15 Quantum Many-Body Simulations on Digital Quantum Computers

Benedikt Fauseweh
TU Dortmund University
Otto-Hahn-Str 4, 44227 Dortmund

Contents

E. Pavarini and E. Koch (eds.)
Correlated Materials: Methods and Applications
Modeling and Simulation Vol. 16, Forschungszentrum Jülich, 2026
ISBN 978-3-95806-909-1 http://www.cond-mat.de/events/correl26

1 Introduction

Solving the quantum many-body problem is notoriously difficult, due to the exponential scaling of the total Hilbert space dimension with the number of particles. We can understand this problem by starting from the arguably simplest quantum mechanical system: a two-level atom. We denote the orthonormal states $|0\rangle$ and $|1\rangle$ as a basis in the two-dimensional Hilbert space. A quantum system built from N coupled two-level systems defines a Hilbert space $\mathcal{H} = (\mathbb{C}^2)^{\otimes N}$ of dimension 2^N. A generic pure state

$$|\psi\rangle = \sum_{x\in\{0,1\}^N} c_x |x\rangle , \qquad \sum_x |c_x|^2 = 1, \tag{1}$$

is determined by 2^N complex amplitudes c_x and hence the number of parameters needed merely to write down the state grows exponentially with N. In contrast the resources of the laboratory, e.g., the number of atoms, grow typically linearly with N. To follow the time evolution of such a state on a classical computer one must, in the absence of other special structures like symmetries, store and update this exponentially long list. Recently such a time evolution was implemented numerically on classical computers for $N = 50$ [1], using the JUPITER supercomputer at Forschungszentrum Jülich. It is probably safe to say, that a vector with $N = 300$ and 128 bits of precision will not be used in a classical simulation in our lifetime, as the number of amplitudes is larger than the number of atoms in the visible universe. This single observation, that the description of correlated quantum matter is generically intractable by classical means, is the reason we want to think more about what it means to compute a quantum system.
A possible response is to turn the difficulty into a tool. If a classical computer struggles because it must track an exponential amount of quantum information, then a controllable quantum system, which carries that information natively, should not. The task of using one quantum system to imitate the dynamics of another is *quantum simulation*, and the realization of that task on a programmable, gate-based quantum computer is *digital quantum simulation*.

1.1 A computational interpretation of quantum mechanics

The idea that quantum mechanics can be used computationally on its own emerged around 1980. Manin remarked that the dimension of a many-particle state space multiplies under composition, whereas classical configuration spaces only add, and suggested that "quantum automata" might therefore compute in ways no classical device could [2]. Benioff constructed an explicit quantum-mechanical Hamiltonian whose unitary evolution implements an arbitrary Turing machine, showing that classical computation is contained within quantum dynamics [3].
The proposal of a physical quantum computer is due to Feynman [4] in 1982. Feynman asked whether a classical computer can simulate quantum physics, and argued that it cannot do so efficiently. A faithful classical model of the correlations measured on an entangled pair of photons, would have to reproduce the joint statistics of incompatible measurements, and any local assignment of probabilities to the underlying variables is forced to take negative values. Since a classical probabilistic machine samples from non-negative distributions, it cannot reproduce

these correlations without exponential overhead. Feynman then proposed building the simulator out of quantum elements that obey the same laws as the target, conjecturing that such a system could serve as a quantum computer.

In 1985, Deutsch placed this idea on a sharper footing [5] and defined the universal quantum computer as a model of computation in its own right. He introduced an artificial task on which it outperforms any classical machine, introducing the concept of "quantum parallelism". The computational and the physical notions of simulation where thereby unified.

In 1994 Shor developed a quantum algorithm that factors integers and computes discrete logarithms exponentially faster than the best known classical method [6, 7]. Shor's result detached quantum computation from its origin in physics simulation and established it as a general computational paradigm.

The idea begun by Feynman was further developed by Lloyd in 1996, who proved the conjecture that a quantum computer can efficiently simulate the time evolution of any quantum system governed by local interactions [8].

1.2 Quantum mechanics as a generalized probability theory

It is worth asking what separates a quantum device from a classical randomized one. The cleanest answer, due in this form to Aaronson [9], is that quantum mechanics is the theory one obtains by replacing classical probability with the ℓ^1 norm by the ℓ^2 norm.

A classical state over N outcomes is a probability vector $p = (p_1, \dots, p_N)$ with $p_i \geq 0$ and

$$\|p\|_1 \;=\; \sum_i p_i \;=\; 1. \tag{2}$$

The maps that send probability vectors to probability vectors are exactly the (column-)stochastic matrices, that is, the linear maps preserving the ℓ^1 norm for non-negative probabilities. A quantum state is instead a vector $|\psi\rangle \in \mathbb{C}^N$ normalized in the ℓ^2 sense,

$$\|\psi\|_2 \;=\; \Big(\sum_i |\psi_i|^2\Big)^{1/2} \;=\; 1, \tag{3}$$

and the linear maps preserving this norm are exactly the unitaries $U \in \mathrm{U}(N)$. The Born rule $p_i = |\psi_i|^2$ is the probability distribution determined from the amplitudes at the moment of measurement, and Gleason's theorem [10] singles it out as the only consistent choice in dimension three and above.

That the exponents 1 and 2 are the only ones that yield a non-trivial theory is not an accident. Demanding that a linear map preserve the ℓ^p norm imposes a system of polynomial constraints on its entries, and one can show that the resulting transformations are only permutations and sign flips unless $p = 1$, which gives classical probability, or $p = 2$, which gives quantum mechanics [11]. The use of complex rather than real amplitudes is fixed by a comparable demand: requiring that every allowed evolution possess a square root, so that it may be applied over half a time step, forces the field to be $\mathbb{C}$, although alternative formulations exist [12].

The single feature that distinguishes the ℓ^2 theory from the ℓ^1 theory is that amplitudes may be negative or complex, and so may cancel. Consider the Hadamard gate in the standard basis

$$H \;=\; \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{4}$$

Acting on $|0\rangle$ it produces the superposition $\frac{1}{\sqrt{2}}\big(\,|0\rangle + |1\rangle\,\big)$, which on measurement looks exactly like a fair coin. A classical fair coin tossed twice leaves one no more certain of the outcome than after one toss. The quantum coin behaves differently:

$$H^2\,|0\rangle \;=\; |0\rangle\,. \tag{5}$$

The amplitude to reach $|1\rangle$ accumulates over two paths, through the intermediate $|0\rangle$ and through the intermediate $|1\rangle$, and these carry opposite signs,

$$\frac{1}{\sqrt{2}}\cdot\frac{1}{\sqrt{2}} \;+\; \frac{1}{\sqrt{2}}\cdot\Big(-\frac{1}{\sqrt{2}}\Big) \;=\; 0, \tag{6}$$

so the outcome $|1\rangle$ disappears through destructive interference. No dynamics on non-negative probabilities can produce this cancellation. This is how a quantum computer differs uniquely from a classical computer.

