantum simulation of quantum chemistry *J. Chem. Theory Comput.* **14** 5617–30

[373] O'Brien O and Strelchuk S 2024 Ultrafast hybrid fermion-to-qubit mapping *Phys. Rev.* B **109** 115149

[374] Derby C, Klassen J, Bausch J and Cubitt T 2021 Compact fermion to qubit mappings *Phys. Rev.* B **104** 035118

[375] Wecker D, Bauer B, Clark B K, Hastings M B and Troyer M 2014 Gate-count estimates for performing quantum chemistry on small quantum computers *Phys. Rev.* A **90** 022305

[376] Babbush R, Gidney C, Berry D W, Wiebe N, McClean J, Paler A, Fowler A and Neven H 2018 Encoding electronic spectra in quantum circuits with linear T complexity *Phys. Rev.* X **8** 041015

[377] Bartlett R J, Kucharski S A and Noga J 1989 Alternative coupled-cluster ansätze II. The unitary coupled-cluster method *Chem. Phys. Lett.* **155** 133–40

[378] Motta M, Ye E, McClean J R, Li Z, Minnich A J, Babbush R and Chan G K-L 2021 Low rank representations for quantum simulation of electronic structure *npj Quantum Inf.* **7** 83

[379] Matsuzawa Y and Kurashige Y 2020 Jastrow-type decomposition in quantum chemistry for low-depth quantum circuits *J. Chem. Theory Comput.* **16** 944–52

[380] Setia K, Bravyi S, Mezzacapo A and Whitfield J D 2019 Superfast encodings for fermionic quantum simulation *Phys. Rev. Res.* **1** 033033

[381] Wecker D, Hastings M B and Troyer M 2015 Progress towards practical quantum variational algorithms *Phys. Rev.* A **92** 042303

[382] Wiersema R, Zhou C, de Sereville Y, Carrasquilla J F, Kim Y B and Yuen H 2020 Exploring entanglement and optimization within the Hamiltonian variational ansatz *PRX Quantum* **1** 020319

[383] Grimsley H R, Economou S E, Barnes E and Mayhall N J 2019 An adaptive variational algorithm for exact molecular simulations on a quantum computer *Nat. Commun.* **10** 3007

[384] Tang H L, Shkolnikov V O, Barron G S, Grimsley H R, Mayhall N J, Barnes E and Economou S E 2021 Qubit-ADAPT-VQE: an adaptive algorithm for constructing hardware-efficient ansätze on a quantum processor *PRX Quantum* **2** 020310

[385] Yordanov Y S, Armaos V, Barnes C H W and Arvidsson-Shukur D R M 2021 Qubit-excitation-based adaptive variational quantum eigensolver *Commun. Phys.* **4** 228

[386] Takeshita T, Rubin N C, Jiang Z, Lee E, Babbush R and McClean J R 2020 Increasing the representation accuracy of quantum simulations of chemistry without extra quantum resources *Phys. Rev.* X **10** 011004

[387] Sokolov I O, Barkoutsos P K I, Ollitrault P J, Greenberg D, Rice J, Pistoia M and Tavernelli I 2020 Quantum orbital-optimized unitary coupled cluster methods in the strongly correlated regime: can quantum algorithms outperform their classical equivalents? *J. Chem. Phys.* **152** 124107

[388] Vaquero-Sabater N, Carreras A, Orús R, Mayhall N J and Casanova D 2024 Physically motivated improvements of variational quantum eigensolvers *J. Chem. Theory Comput.* **20** 5133–44

[389] Grimsley H R, Barron G S, Barnes E, Economou S E and Mayhall N J 2023 Adaptive, problem-tailored variational quantum eigensolver mitigates rough parameter landscapes and barren plateaus *npj Quantum Inf.* **9** 19

[390] Feniou C, Hassan M, Traoré D, Giner E, Maday Y and Piquemal J-P 2023 Overlap-ADAPT-VQE: practical quantum chemistry on quantum computers via overlap-guided compact Ansätze *Commun. Phys.* **6** 192

[391] Markov I L and Shi Y 2008 Simulating quantum computation by contracting tensor networks *SIAM J. Comput.* **38** 963–81

[392] Liu J-G, Zhang Y-H, Wan Y and Wang L 2019 Variational quantum eigensolver with fewer qubits *Phys. Rev. Res.* **1** 023025

[393] Barratt F, Dborin J, Bal M, Stojevic V, Pollmann F and Green A G 2021 Parallel quantum simulation of large systems on small NISQ computers *npj Quantum Inf.* **7** 79

[394] Fujii K, Mizuta K, Ueda H, Mitarai K, Mizukami W and Nakagawa Y O 2022 Deep variational quantum eigensolver: a divide-and-conquer method for solving a larger problem with smaller size quantum computers *PRX Quantum* **3** 010346

[395] Catlow C R A, Buckeridge J, Farrow M R, Logsdail A J and Sokol A A 2017 Quantum mechanical/molecular mechanical (QM/MM) approaches *Handbook of Solid State Chemistry* ed R Dronskowski, S Kikkawa and A Stein (Wiley) (https://doi.org/10.1002/9783527691036.hsscvol5012)

[396] Berger D, Logsdail A J, Oberhofer H, Farrow M R, Catlow C R A, Sherwood P, Sokol A A, Blum V and Reuter K 2014 Embedded-cluster calculations in a numeric atomic orbital density-functional theory framework *J. Chem. Phys.* **141** 024105

[397] Knizia G and Chan G K-L 2012 Density matrix embedding: a simple alternative to dynamical mean-field theory *Phys. Rev. Lett.* **109** 186404