1.3 The formal task of quantum simulation

In this section we want to formalize the task of quantum simulation mathematically. Throughout we set $\hbar = 1$. We start from the time-dependent Schrödinger equation

$$\mathrm{i}\,\frac{\mathrm{d}}{\mathrm{d}t}\,|\psi(t)\rangle \;=\; H\,|\psi(t)\rangle \tag{7}$$

which governs the evolution of a closed system under a Hamiltonian H. For time-independent H its solution is generated by the unitary propagator

$$|\psi(t)\rangle \;=\; U(t)\,|\psi(0)\rangle\,, \qquad U(t) \;=\; \mathrm{e}^{-\mathrm{i}Ht}. \tag{8}$$

This defines the first and central problem, *Hamiltonian* or *dynamical simulation*: given a description of a local Hamiltonian H on n qubits, a time t, an accuracy ε, and a means of preparing an input $|\psi_0\rangle$, produce a state $|\phi\rangle$ with

$$\big\|\,|\phi\rangle - U(t)\,|\psi_0\rangle\,\big\| \;\le\; \varepsilon. \tag{9}$$

In this context, *local* does not mean geometrically local, but it refers to the terms in the Hamiltonian that act nontrivially only on a small number of qubits. For example the term k-local, refers to a Hamiltonian in which every single term acts at most onto k qubits.
In contrast the time-independent Schrödinger equation poses a different problem,

$$H\,|n\rangle \;=\; E_n\,|n\rangle\,, \tag{10}$$

the determination of the spectrum and eigenstates of H, most often its ground state energy $E_0 \leq E_n$ and ground state $|0\rangle$ or equivalently its ground state manifold.
The two tasks are not of equal difficulty. Dynamical simulation (9) is tractable on a quantum computer, whereas estimating the ground-state energy of a generic local Hamiltonian is Quantum Merlin Arthur (QMA)-complete [13, 14]. A problem that is QMA-complete is believed to be hard even with a perfect quantum computer available. Thus simulating how a system evolves is one thing, and finding the ground state is another.
The tractability of the dynamical problem is the content of Lloyd's theorem [8]. Suppose the Hamiltonian is a sum of local terms,

$$H \;=\; \sum_{l=1}^{L} H_l, \tag{11}$$

each H_l is k-local. The exponential $\mathrm{e}^{-\mathrm{i}Ht}$ does not factor into a product over the H_l because the terms do not commute. It can, however, be approximated by such a product over a short time. Writing r for the number of time slices, the first-order Lie-Trotter formula [15] reads

$$\mathrm{e}^{-\mathrm{i}Ht} \;=\; \Big(\prod_{l=1}^{L} \mathrm{e}^{-\mathrm{i}H_l t/r}\Big)^{r} \;+\; \mathcal{O}\Big(\frac{t^2}{r}\sum_{l<j}\big\|[H_l,H_j]\big\|\Big), \tag{12}$$

where the error term is the leading contribution of the Baker-Campbell-Hausdorff expansion and is controlled by the commutators of H_l [16]. To reach accuracy ε over time t it therefore suffices to take

$$r \;=\; \mathcal{O}\big(t^2/\varepsilon\big) \tag{13}$$

slices, each built from L elementary exponentials, every one of which acts on at most k constituents. The total count of exponentials grows polynomially in n, t and $1/\varepsilon$. For a quantum device the application of $\mathrm{e}^{-\mathrm{i}H_l t/r}$ is assumed to be the elementary operation, while for classical computers this step scales with the dimension of the Hilbert space $\dim\mathcal{H} = 2^N$. This is the precise sense in which a quantum computer simulates local quantum dynamics *efficiently*, while the direct classical evolution of (1) does not [8]. Higher-order product formulas [17] and methods based on sparse-Hamiltonian techniques and quantum signal processing [18, 19] improve the dependence on t and on ε.

1.4 Physics as a quantum program

Equation (12) allows us to interpret quantum dynamics from the perspective of computer science. The Hamiltonian is no longer only an object of study but a *program*: the physics of interest is specified by the list of local terms H_l, and the job of the simulator is to run the propagator $U(t)$ they generate. Two strategies exist for doing so, and divide the field of quantum simulation [20, 21].
In *analog* quantum simulation one engineers a controllable system whose native Hamiltonian H_{sim} reproduces, under a known mapping, the target H_{tgt}, and one simply lets it evolve. The

continuous evolution under H_{sim} is itself the computation and no decomposition into elementary steps occurs. Ultracold atoms in optical lattices realizing Hubbard models [22, 23] and trapped-ion spin chains [24] are examples of such analog simulations. Analog quantum simulations reach large particle numbers and incur no Trotter error in the ideal limit, but each is tied to a particular family of Hamiltonians and offers no route to error correction.

In *digital* quantum simulation, one instead compiles the propagator into a sequence of gates drawn from a fixed universal set, exactly as in Eq. (12). The price is a discretization error and a circuit depth that grows with t and $1/\varepsilon$. The reward is universality, since any local Hamiltonian can be simulated by the same hardware, and compatibility with quantum error correction, which is what makes accurate long-time simulation conceivable in principle.

2 Quantum bits and quantum circuits

In the first section we introduced the concept of quantum computers and the formal task of quantum simulation. In this section we want to make this approach more concrete. We define the quantum bit, characterize the operations one may perform on it, introduce the concept of entanglement and how to induce it on a quantum register, and prove that a small set of operations suffices to realize any arbitrary unitary transformation.

2.1 Quantum bit and unitary gates

A quantum bit, or qubit, is a normalized vector in a two-dimensional complex Hilbert space equipped with a orthonormal basis $\{|0\rangle, |1\rangle\}$,

$$|\psi\rangle \;=\; \alpha\,|0\rangle + \beta\,|1\rangle\,, \qquad |\alpha|^2 + |\beta|^2 = 1, \qquad \alpha,\beta \in \mathbb{C}. \tag{14}$$

Of the four real parameters in α and β, normalization removes one and the irrelevance of a global phase removes another. Thus, without loss of generality, we can write a single qubit state as,

$$|\psi\rangle \;=\; \cos\frac{\vartheta}{2}\,|0\rangle + \mathrm{e}^{\mathrm{i}\varphi}\sin\frac{\vartheta}{2}\,|1\rangle\,, \qquad \vartheta \in [0,\pi],\ \varphi \in [0,2\pi). \tag{15}$$

One interprets the angles (ϑ, φ) as the polar and azimuthal angles of a point on the unit sphere, the Bloch sphere. The basis states $|0\rangle$ and $|1\rangle$ sit at the poles, and orthogonal states are antipodal rather than separated by a right angle. In contrast to a classical bit, which can only occupy the poles, a qubit state can exist over the entire surface.

Between preparation and measurement a qubit evolves under the Schrödinger equation, whose solution is the propagator $U(t) = \mathrm{e}^{-\mathrm{i}Ht}$ of Section 1.3. Thus, whenever we control a qubit by applying a pulse, turning on an interaction, or letting it evolve for a prescribed time, the corresponding operation on the state is a unitary transformation. A quantum gate is such a controlled unitary time evolution. Instead of describing the detailed Hamiltonian, pulse shape, and duration that realize an operation in a particular device, we describe the net effect on the

qubits by a unitary matrix U. For example, a gate may represent the evolution generated by some control Hamiltonian H_{gate} for a fixed time τ,

$$U_{\mathrm{gate}} = \mathrm{e}^{-\mathrm{i}H_{\mathrm{gate}}\tau}. \tag{16}$$

In this sense, gates are not fundamentally different from time-evolution operators. They are time-evolution operators viewed at selected times and used as elementary building blocks of a computation. The defining property of a unitary gate is that its adjoint is its inverse,

$$U^{\dagger}U = UU^{\dagger} = I. \tag{17}$$

Hence every quantum gate is reversible. This reversibility is in contrast to many ordinary classical logic gates such as AND and OR. These gates map two input bits to one output bit and therefore discard information: from the output alone one cannot reconstruct the input. A unitary gate, by contrast, maps input states to output states with the same dimension. No information is lost during the coherent part of the quantum circuit. Another consequence of unitary gates is that they preserve the ℓ^2 norm of a state vector,

$$\big\| U\,|\psi\rangle \big\|^2 = \langle\psi|\,U^{\dagger}U\,|\psi\rangle = \langle\psi|\psi\rangle \tag{18}$$

and therefore preserve the total probability discussed in Section 1.2. The action of a gate may redistribute probability amplitudes among basis states, and it may change their relative phases, but it cannot change the total probability.

The operators of one qubit can be expressed through the Pauli matrices

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y = \begin{pmatrix} 0 & -\mathrm{i} \\ \mathrm{i} & 0 \end{pmatrix}, \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{19}$$

Each is Hermitian, unitary and traceless, squares to the identity, and has eigenvalues ± 1. They satisfy the algebra

$$\sigma_j\sigma_k \;=\; \delta_{jk}\, I + \mathrm{i}\,\epsilon_{jkl}\,\sigma_l, \tag{20}$$

and together with the identity I they span the complex 2×2 matrices, so any single-qubit Hamiltonian takes the form $H = h_0 I + \mathbf{h}\cdot\boldsymbol{\sigma}$ with real h_0 and $\mathbf{h}$. Exponentiating a Pauli operator generates a rotation. For a unit vector $\mathbf{n}$, the identity $(\mathbf{n}\cdot\boldsymbol{\sigma})^2 = I$ gives

$$R_{\mathbf{n}}(\vartheta) \;=\; \mathrm{e}^{-\mathrm{i}\frac{\vartheta}{2}\,\mathbf{n}\cdot\boldsymbol{\sigma}} \;=\; \cos\frac{\vartheta}{2}\, I - \mathrm{i}\sin\frac{\vartheta}{2}\,\mathbf{n}\cdot\boldsymbol{\sigma}, \tag{21}$$

which rotates the Bloch vector through angle ϑ about the axis $\mathbf{n}$. These rotations exhaust the single-qubit gates up to a phase: every $U \in \mathrm{U}(2)$ admits the decomposition

$$U \;=\; \mathrm{e}^{\mathrm{i}\alpha}\, R_z(\beta)\, R_y(\gamma)\, R_z(\delta) \tag{22}$$

for real angles $\alpha, \beta, \gamma, \delta$, see also [25, 26]. Three rotations about two fixed axes therefore reach any single-qubit operation.

Several of these unitaries recur often enough to carry their own names. The three Pauli operators act as gates in their own right, written $\sigma_x = X$, $\sigma_y = Y$ and $\sigma_z = Z$ in this context. X is the quantum bit flip $|0\rangle \leftrightarrow |1\rangle$, Z leaves $|0\rangle$ fixed while sending $|1\rangle \mapsto -|1\rangle$, i.e., Z is a phase flip, and Y applies both a bit flip and a phase flip. The Hadamard gate, already met in Section 1.2, maps the basis onto the states $|\pm\rangle = \big(|0\rangle \pm |1\rangle\big)/\sqrt{2}$ and obeys $HXH = Z$ and $HZH = X$. Two more diagonal gates are important for error correction, the phase gate S and the T gate,

$$S \;=\; \begin{pmatrix} 1 & 0 \\ 0 & \mathrm{i} \end{pmatrix}, \qquad T \;=\; \begin{pmatrix} 1 & 0 \\ 0 & \mathrm{e}^{\mathrm{i}\pi/4} \end{pmatrix}, \tag{23}$$

which up to a global phase are the rotations $R_z(\pi/2)$ and $R_z(\pi/4)$ and satisfy $S = T^2$ and $Z = S^2$. The T gate is also called the $\pi/8$ gate, after the half-angles $\pm\pi/8$ that appear on the diagonal of $R_z(\pi/4)$.

Having introduced the Pauli matrices allows us to also consider mixed quantum states. A density operator ρ of a single qubit is a Hermitian, positive operator on $\mathbb{C}^2$ with unit trace. The identity I and the three Pauli matrices $\boldsymbol{\sigma} = (\sigma_x, \sigma_y, \sigma_z)$ form a basis of the Hermitian $2{\times}2$ matrices, so ρ can always be expanded in them, leading to $\rho = \frac{1}{2}(I + \mathbf{r}\cdot\boldsymbol{\sigma})$. The vector $|\mathbf{r}| \le 1$ occupies the interior of the Bloch sphere.

2.2 Entanglement and controlled-NOT

Single-qubit gates alone, however freely composed, cannot generate every state. Acting on a register they appear as a tensor product, and such an operator sends products to products,

$$(U_1 \otimes U_2)\big(|\psi\rangle \otimes |\phi\rangle\big) \;=\; \big(U_1|\psi\rangle\big) \otimes \big(U_2|\phi\rangle\big). \tag{24}$$

A state that cannot be written as a single tensor product is called entangled. The Bell state $\big(|00\rangle + |11\rangle\big)/\sqrt{2}$ is the canonical example, and no product of single-qubit gates produces it from an unentangled input. This is a consequence of the counting argument of Section 1, i.e., a product state of n qubits is fixed by $\mathcal{O}(n)$ parameters, a general state by 2^n, and only a gate that couples qubits can carry the register between the two regimes.