[398] Anisimov V I, Poteryaev A I, Korotin M A, Anokhin A O and Kotliar G 1997 First-principles calculations of the electronic structure and spectra of strongly correlated systems: dynamical mean-field theory *J. Phys.: Condens. Matter* **9** 7359

[399] Gao Q, Nakamura H, Gujarati T P, Jones G O, Rice J E, Wood S P, Pistoia M, Garcia J M and Yamamoto N 2021 Computational investigations of the lithium superoxide dimer rearrangement on noisy quantum devices *J. Phys. Chem.* A **125** 1827–36

[400] Gujarati T P *et al* 2023 Quantum computation of reactions on surfaces using local embedding *npj Quantum Inf.* **9** 88

[401] Westermayr J and Marquetand P 2021 Machine learning for electronically excited states of molecules *Chem. Rev.* **121** 9873–926

[402] Ollitrault P J, Mazzola G and Tavernelli I 2020 Nonadiabatic molecular quantum dynamics with quantum computers *Phys. Rev. Lett.* **125** 260511

[403] Colless J I, Ramasesh V V, Dahlen D, Blok M S, Kimchi-Schwartz M E, McClean J R, Carter J, de Jong W A and Siddiqi I 2018 Computation of molecular spectra on a quantum processor with an error-resilient algorithm *Phys. Rev.* X **8** 011021

[404] Ollitrault P J, Miessen A and Tavernelli I 2021 Molecular quantum dynamics: a quantum computing perspective *Acc. Chem. Res.* **54** 4229–38

[405] Babbush R, McClean J, Wecker D, Aspuru-Guzik A and Wiebe N 2015 Chemical basis of Trotter-Suzuki errors in quantum chemistry simulation *Phys. Rev.* A **91** 022311

[406] Miessen A, Ollitrault P J and Tavernelli I 2021 Quantum algorithms for quantum dynamics: a performance study on the spin-boson model *Phys. Rev. Res.* **3** 043212

[407] Fedorov A K and Gelfand M S 2021 Towards practical applications in quantum computational biology *Nat. Comput. Sci.* **1** 114–9

[408] Teplukhin A, Kendrick B K, Tretiak S and Dub P A 2020 Electronic structure with direct diagonalization on a D-Wave quantum annealer *Sci. Rep.* **10** 20753

[409] Teplukhin A, Kendrick B K, Mniszewski S M, Zhang Y, Kumar A, Negre C F A, Anisimov P M, Tretiak S and Dub P A 2021 Computing molecular excited states on a D-Wave quantum annealer *Sci. Rep.* **11** 18796

[410] Streif M, Neukart F and Leib M 2019 Solving quantum chemistry problems with a D-Wave quantum annealer *QTOP 2019* (Springer) pp 111–22

[411] Genin S N, Ryabinkin I G and Izmaylov A F 2019 Quantum chemistry on quantum annealers (arXiv:1901.04715)

[412] Mulligan V K *et al* 2019 Designing peptides on a quantum computer *bioRxiv Preprint* https://doi.org/10.1101/752485 (accessed 01 April 2023)

[413] Fox D M, Branson K M and Walker R C 2021 mRNA codon optimization with quantum computers *PLoS One* **16** e0259101

[414] Hatakeyama-Sato K, Kashikawa T, Kimura K and Oyaizu K 2021 Tackling the challenge of a huge materials science search space with quantum-inspired annealing *Adv. Intell. Syst.* **3** 2000209

[415] Gao Q, Jones G O, Kobayashi T, Sugawara M, Yamashita H, Kawaguchi H, Tanaka S and Yamamoto N 2023 Quantum-classical computational molecular design of deuterated high-efficiency OLED emitters *Intell. Comput.* **2** 0037

[416] Bednorz J G and Müller K A 1986 Possible high T_c superconductivity in the Ba-La-Cu-O system *Z. Phys.* B **64** 189–93

[417] Sheng Z Z and Hermann A M 1988 Bulk superconductivity at 120 K in the Tl-Ca/Ba-Cu-O system *Nature* **332** 138–9

[418] Zhou X, Lee W-S, Imada M, Trivedi N, Phillips P, Kee H-Y, Törmä P and Eremets M 2021 High-temperature superconductivity *Nat. Rev. Phys.* **3** 462–5

[419] Lilia B *et al* 2022 The 2021 room-temperature superconductivity roadmap *J. Phys.: Condens. Matter* **34** 183002

[420] Bardeen J, Cooper L N and Schrieffer J R 1957 Microscopic theory of superconductivity *Phys. Rev.* **106** 162–4

[421] Bardeen J, Cooper L N and Schrieffer J R 1957 Theory of superconductivity *Phys. Rev.* **108** 1175–204

[422] International Energy Agency 2021 Net Zero by 2050 *Flagship Report* (IEA) (available at: www.iea.org/reports/net-zero-by-2050) (Accessed 1 June 2023)

[423] Ball P 2021 The chase for fusion energy *Nature* **599** 362–6

[424] Dagotto E 1994 Correlated electrons in high-temperature superconductors *Rev. Mod. Phys.* **66** 763–840

[425] Fradkin E, Kivelson S A and Tranquada J M 2015 Colloquium: Theory of intertwined orders in high temperature superconductors *Rev. Mod. Phys.* **87** 457–82

[426] LeBlanc J P F *et al* 2015 Solutions of the two-dimensional Hubbard model: benchmarks and results from a wide range of numerical algorithms *Phys. Rev.* X **5** 041041