For two subsystems (A) and (B), this distinction can be made precise by the Schmidt decomposition. Any pure state can be written as

$$|\Psi\rangle = \sum_{\mu=1}^{\chi} \lambda_\mu \, |\mu\rangle_A \otimes |\mu\rangle_B \,, \tag{25}$$

with non-negative Schmidt coefficients λ_μ satisfying $\sum_\mu \lambda_\mu^2 = 1$. A product state has Schmidt rank $\chi = 1$. Entanglement is present when more than one Schmidt coefficient is nonzero.

The same information is contained in the reduced density matrix of either subsystem. Tracing out (B) gives

$$\rho_A = \mathrm{tr}_B \, |\Psi\rangle\langle\Psi| = \sum_{\mu=1}^{\chi} \lambda_\mu^2 \, |\mu\rangle_A \langle\mu|_A \,. \tag{26}$$

For a product state, ρ_A is again a pure-state projector. For an entangled state, ρ_A is mixed, even though the full state is pure. The corresponding entanglement entropy is

$$S_A = -\mathrm{tr}_A\, \rho_A \log_2 \rho_A = -\sum_{\mu=1}^{\chi} \lambda_\mu^2 \log_2 \lambda_\mu^2. \tag{27}$$

It vanishes for product states and is positive for entangled states. Thus the entanglement entropy is a measure for entanglement between two subsystems A and B. For the Bell state

$$\left|\Phi^+\right\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), \tag{28}$$

the two Schmidt coefficients are both $1/\sqrt{2}$, so that

$$\rho_A = \frac{I}{2}, \qquad S_A = 1. \tag{29}$$

Thus the Bell state contains one bit of entanglement between the two qubits. This is also called an *ebit*.

A gate that can create entanglement out of a product state is the controlled-NOT or, for short, CNOT gate [25]. It is defined by its action on the two qubits,

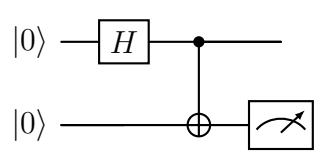
$$\mathrm{CNOT} \;=\; |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes \sigma_x, \tag{30}$$

which flips the target qubit when the control is $|1\rangle$, that is $|c, t\rangle \mapsto |c,\, t \oplus c\rangle$, where $\oplus$ is binary addition. Acting on a control that is prepared in a superposition entangles the pair of control and target qubit,

$$\mathrm{CNOT}\ |+, 0\rangle \;=\; \frac{1}{\sqrt{2}}\big(\,|0, 0\rangle + |1, 1\rangle\,\big), \qquad |+\rangle = \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle). \tag{31}$$

2.3 Quantum circuit diagrams

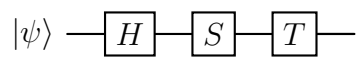
It is convenient to display a computation as a diagram. Each qubit is drawn as a horizontal wire, time runs from left to right, and a gate is a labelled box placed on the wire or wires it acts on. The controlled-NOT has its own symbol, a filled dot on the control wire joined by a vertical line to an $\oplus$ sign on the target, and a measurement in the computational basis is drawn as a meter in a box. The entangling operation of the previous subsection, which prepares a Bell state from the input $|00\rangle$ and then reads out the second qubit, is



(32)

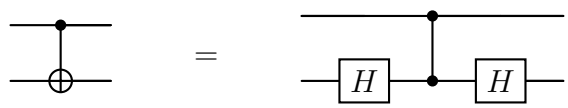
The diagram is read from left to right: the Hadamard is applied first on the upper qubit, and the controlled-NOT follows. The unitary represented by a circuit is the product of its gate operators

in the *reverse* order, because the gate drawn first acts first and therefore stands rightmost in the product. A single qubit with H, then S, then T applied,



(33)

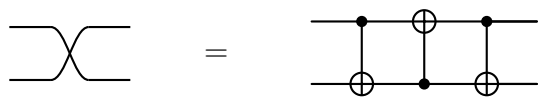
implements the operator $U = T\,S\,H$, in the same convention by which $TSH\,|\psi\rangle$ has H act first. Diagrams also state identities between circuits. The relation $\mathrm{CNOT} = (I\otimes H)\,\mathrm{CZ}\,(I\otimes H)$, with CZ the controlled-Z drawn as two linked dots, becomes the statement that the circuits



(34)

represent the same unitary. We use such diagrams throughout to specify simulation algorithms compactly. A single Trotter step, for instance, will appear in Section 3 as a fixed pattern of gates that is then repeated along the time axis.

Connectivity constraints make circuit identities important. On many quantum processors, such as superconducting qubits, two-qubit gates cannot be applied between arbitrary pairs of qubits, but only along the edges of a typically sparse hardware graph. To bring two qubits next to one another, one can exchange their quantum states using a SWAP gate. In terms of CNOT gates, this operation is implemented as



(35)

where the three CNOTs exchange the computational basis states of the two qubits. SWAP gates are therefore often inserted by a compiler to route qubits through the hardware graph until the desired two-qubit interaction becomes local.

2.4 Universality of single-qubit gates and the CNOT

A small set of elementary gates is useful only if it is sufficiently expressive. The central structural result behind the circuit model is that arbitrary unitaries U can be built from local gates. More precisely, arbitrary single-qubit gates together with one entangling two-qubit gate, conventionally chosen as the CNOT, form a universal gate set [26, 27, 25]. This means that every unitary operation on an n-qubit register can be decomposed into a finite sequence of one-qubit gates and CNOT gates.

This statement should be understood in close analogy with universality in classical computation. In classical circuits, gates such as NAND are universal because arbitrary Boolean functions can be constructed from them. In quantum circuits, the objects to be constructed are not Boolean functions but unitary matrices. Universality therefore says that a small set of experimentally available operations is enough, in principle, to realize any arbitrary unitary matrix.

A general unitary on n qubits is a $2^n\times 2^n$ matrix, and therefore acts on the full Hilbert space at once. The core idea is to decompose this large unitary into *two-level unitaries*. Let $|x\rangle$

denote the computational basis. A two-level unitary G_{pq} acts non-trivially only on the subspace spanned by two basis states $|p\rangle$ and $|q\rangle$,

$$G_{pq}(\vartheta,\varphi,\chi) = \sum_{x\neq p,q} |x\rangle\langle x| + a\,|p\rangle\langle p| + b\,|p\rangle\langle q| + c\,|q\rangle\langle p| + d\,|q\rangle\langle q|\,, \tag{36}$$

with a complex Givens rotation matrix

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} \mathrm{e}^{\mathrm{i}\varphi}\cos\vartheta & \mathrm{e}^{\mathrm{i}\chi}\sin\vartheta \\ -\mathrm{e}^{-\mathrm{i}\chi}\sin\vartheta & \mathrm{e}^{-\mathrm{i}\varphi}\cos\vartheta \end{pmatrix}. \tag{37}$$

It therefore mixes only two amplitudes of a state $|\psi\rangle = \sum_x c_x\,|x\rangle$,

$$\begin{pmatrix} c_p \\ c_q \end{pmatrix} \mapsto \begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} c_p \\ c_q \end{pmatrix}, \qquad c_x \mapsto c_x \quad (x\neq p,q). \tag{38}$$

By choosing the angles one can represent a selected matrix element of U. These elementary transformations can be decomposed into one-qubit gates and CNOTs. Note that this decomposition is universal, but not necessarily efficient, i.e., the decomposition of a $2^n\times 2^n$ matrix would require an exponential number of such two-level unitaries.

This *universality*, as stated above, requires a continuous set of single qubit gates. Physically, however, this continuum of operations is not what one wants to use in quantum computer. The microscopic control fields are analog and therefore cannot be set with infinite precision. Moreover, small calibration errors would then translate directly into small, continuously varying logical errors. Thus it makes sense to further reduce this set to a discrete gate set. It turns out, that only having H, S, T is sufficient to efficiently realize any arbitrary single qubit unitary to arbitrary precision [28, 29]. Together with the CNOT, this finite set is thus universal. This limitation to a discrete set of gates is required for the application of quantum error correction.

3 Simulating strongly correlated systems on quantum computers

We have so far introduced quantum circuits as a model of computation and quantum simulation as an abstract task. We now apply this knowledge as a computational tool to tackle strongly correlated quantum systems.

The examples in this section are chosen with this purpose in mind. We start with real-time evolution of spin-1/2 models, because local Pauli operators are already native objects of the qubit register. Fermionic lattice models require an additional encoding step, and we will use them to discuss how fermionic creation and annihilation operators can be mapped to qubit operators. Finally we tackle the ground-state problem. This brings us to variational algorithms an their limitations.

Throughout this section we use $S_i^\alpha = \sigma_i^\alpha/2$ for spin-$1/2$ operators. The elementary gate set is assumed to contain arbitrary single-qubit rotations and the controlled-NOT. This is a convenient compilation target rather than a hardware statement. On a specific device the native entangling gate may be different, but it can be converted to the same language by local basis changes.

3.1 Non-equilibrium dynamics of spin systems

We consider a simple spin Hamiltonian in a time-dependent magnetic field [30]. We aim to describe the time evolution of an initial state through quantum circuits. The time dependent Hamiltonian is given by,

$$H(t) = \sum_{\langle i,j\rangle} J_\perp \big(S_i^x S_j^x + S_i^y S_j^y\big) + J_z S_i^z S_j^z + \sum_i \boldsymbol{H}_i(t)\cdot\boldsymbol{S}_i. \tag{39}$$

The notation $\langle i,j\rangle$ denotes couplings on an arbitrary but sparse interaction graph, e.g., a two-dimensional square lattice. The first two terms describe an XXZ exchange interaction. The last term is a local magnetic field, which can depend on time and on the site. In the Pauli language this is

$$H(t) = \sum_{\langle i,j\rangle} \frac{J_\perp}{4}\big(X_iX_j + Y_iY_j\big) + \frac{J_z}{4} Z_iZ_j + \frac{1}{2}\sum_i \big(h_i^x(t)X_i + h_i^y(t)Y_i + h_i^z(t)Z_i\big). \tag{40}$$

Note that each term in the sum is a Pauli string, i.e., a product of Pauli operators. The time evolution task is to approximate

$$U(t,0) = \mathcal{T}\exp\left[-\mathrm{i}\int_0^t H(s)\,\mathrm{d}s\right] \tag{41}$$

for an initial state that may be a product state, a prepared ground state or some other state that was prepared beforehand. We divide the interval into r pieces, $\delta t = t/r$, and approximate the Hamiltonian on the mth interval by its midpoint value $H_m = H(t_m)$, $t_m = (m{+}1/2)\delta t$. A first-order product formula gives

$$U(t,0) = \prod_{m=r-1}^{0}\left[\prod_i U_i^{\mathrm{field}}(t_m,\delta t)\prod_{\langle i,j\rangle} U_{ij}^{XX}(\delta t)U_{ij}^{YY}(\delta t)U_{ij}^{ZZ}(\delta t)\right] + \mathcal{O}(\delta t). \tag{42}$$

The ordering inside one step is a convention. A symmetric second-order step replaces $\prod_l e^{-\mathrm{i}H_l\delta t}$ by half steps around the last factor and reduces the local error from $\mathcal{O}(\delta t^2)$ to $\mathcal{O}(\delta t^3)$, at the cost of a longer circuit.

The local field terms are single-qubit gates. For example

$$\exp\left[-\mathrm{i}\frac{h_i^z(t_m)}{2}Z_i\delta t\right] = R_z\big(h_i^z(t_m)\delta t\big), \tag{43}$$

and similarly for x and y. For a general field one may either implement a single rotation about the axis $\boldsymbol{h}_i/h$, with $h = |\boldsymbol{h}_i|$, as,

$$U_i^{\mathrm{field}}(t_m,\delta t) = \exp\left[-\frac{\mathrm{i}\delta t}{2}\boldsymbol{h}_i(t_m)\cdot\boldsymbol{\sigma}_i\right] = R_{\hat{\boldsymbol{h}}_i}\big(|\boldsymbol{h}_i(t_m)|\delta t\big). \tag{44}$$

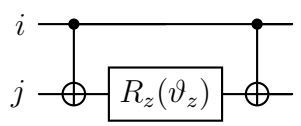
As long as a universal gate set is available this rotation can always be implemented. Let us take a concrete example: A quantum computer only allows R_z and R_x rotations. Then a generic

single-qubit gate may be compiled as $R_z(\alpha)R_x(\beta)R_z(\gamma)$. The rotation around an arbitrary axis can be implemented as $R_{\hat{\boldsymbol{h}}}(\vartheta)=R_z(A+B)R_x(\beta)R_z(A-B)$, where $A=\operatorname{atan2}\big(h_z\sin\frac{\vartheta}{2}, h\cos\frac{\vartheta}{2}\big)$, $B=\operatorname{atan2}(h_y,h_x)$ and $\beta=2\arccos\sqrt{\cos^2\frac{\vartheta}{2}+\frac{h_z^2}{h^2}\sin^2\frac{\vartheta}{2}}$, $\vartheta=h\delta t$.
We decompose the two-qubit terms into single qubit gates and CNOT. A useful primitive is the exponential of a Pauli string. For the ZZ interaction,