[427] Hensgens T, Fujita T, Janssen L, Li X, Van Diepen C J, Reichl C, Wegscheider W, Das Sarma S and Vandersypen L M K 2017 Quantum simulation of a Fermi-Hubbard model using a semiconductor quantum dot array *Nature* **548** 70–73

[428] Wang X, Khatami E, Fei F, Wyrick J, Namboodiri P, Kashid R, Rigosi A F, Bryant G and Silver R 2022 Experimental realization of an extended Fermi-Hubbard model using a 2D lattice of dopant-based quantum dots *Nat. Commun.* **13** 6824

[429] Mazurenko A, Chiu C S, Ji G, Parsons M F, Kanász-Nagy M, Schmidt R, Grusdt F, Demler E, Greif D and Greiner M 2017 A cold-atom Fermi-Hubbard antiferromagnet *Nature* **545** 462–6

[430] Hirthe S, Chalopin T, Bourgund D, Bojović P, Bohrdt A, Demler E, Grusdt F, Bloch I and Hilker T A 2023 Magnetically mediated hole pairing in fermionic ladders of ultracold atoms *Nature* **413** 463–7

[431] Cai Z 2020 Resource estimation for quantum variational simulations of the Hubbard model *Phys. Rev. Appl.* **14** 014059

[432] Stanisic S, Bosse J L, Gambetta F M, Santos R A, Mruczkiewicz W, O'Brien T E, Ostby E and Montanaro A 2022 Observing ground-state properties of the Fermi-Hubbard model using a scalable algorithm on a quantum computer *Nat. Commun.* **13** 5743

[433] Google Quantum AI and Collaborators 2020 Observation of separated dynamics of charge and spin in the Fermi-Hubbard model (arXiv:2010.07965)

[434] Kivlichan I D *et al* 2020 Improved fault-tolerant quantum simulation of condensed-phase correlated electrons via trotterization *Quantum* **4** 296

[435] Clinton L, Bausch J and Cubitt T 2021 Hamiltonian simulation algorithms for near-term quantum hardware *Nat. Commun.* **12** 4989

[436] Drozdov A P, Eremets M I, Troyan I A, Ksenofontov V and Shylin S I 2015 Conventional superconductivity at 203 kelvin at high pressures in the sulfur hydride system *Nature* **525** 73–76

[437] Castelvecchi D 2023 Why superconductor research is in a 'golden age'—despite controversy (https://doi.org/10.1038/d41586-023-03551-z)

[438] Li Y, Hao J, Liu H, Li Y and Ma Y 2014 The metallization and superconductivity of dense hydrogen sulfide *J. Chem. Phys.* **140** 174712

[439] Xiao B, He Y-Y, Georges A and Zhang S 2023 Temperature dependence of spin and charge orders in the doped two-dimensional Hubbard model *Phys. Rev.* X **13** 011007

[440] Xu H, Chung C-M, Qin M, Schollwöck U, White S R and Zhang S 2024 Coexistence of superconductivity with partially filled stripes in the Hubbard model *Science* **384** eadh7691

[441] Xu M, Kendrick L H, Kale A, Gang Y, Ji G, Scalettar R T, Lebrat M and Greiner M 2023 Frustration- and doping-induced magnetism in a Fermi–Hubbard simulator *Nature* **620** 971–6

[442] Homeier L, Harris T J, Blatz T, Geier S, Hollerith S, Schollwöck U, Grusdt F and Bohrdt A 2024 Antiferromagnetic bosonic $t-J$ models and their quantum simulation in tweezer arrays *Phys. Rev. Lett.* **132** 230401

[443] Bohrdt A, Homeier L, Reinmoser C, Demler E and Grusdt F 2021 Exploration of doped quantum magnets with ultracold atoms *Ann. Phys., NY* **435** 168651

[444] Lv C *et al* 2021 Selective electrocatalytic synthesis of urea with nitrate and carbon dioxide *Nat. Sustain.* **4** 868–76

[445] Boston Consulting Group 2020 A quantum advantage in fighting climate change (available at: www.bcg.com/publications/2020/quantum-advantage-fighting-climate-change) (Accessed 1 December 2022)

[446] Burgess B K and Lowe D J 1996 Mechanism of molybdenum nitrogenase *Chem. Rev.* **96** 2983–3012

[447] Vogiatzis K D, Polynski M V, Kirkland J K, Townsend J, Hashemi A, Liu C and Pidko E A 2019 Computational approach to molecular catalysis by 3d transition metals: challenges and opportunities *Chem. Rev.* **119** 2453–523

[448] Hoffman B M, Lukoyanov D, Yang Z-Y, Dean D R and Seefeldt L C 2014 Mechanism of nitrogen fixation by nitrogenase: the next stage *Chem. Rev.* **114** 4041–62

[449] Fowler A G, Mariantoni M, Martinis J M and Cleland A N 2012 Surface codes: towards practical large-scale quantum computation *Phys. Rev.* A **86** 032324

[450] Lee J, Berry D W, Gidney C, Huggins W J, McClean J R, Wiebe N and Babbush R 2021 Even more efficient quantum computations of chemistry through tensor hypercontraction *PRX Quantum* **2** 030305

[451] Low G H, Kliuchnikov V and Schaeffer L 2024 Trading T-gates for dirty qubits in state preparation and unitary synthesis *Quantum* **8** 1375

[452] Aharonov D and Ta-Shma A 2003 Adiabatic quantum state generation and statistical zero knowledge *Proc. 35rd ACM STOC* (ACM) pp 20–29