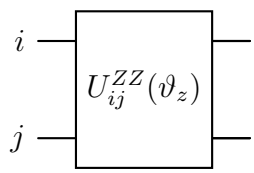
$$U_{ij}^{ZZ}(\delta t)=\exp\left[-\mathrm{i}\frac{J_z\delta t}{4}Z_iZ_j\right]=\exp\left[-\mathrm{i}\frac{\vartheta_z}{2}Z_iZ_j\right],\qquad \vartheta_z=\frac{J_z\delta t}{2}\,. \tag{45}$$

The circuit is given by

$$\exp\left[-\mathrm{i}\frac{\vartheta_z}{2}Z_iZ_j\right]=$$



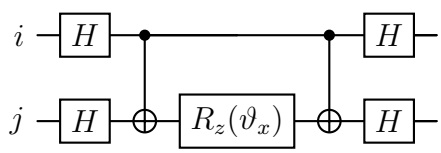
$$=$$



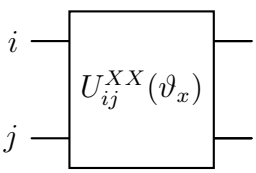
(46)

To see this, note that the first CNOT maps Z_j to Z_iZ_j. Thus the middle rotation becomes an entangling phase after the basis change is undone.
The XX and YY interactions are the same construction in a different basis. Since $HXH=Z$, the XX exponential is obtained by rotating both qubits from the z basis to the x basis, applying the ZZ construction, and rotating back

$$\exp\left[-\mathrm{i}\frac{\vartheta_x}{2}X_iX_j\right]=$$



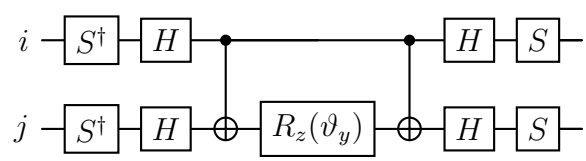
$$=$$



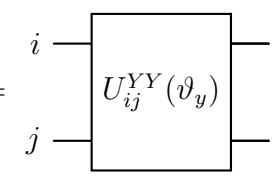
(47)

with $\vartheta_x=J_\perp\delta t/2$. For the YY term one may use $S^\dagger HYHS=Z$, or equivalently rotate the y axis to the z axis before applying the same entangling phase

$$\exp\left[-\mathrm{i}\frac{\vartheta_y}{2}Y_iY_j\right]=$$



$$=$$



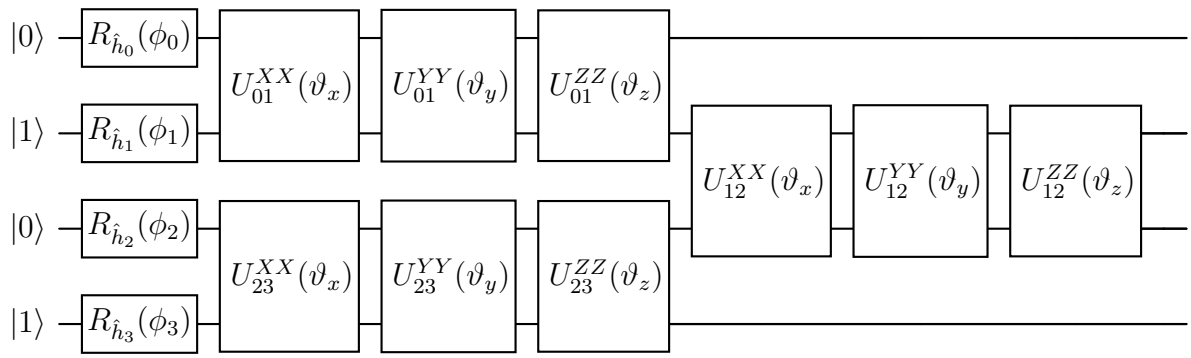
(48)

with $\vartheta_y=J_\perp\delta t/2$. For one pair (i,j) the three Pauli products X_iX_j, Y_iY_j and Z_iZ_j commute. Their product therefore gives the exact exponential of the pair Hamiltonian,

$$\exp\left[-\mathrm{i}\delta t\left(\frac{J_\perp}{4}X_iX_j+\frac{J_\perp}{4}Y_iY_j+\frac{J_z}{4}Z_iZ_j\right)\right]=U_{ij}^{XX}U_{ij}^{YY}U_{ij}^{ZZ}. \tag{49}$$

The Trotter error in Eq. (42) is not caused by these three factors on the same bond. It comes from non-commuting terms that share a *single* site, and from the time dependence of $\boldsymbol{H}_i(t)$.

A complete first-order Trotter step for a four site chain with open boundary conditions therefore has the structure shown below.



(50)

Here $\phi_i = |\boldsymbol{h}_i(t_m)|\,\delta t$. The first entangling layer acts on bonds $(0,1)$ and $(2,3)$, and the second layer acts on bond $(1,2)$. Longer chains repeat the same odd-even pattern. We can estimate the required number of gates from this construction. Note that typically the entangling gates, e.g., the CNOTs, are the most noisy gates on current quantum computing platforms. It thus makes sense to count the entangling gates to estimate the reliability of a quantum circuit. A single Pauli two-qubit exponential costs two CNOTs. An XXZ bond therefore costs six CNOTs if implemented by the three separate circuits. For a chain with N sites and open boundary conditions, one first-order step costs $6(N-1)$ CNOTs for the exchange layers and N one-qubit field rotations. For r steps the number of gates thus scales as $\mathcal{O}(rN)$ for a one-dimensional local Hamiltonian. As a result the circuit depth grows with the spacetime volume of the local problem, not with the Hilbert-space dimension.

In Fig. 1 we see an example of this time evolution for a spin plaquette driven by a short magnetic field pulse, see inset. The experiment was realized on an IBM superconducting quantum processor [30]. Shown is the staggered magnetization of the system. Since only antiferromagnetic ZZ interactions are present in this model, the system starts at $t = 0$ from a ground state configuration. Here six time steps are sufficient to realize an accurate time evolution. The deviation of the experimental results from the Trotterized time evolution is due to noisy gates and noisy readout.

3.2 Simulating fermions

Since spin systems are already qubit systems no additional mapping is required. For fermions the situation is different and we require a mapping to the Pauli algebra. Fermionic creation and annihilation operators obey

$$\left\{c_p, c_q^\dagger\right\} = \delta_{pq}, \qquad \left\{c_p, c_q\right\} = \left\{c_p^\dagger, c_q^\dagger\right\} = 0. \tag{51}$$

In contrast Pauli operators on different qubits commute. A fermion-to-qubit mapping must encode these signs. Historically this problem was solved through the Jordan-Wigner transfor-

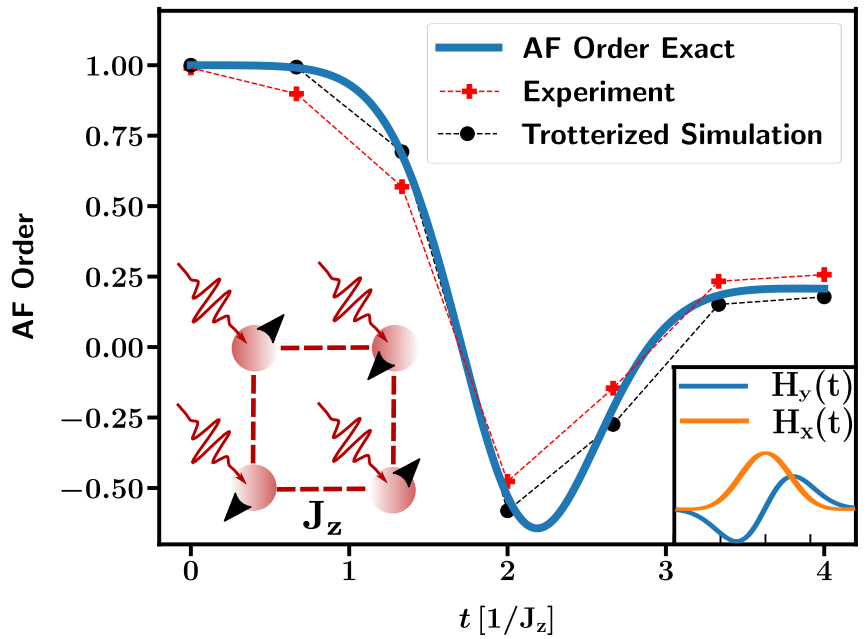


Fig. 1: *Nonequilibrium quantum simulation of a spin plaquette subject to a magnetic field pulse. Only the non-vanishing spin-spin coupling is shown. The circular polarized pulse is parameterized as* $H_x(t) = h_0 \exp\left(-(t-t_0)/(2\tau^2)\right) \cos\left(\omega(t-t_0)\right)$ *and* $H_y(t) = h_0 \exp\left(-(t-t_0)/(2\tau^2)\right) \sin\left(\omega(t-t_0)\right)$ *with parameters* h_0=2, ω=1, τ=0.7, t_0=2, *see inset. Displayed is the staggered magnetization* $S_{\mathrm{AF}} = \sum_{i=1}^{4} (-1)^i S_z^i$. *Comparison between the experimental results, the ideal Trotterized time evolution and the exact time evolution. Adopted from [30].*

mation, which is also the most pedagogical introduction to the topic. We define

$$c_p^\dagger = \left(\prod_{q<p} Z_q\right) \sigma_p^-, \qquad c_p = \left(\prod_{q<p} Z_q\right) \sigma_p^+, \tag{52}$$

where $\sigma^+ = (X+\mathrm{i}Y)/2$ and $\sigma^- = (X-\mathrm{i}Y)/2$ in the convention $|0\rangle$ empty, $|1\rangle$ occupied. The string of Z operators implements the parity of all modes to the left of p. This string produces the minus sign when two fermionic operators are interchanged.
For example, the number operator is local,

$$n_p = c_p^\dagger c_p = (1-Z_p)/2, \tag{53}$$

whereas a hopping term between $p < q$ becomes

$$c_p^\dagger c_q + c_q^\dagger c_p = \frac{1}{2}\left(X_p\, Z_{p+1} \cdots Z_{q-1}\, X_q + Y_p\, Z_{p+1} \cdots Z_{q-1}\, Y_q\right). \tag{54}$$

A density-density interaction is again local in the occupation basis,

$$n_p n_q = \frac{1}{4}\left(1 - Z_p - Z_q + Z_p Z_q\right). \tag{55}$$

Thus the cost of Jordan-Wigner depends on the ordering of the fermionic operators. From this it follows straightforwardly that one-dimensional fermionic systems with short range interactions

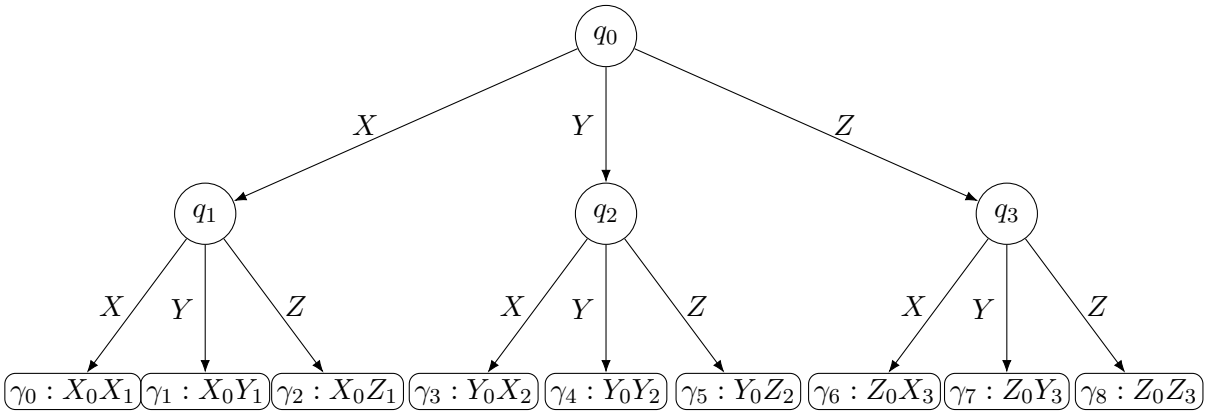


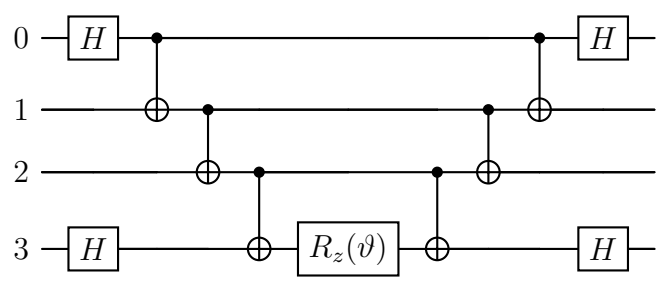
Fig. 2: *Schematic ternary-tree assignment of Majorana operators to Pauli strings [31]. The labels along a root-to-leaf path specify the Pauli operators in the string. In a full mapping one selects $2M$ of these leaves for M fermionic modes and pairs them into creation and annihilation operators as in Eq.* (58)*.*

can be implemented with short string operators, but two and three-dimensional lattices or general quantum-chemistry Hamiltonians will contain long parity strings. This affects the number of entangling gates in the circuit.