[453] Childs A M and Wiebe N 2012 Hamiltonian simulation using linear combinations of unitary operations *Quantum Inf. Comput.* **12** 901–24

[454] Berry D W, Childs A M, Cleve R, Kothari R and Somma R D 2015 Simulating Hamiltonian dynamics with a truncated Taylor series *Phys. Rev. Lett.* **114** 090502

[455] Szegedy M 2004 Quantum speed-up of Markov chain based algorithms *45th IEEE SFCS* pp 32–41

[456] Childs A M 2010 On the relationship between continuous-and discrete-time quantum walk *Commun. Math. Phys.* **294** 581–603

[457] Marriott C and Watrous J 2005 Quantum Arthur-Merlin games *Comput. Complex.* **14** 122–52

[458] Tazhigulov R N, Sun S-N, Haghshenas R, Zhai H, Tan A T K, Rubin N C, Babbush R, Minnich A J and Chan G K-L 2022 Simulating models of challenging correlated molecules and materials on the Sycamore quantum processor *PRX Quantum* **3** 040318

[459] Debnath L 2012 *Nonlinear Partial Differential Equations for Scientists and Engineers* 3rd edn (Springer) (https://doi.org/10.1007/978-0-8176-8265-1)

[460] Tadmore E 2012 A review of numerical methods for nonlinear partial differential equations *Bull. Am. Math. Soc.* **49** 507–54

[461] Bartels S 2015 *Numerical Methods for Nonlinear Partial Differential Equations* (Springer) (https://doi.org/10.1007/978-3-319-13797-1)

[462] Han J, Jentzen A and E W 2018 Solving high-dimensional partial differential equations using deep learning *Proc. Natl Acad. Sci.* **115** 8505–10

[463] Muther T, Dahaghi A K, Syed F I and Pham V V 2023 Physical laws meet machine intelligence: current developments and future directions *Artif. Intell. Rev.* **56** 6947–7013

[464] Biswas R *et al* 2017 A NASA perspective on quantum computing: opportunities and challenges *Parallel Comput.* **64** 81–98

[465] Rieffel E G *et al* 2019 From Ansätze to Z-gates: a NASA view of quantum computing *Future Trends of HPC in a Disruptive Scenario* (*Advances in Parallel Computing* vol 34) ed L Grandinetti, G R Joubert, K Michielsen, S L Mirtaheri, M Taufer and R Yokota (IOS Press) pp 133–60

[466] Rieffel E G *et al* 2024 Assessing and advancing the potential of quantum computing: a NASA case study *Future Gener. Comput. Syst.* **160** 598–618

[467] Moin P and Mahesh K 1998 Direct numerical simulation: a tool in turbulence research *Annu. Rev. Fluid Mech.* **30** 539–78

[468] Raissi M, Perdikaris P and Karniadakis G E 2019 Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations *J. Comput. Phys.* **379** 686–707

[469] Karniadakis G E, Kevrekidis I G, Lu L, Perdikaris P, Wang S and Yang L 2021 Physics-informed machine learning *Nat. Rev. Phys.* **3** 422–40

[470] Brunton S L, Noack B R and Koumoutsakos P 2020 Machine learning for fluid mechanics *Annu. Rev. Fluid Mech.* **52** 477–508

[471] Raissi M, Yazdani A and Karniadakis G E 2020 Hidden fluid mechanics: learning velocity and pressure fields from flow visualizations *Science* **367** 1026–30

[472] Huang S, Feng W, Tang C and Lv J 2022 Partial differential equations meet deep neural networks: a survey (arXiv:2211.05567)

[473] Fallah A and Aghdam M M 2024 Physics-informed neural network for solution of nonlinear differential equations *Nonlinear Approaches in Engineering Application: Automotive Engineering Problems* ed R N Jazar and L Dai (Springer) ch 5, pp 163–78

[474] Hafiz A M, Faiq I and Hassaballah M 2024 Solving partial differential equations using large-data models: a literature review *Artif. Intell. Rev.* **57** 152

[475] Chung T J 2002 *Computational Fluid Dynamics* (Cambridge University Press) (https://doi.org/10.1017/CBO9780511606205)

[476] Liu G R 2003 *Mesh Free Methods: Moving Beyond the Finite Element Method* (CRC Press)

[477] Monaghan J J 2005 Smoothed particle hydrodynamics *Rep. Prog. Phys.* **68** 1703–59

[478] Monaghan J J 2012 Smoothed particle hydrodynamics and its diverse applications *Annu. Rev. Fluid Mech.* **44** 323–46

[479] Lucy L B 1977 A numerical approach to the testing of the fission hypothesis *Astron. J.* **82** 1013–24

[480] Springel V 2010 Smoothed particle hydrodynamics in astrophysics *Annu. Rev. Astron. Astrophys.* **48** 391–430

[481] Succi S 2001 *The Lattice Boltzmann equation for Fluid Dynamics and Beyond* (Oxford University Press)

[482] Krüger T, Kusumaatmaja H, Kuzmin A, Shardt O, Silva G and Viggen E M 2017 *The Lattice Boltzmann Method* (Springer) (https://doi.org/10.1007/978-3-319-44649-3)

[483] Feichtinger C, Habich J, Köstler H, Rüde U and Aoki T 2015 Performance modeling and analysis of heterogeneous lattice Boltzmann simulations on CPU-GPU clusters *Parallel Comput.* **46** 1–13

[484] Liu S, Zou N, Cui Y and Wu W 2017 Accelerating the parallelization of lattice Boltzmann method by exploiting the temporal locality *2017 IEEE ISPA/IUCC* (IEEE) pp 1186–93

[485] Tran N P, Lee M and Choi D H 2015 Memory-efficient parallelization of 3D lattice Boltzmann flow solver on a GPU *2015 IEEE 22nd Int. Conf. HiPC* (IEEE) pp 315–24

[486] Chen S and Doolen G D 1998 Lattice Boltzmann method for fluid flows *Annu. Rev. Fluid Mech.* **30** 329–64

[487] Alowayyed S, Groen D, Coveney P V and Hoekstra A G 2017 Multiscale computing in the exascale era *J. Comput. Sci.* **22** 15–25

[488] Succi S, Amati G, Bernaschi M, Falcucci G, Lauricella M and Montessori A 2019 Towards exascale lattice Boltzmann computing *Comput. Fluids* **181** 107–15

[489] Wei X, Li W, Mueller K and Kaufman A E 2004 The lattice-Boltzmann method for simulating gaseous phenomena *IEEE Trans Viz. Comput. Graph.* **10** 164–76

[490] Krause M J *et al* 2021 OpenLB–open source lattice Boltzmann code *Comput. Math. Appl.* **81** 258–88

[491] Latt J *et al* 2021 Palabos: parallel lattice Boltzmann solver *Comput. Math. Appl.* **81** 334–50

[492] Yepez J 2001 Quantum lattice-gas model for computational fluid dynamics *Phys. Rev.* E **63** 046702

[493] Berman G P, Ezhov A A, Kamenev D I and Yepez J 2002 Simulation of the diffusion equation on a type-II quantum computer *Phys. Rev.* A **66** 012310

[494] Yepez J 2001 Type-II quantum computers *Int. J. Mod. Phys.* C **12** 1273–84

[495] Love P J and Boghosian B M 2006 Type II quantum algorithms *Physica* A **362** 210–4

[496] Kocherla S, Song Z, Chrit F E, Gard B, Dumitrescu E F, Alexeev A and Bryngelson S H 2024 Fully quantum algorithm for mesoscale fluid simulations with application to partial differential equations *AVS Quantum Sci.* **6** 033806

[497] Succi S, Fillion-Gourdeau F and Palpacelli S 2015 Quantum lattice Boltzmann is a quantum walk *EPJ Quantum Technol.* **2** 12

[498] Todorova B N and Steijl R 2020 Quantum algorithm for the collisionless Boltzmann equation *J. Comput. Phys.* **409** 109347

[499] Somma R, Ortiz G, Gubernatis J E, Knill E and Laflamme R 2002 Simulating physical phenomena by quantum networks *Phys. Rev.* A **65** 042323

[500] Itani W and Succi S 2022 Analysis of Carleman linearization of lattice Boltzmann *Fluids* **7** 24

[501] Itani W, Sreenivasan K R and Succi S 2024 Quantum algorithm for lattice Boltzmann (QALB) simulation of incompressible fluids with a nonlinear collision term *Phys. Fluids* **36** 017112

[502] Budinski L 2021 Quantum algorithm for the advection-diffusion equation simulated with the lattice Boltzmann method *Quantum Inf. Process.* **20** 57

[503] Ljubomir B 2022 Quantum algorithm for the Navier-Stokes equations by using the streamfunction-vorticity formulation and the lattice Boltzmann method *Int. J. Quantum Inf.* **20** 2150039

[504] Schalkers M A and Möller M 2024 On the importance of data encoding in quantum Boltzmann methods *Quantum Inf. Process.* **23** 20

[505] Ferrari A, Dumbser M, Toro E F and Armanini A 2009 A new 3D parallel SPH scheme for free surface flows *Comput. Fluids* **38** 1203–17

[506] Domínguez J M, Crespo A J C and Gómez-Gesteira M 2013 Optimization strategies for CPU and GPU implementations of a smoothed particle hydrodynamics method *Comput. Phys. Commun.* **184** 617–27

[507] Oger G, Le Touzé D, Guibert D, de Leffe M, Biddiscombe J, Soumagne J and Piccinali J-G 2016 On distributed memory MPI-based parallelization of SPH codes in massive HPC context *Comput. Phys. Commun.* **200** 1–14

[508] Domínguez J M *et al* 2022 DualSPHysics: from fluid dynamics to multiphysics problems *Comput. Part. Mech.* **9** 867–95

[509] Schaller M *et al* 2024 Swift: a modern highly-parallel gravity and smoothed particle hydrodynamics solver for astrophysical and cosmological applications *Mon. Not. R. Astron. Soc.* **530** 2378–419

[510] Springel V 2005 The cosmological simulation code GADGET-2 *Mon. Not. R. Astron. Soc.* **364** 1105–34

[511] Lind S J, Rogers B D and Stansby P K 2020 Review of smoothed particle hydrodynamics: towards converged Lagrangian flow modelling *Proc. R. Soc.* A **476** 20190801

[512] Vacondio R, Altomare C, De Leffe M, Hu X, Le Touzé D, Lind S, Marongiu J-C, Marrone S, Rogers B D and Souto-Iglesias A 2021 Grand challenges for smoothed particle hydrodynamics numerical schemes *Comput. Part. Mech.* **8** 575–88

[513] Au-Yeung R, Williams A J, Kendon V M and Lind S J 2024 Quantum algorithm for smoothed particle hydrodynamics *Comput. Phys. Commun.* **294** 108909

[514] Buhrman H, Cleve R, Watrous J and de Wolf R 2001 Quantum fingerprinting *Phys. Rev. Lett.* **87** 167902

[515] Fanizza M, Rosati M, Skotiniotis M, Calsamiglia J and Giovannetti V 2020 Beyond the swap test: optimal estimation of quantum state overlap *Phys. Rev. Lett.* **124** 060503

[516] Cincio L, Subaşı Y, Sornborger A T and Coles P J 2018 Learning the quantum algorithm for state overlap *New J. Phys.* **20** 113022

[517] Aaronson S and Rall P 2019 Quantum approximate counting, simplified *SOSA 2020* (SIAM) pp 24–32

[518] Domínguez J M, Crespo A J C, Gómez-Gesteira M and Marongiu J C 2011 Neighbour lists in smoothed particle hydrodynamics *Int. J. Numer. Meth. Fluids* **67** 2026–42

[519] Courant R, Friedrichs K and Lewy H 1967 On the partial difference equations of mathematical physics *IBM J. Res. Dev.* **11** 215–34

[520] Violeau D and Leroy A 2014 On the maximum time step in weakly compressible SPH *J. Comput. Phys.* **256** 388–415

[521] Violeau D and Leroy A 2015 Optimal time step for incompressible SPH *J. Comput. Phys.* **288** 119–30

[522] He L, Liu S, Gan Y, Seaid M and Niu C 2021 Development of time-space adaptive smoothed particle hydrodynamics method with Runge-Kutta Chebyshev scheme *Eng. Anal. Bound. Elem.* **126** 55–67

[523] Childs A M, Liu J-P and Ostrander A 2021 High-precision quantum algorithms for partial differential equations *Quantum* **5** 574

[524] Linden N, Montanaro A and Shao C 2022 Quantum vs. classical algorithms for solving the heat equation *Commun. Math. Phys.* **395** 601–41

[525] Liu H-L, Wu Y-S, Wan L-C, Pan S-J, Qin S-J, Gao F and Wen Q-Y 2021 Variational quantum algorithm for the Poisson equation *Phys. Rev.* A **104** 022418

[526] Sato Y, Kondo R, Koide S, Takamatsu H and Imoto N 2021 Variational quantum algorithm based on the minimum potential energy for solving the Poisson equation *Phys. Rev.* A **104** 052409

[527] Robson W, Saha K K, Howington C, Suh I S and Nabrzyski J 2022 Advanced quantum Poisson solver in the NISQ era *2022 QCE* (IEEE) pp 741–4

[528] Saha K K, Robson W, Howington C, Suh I-S, Wang Z and Nabrzyski J 2024 Enhancing scalability and accuracy of quantum Poisson solver *Quantum Inf. Process.* **23** 209

[529] Balducci G T, Chen B, Möller M, Gerritsma M and De Breuker R 2022 Review and perspectives in quantum computing for partial differential equations in structural mechanics *Front. Mech. Eng.* **8** 914241

[530] Leyton S K and Osborne T J 2008 A quantum algorithm to solve nonlinear differential equations (arXiv:0812.4423)

[531] Liu J-P, Kolden H Ø, Krovi H K, Loureiro N F, Trivisa K and Childs A M 2021 Efficient quantum algorithm for dissipative nonlinear differential equations *Proc. Natl Acad. Sci.* **118** e2026805118

[532] Gaitan F 2020 Finding flows of a Navier-Stokes fluid through quantum computing *npj Quantum Inf.* **6** 61

[533] Gaitan F 2021 Finding solutions of the Navier-Stokes equations through quantum computing–recent progress, a generalization and next steps forward *Adv. Quantum Technol.* **4** 2100055

[534] Kacewicz B Z 1987 Optimal solution of ordinary differential equations *J. Complex.* **3** 451–65

[535] Ingelman J, Bharadwaj S S, Pfeffer P, Sreenivasan K R and Schumacher J 2024 Two quantum algorithms for solving the one-dimensional advection-diffusion equation *Comput. Fluids* **281** 106369

[536] Weiss K, Khoshgoftaar T M and Wang D 2016 A survey of transfer learning *J. Big Data* **3** 9

[537] Kordzanganeh M, Sekatski P, Fedichkin L and Melnikov A 2023 An exponentially-growing family of universal quantum circuits *Mach. Learn.: Sci. Technol.* **4** 035036

[538] Sedykh A, Podapaka M, Sagingalieva A, Pinto K, Pflitsch M and Melnikov A 2024 Hybrid quantum physics-informed neural networks for simulating computational fluid dynamics in complex shapes *Mach. Learn.: Sci. Technol.* **5** 025045

[539] Jaderberg B, Gentile A A, Ghosh A, Elfving V E, Jones C, Vodola D, Manobianco J and Weiss H 2024 Potential of quantum scientific machine learning applied to weather modelling (arXiv:2404.08737)

[540] Jaksch D, Givi P, Daley A J and Rung T 2023 Variational quantum algorithms for computational fluid dynamics *AIAA J.* **61** 1885–94
[541] Mocz P and Szasz A 2021 Toward cosmological simulations of dark matter on quantum computers *Astrophys. J.* **910** 29
[542] Leong F Y, Ewe W-B and Koh D E 2024 Variational quantum evolution equation solver *Sci. Rep.* **12** 10817
[543] Sarma A, Watts T W, Moosa M, Liu Y and McMahon P L 2024 Quantum variational solving of nonlinear and multidimensional partial differential equations *Phys. Rev.* A **109** 062616
[544] Steijl R and Barakos G N 2018 Parallel evaluation of quantum algorithms for computational fluid dynamics *Comput. Fluids* **173** 22–28
[545] Gourianov N, Lubasch M, Dolgov S, van den Berg Q Y, Babaee H, Givi P, Kiffner M and Jaksch D 2022 A quantum-inspired approach to exploit turbulence structures *Nat. Comput. Sci.* **2** 30–37
[546] HPCQS 2023 GENCI/CEA, FZJ, and PASQAL announce significant milestone in hybrid computing (available at: www.hpcqs.eu/news/genci-cea-fzj-and-pasqal-announce-significant-milestone-in-hybrid-computing) (Accessed 1 December 2023)
[547] University of Innsbruck 2024 Supercomputer and quantum computer in harmony (available at: www.uibk.ac.at/en/newsroom/2024/supercomputer-and-quantum-computer-in-harmony) (Accessed 1 August 2024)
[548] Google Quantum A I 2023 Suppressing quantum errors by scaling a surface code logical qubit *Nature* **614** 676–81
[549] Madsen L S *et al* 2022 Quantum computational advantage with a programmable photonic processor *Nature* **606** 75–81
[550] Blumoff J Z *et al* 2022 Fast and high-fidelity state preparation and measurement in triple-quantum-dot spin qubits *PRX Quantum* **3** 010352
[551] Harty T P, Allcock D T C, Ballance C J, Guidoni L, Janacek H A, Linke N M, Stacey D N and Lucas D M 2014 High-fidelity preparation, gates, memory and readout of a trapped-ion quantum bit *Phys. Rev. Lett.* **113** 220501
[552] Todaro S L, Verma V B, McCormick K C, Allcock D T C, Mirin R P, Wineland D J, Nam S W, Wilson A C, Leibfried D and Slichter D H 2021 State readout of a trapped ion qubit using a trap-integrated superconducting photon detector *Phys. Rev. Lett.* **126** 010501
[553] Kwon M, Ebert M F, Walker T G and Saffman M 2017 Parallel low-loss measurement of multiple atomic qubits *Phys. Rev. Lett.* **119** 180504
[554] Istrati D *et al* 2020 Sequential generation of linear cluster states from a single photon emitter *Nat. Commun.* **11** 5501
[555] Cogan D, Su Z-E, Kenneth O and Gershoni D 2023 Deterministic generation of indistinguishable photons in a cluster state *Nat. Photon.* **17** 324–9
[556] Nickerson N H, Fitzsimons J F and Benjamin S C 2014 Freely scalable quantum technologies using cells of 5-to-50 qubits with very lossy and noisy photonic links *Phys. Rev.* X **4** 041041
[557] Zhong H-S *et al* 2021 Phase-programmable Gaussian boson sampling using stimulated squeezed light *Phys. Rev. Lett.* **127** 180502
[558] Long G-L and Sun Y 2001 Efficient scheme for initializing a quantum register with an arbitrary superposed state *Phys. Rev.* A **64** 014303
[559] Plesch M and Brukner Č 2011 Quantum-state preparation with universal gate decompositions *Phys. Rev.* A **83** 032302
[560] Sanders Y R, Low G H, Scherer A and Berry D W 2019 Black-box quantum state preparation without arithmetic *Phys. Rev. Lett.* **122** 020502
[561] Nakaji K, Uno S, Suzuki Y, Raymond R, Onodera T, Tanaka T, Tezuka H, Mitsuda N and Yamamoto N 2022 Approximate amplitude encoding in shallow parameterized quantum circuits and its application to financial market indicators *Phys. Rev. Res.* **4** 023136
[562] Mitsuda N, Ichimura T, Nakaji K, Suzuki Y, Tanaka T, Raymond R, Tezuka H, Onodera T and Yamamoto N 2024 Approximate complex amplitude encoding algorithm and its application to data classification problems *Phys. Rev.* A **109** 052423
[563] Ollitrault P J, Kandala A, Chen C-F, Barkoutsos P K, Mezzacapo A, Pistoia M, Sheldon S, Woerner S, Gambetta J M and Tavernelli I 2020 Quantum equation of motion for computing molecular excitation energies on a noisy quantum processor *Phys. Rev. Res.* **2** 043140
[564] Chan H H S, Meister R, Jones T, Tew D P and Benjamin S C 2023 Grid-based methods for chemistry simulations on a quantum computer *Sci. Adv.* **9** eabo7484
[565] Jaques S and Rattew A G 2023 QRAM: a survey and critique (arXiv:2305.10310)
[566] Hann C T, Lee G, Girvin S M and Jiang L 2021 Resilience of quantum random access memory to generic noise *PRX Quantum* **2** 020311
[567] Giovannetti V, Lloyd S and Maccone L 2008 Quantum random access memory *Phys. Rev. Lett.* **100** 160501
[568] Giovannetti V, Lloyd S and Maccone L 2008 Architectures for a quantum random access memory *Phys. Rev.* A **78** 052310
[569] Ciliberto C, Herbster M, Ialongo A D, Pontil M, Rocchetto A, Severini S and Wossnig L 2018 Quantum machine learning: a classical perspective *Proc. R. Soc.* A **474** 20170551
[570] Arunachalam S, Gheorghiu V, Jochym-O'Connor T, Mosca M and Srinivasan P V 2015 On the robustness of bucket brigade quantum RAM *New J. Phys.* **17** 123010
[571] Di Matteo O, Gheorghiu V and Mosca M 2020 Fault-tolerant resource estimation of quantum random-access memories *IEEE Trans. Quantum Eng.* **1** 1–13
[572] IEEE 2019 IEEE standard for floating-point arithmetic *IEEE Std 754-2019 (Revision of IEEE 754-2008)* pp 1–84
[573] IEEE Standards Association 2019 P7130: standard for quantum technologies definitions (available at: https://standards.ieee.org/ieee/7130/10680/) (Accessed 1 December 2023)
[574] Seidel R, Tcholtchev N, Bock S, Becker C K-U and Hauswirth M 2022 Efficient floating point arithmetic for quantum computers *IEEE Access* **10** 72400–15
[575] Gilyén A, Su Y, Low G H and Wiebe N 2019 Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics *Proc. 51st ACM SIGACT STOC* (ACM) pp 193–204
[576] Camps D and Van Beeumen R 2020 Approximate quantum circuit synthesis using block encodings *Phys. Rev.* A **102** 052411
[577] Sünderhauf C, Campbell E and Camps J 2024 Block-encoding structured matrices for data input in quantum computing *Quantum* **8** 1226
[578] Chakraborty S, Gilyén A and Jeffery S 2019 The power of block-encoded matrix powers: improved regression techniques via faster Hamiltonian simulation *46th ICALP 2019* (*LIPIcs* vol 132) ed C Baier, I Chatzigiannakis, P Flocchini and S Leonardi (Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik) pp 33:1–33:14
[579] Martyn J M, Rossi Z M, Tan A K and Chuang I L 2021 Grand unification of quantum algorithms *PRX Quantum* **2** 040203

[580] Chancellor N 2019 Domain wall encoding of discrete variables for quantum annealing and QAOA *Quantum Sci. Technol.* **4** 045004

[581] Berwald J, Chancellor N and Dridi R 2023 Understanding domain-wall encoding theoretically and experimentally *Phil. Trans. R. Soc.* A **381** 20210410

[582] Plewa J, Sieńko J and Rycerz K 2021 Variational algorithms for workflow scheduling problem in gate-based quantum devices *Comput. Inform.* **40** 897–929

[583] Barnes K M *et al* 2023 Optimising the quantum/classical interface for efficiency and portability with a multi-level hardware abstraction layer for quantum computers *EPJ Quantum Technol.* **10** 36

[584] Statista 2019 Number of qubits achieved in quantum computers by company/organization from 1998 to 2019 (available at: www.statista.com/statistics/993634/quantum-computers-by-number-of-qubits/) (Accessed 1 December 2023)

[585] Mandelbaum R 2021 Five years ago today, we put the first quantum computer on the cloud. Here's how we did it (available at: https://research.ibm.com/blog/quantum-five-years) (Accessed 1 November 2023)

[586] Castelvecchi D 2023 IBM releases first-ever 1,000-qubit quantum chip *Nature* **624** 238

[587] Leu A D, Gely M F, Weber M A, Smith M C, Nadlinger D P and Lucas D M 2023 Fast, high-fidelity addressed single-qubit gates using efficient composite pulse sequences *Phys. Rev. Lett.* **131** 120601

[588] Cong I, Levine H, Keesling A, Bluvstein D, Wang S-T and Lukin M D 2022 Hardware-efficient, fault-tolerant quantum computation with Rydberg atoms *Phys. Rev.* X **12** 021049

[589] Wintersperger K, Dommert F, Ehmer T, Hoursanov A, Klepsch J, Mauerer W, Reuber G, Strohm T, Yin M and Luber S 2023 Neutral atom quantum computing hardware: performance and end-user perspective *EPJ Quantum Technol.* **10** 32

[590] Dhar P 2020 The carbon impact of artificial intelligence *Nat. Mach. Intell.* **2** 423–5

[591] Debus C, Piraud M, Streit A, Theis F and Götz M 2023 Reporting electricity consumption is essential for sustainable AI *Nat. Mach. Intell.* **5** 1176–8

[592] Strubell E, Ganesh A and McCallum A 2019 Energy and policy considerations for deep learning in NLP *Proc. 57th ACL* (ACL) pp 3645–50

[593] Royal Academy of Engineering 2024 Quantum infrastructure review: an independent review of the UK's quantum sector's infrastructure requirements for the next decade (available at: https://raeng.org.uk/policy-and-resources/research-and-innovation/how-can-we-improve-quantum-infrastructure-in-the-uk)

[594] Chen S 2023 The future is quantum: universities look to train engineers for an emerging industry *Nature* **623** 653–5

[595] Goorney S, Sarantinou M and Sherson J 2024 The quantum technology open master: widening access to the quantum industry *EPJ Quantum Technol.* **11** 7

[596] Heidt A 2024 Quantum computing aims for diversity, one qubit at a time *Nature* **632** 464–5

[597] Meyer J C, Passante G and Wilcox B 2024 Disparities in access to U.S. quantum information education *Phys. Rev. Phys. Educ. Res.* **20** 010131

[598] Rosenberg J L, Holincheck N and Colandene M 2024 Science, technology, engineering and mathematics undergraduates' knowledge and interest in quantum careers: barriers and opportunities to building a diverse quantum workforce *Phys. Rev. Phys. Educ. Res.* **20** 010138

[599] Ten Holter C, Inglesant P and Jirotka M 2023 Reading the road: challenges and opportunities on the path to responsible innovation in quantum computing *Technol. Anal. Strat. Manage.* **35** 844–56

[600] World Economic Forum, IBM and SandboxAQ 2024 Quantum economy blueprint *Insight Report* (World Economic Forum) (available at: www.weforum.org/publications/quantum-economy-blueprint/)

[601] Neumann N M P, van Heesch M P P, Phillipson F and Smallegange A A P 2021 Quantum computing for military applications *2021 ICMCIS* (IEEE) pp 1–8