For a general Pauli string the circuits can be implemented in the following fashion. Suppose

$$P = X_p Z_{p+1} \cdots Z_{q-1} X_q, \qquad U_P(\vartheta) = \exp(-\mathrm{i}\vartheta P/2). \tag{56}$$

First rotate every non-Z factor to Z. Then compute the parity onto one qubit with a ladder of CNOTs, apply $R_z(\vartheta)$, and uncompute. For the string $X_0Z_1Z_2X_3$ this gives



(57)

For a length-ℓ string this costs $2(\ell-1)$ CNOTs and is hence undesirable.

To overcome this issue in higher dimensions we use tree-based decompositions to improve the mapping by distributing parity information over a tree rather than putting it into one long line. A particularly symmetric construction is the ternary-tree mapping of Ref. [31], see Fig. 2. Each edge of a ternary tree is labelled by one of the Pauli matrices X, Y, Z. A leaf ℓ is associated with the Pauli string obtained by multiplying the edge labels along the path from the root to that leaf. With a suitable assignment of leaves, these strings form a set of pairwise anticommuting Majorana operators.

The Majorana operators can then be easily translated to fermionic operators via

$$\gamma_{2p} = c_p + c_p^\dagger, \qquad \gamma_{2p+1} = -\mathrm{i}(c_p - c_p^\dagger), \qquad \{\gamma_a, \gamma_b\} = 2\delta_{ab}. \tag{58}$$

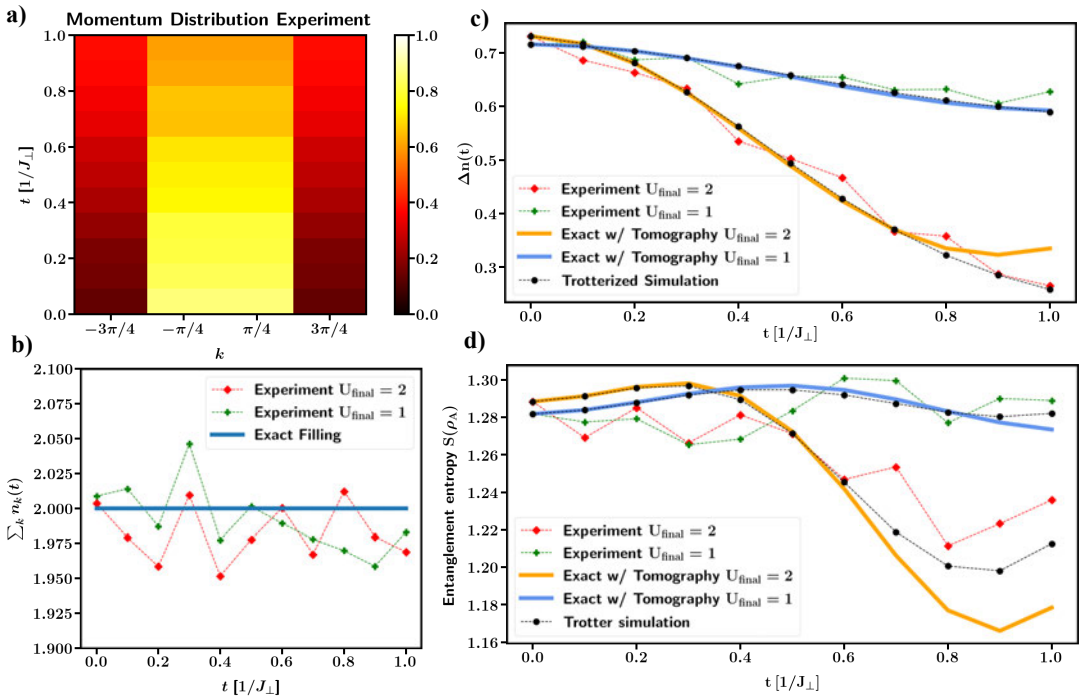


Fig. 3: *Quantum simulation of an interaction quench in a fermion system. a) Time and momentum dependence of the fermionic distribution function on a periodic four site chain after an interaction quench to $U_{\text{final}} = 2$ at t=0. b) Time evolution of the filling factor. Comparison between the exact filling, the weak quench $U_{\text{final}} = 1$ and the strong quench $U_{\text{final}} = 2$. c) Time evolution of the jump in the Fermi distribution. Comparison between the experimental results for weak and strong quenches and classical simulations. d) Entanglement entropy of a bipartition of the system. Adopted from [30].*

A valid mapping sends the $2M$ Majoranas of M fermionic modes to $2M$ qubit Pauli strings that square to identity and anticommute pairwise. In the Jordan-Wigner transformation the anticommutation of all Fermions is enforced by a linear parity string. In the tree construction it is enforced by the branching structure: two paths that first separate at a node carry different Pauli labels on that qubit, and hence the corresponding strings anticommute on only one site. A complete ternary tree of height h has 3^h leaves, so it can host order 3^h Majoranas with Pauli weight h. The Pauli weight, i.e. the number of Pauli operators, is therefore logarithmic in the number of fermionic modes and so is the necessary number of CNOTs to implement such an operator.

As an example of fermionic dynamics we investigate a four-site chain of interacting spinless fermions, equivalent to an XXZ chain,

$$H = \sum_i J_\perp \left(S_i^x S_{i+1}^x + S_i^y S_{i+1}^y\right) + U(t) S_i^z S_{i+1}^z. \tag{59}$$

Using the Jordan-Wigner transformation this is equal to the nearest-neighbor model

$$H = \frac{J_\perp}{2} \sum_i \left(c_i^\dagger c_{i+1} + c_{i+1}^\dagger c_i\right) + U(t) \sum_i \left(n_i - \frac{1}{2}\right)\left(n_{i+1} - \frac{1}{2}\right), \tag{60}$$

up to a constant. We want to investigate how the non-interacting ground state for $U = 0$ evolves if the interaction is suddenly switched on at $t = 0$, i.e., $U(t) = U_{\text{final}}\Theta(t)$. The subsequent unitary dynamics allows us to investigate how many-body systems relax after excitation. On a quantum computer the initial Fermi sea is not a computational basis state as it is already entangled in the site basis. We here skip over this encoding problem of the initial state and focus on the time evolution. We can apply the same $XX + YY$ circuits used in the spin section implement the kinetic term. The interaction term is diagonal and uses the ZZ circuit. In Fig. 3 we see the results from an IBM Quantum Computer for the momentum distribution $n_k(t)$ and the entanglement entropy for a 4-site system. Notably the quantum computation shows a significant amount of noise, which is due to the added encoding circuit not shown here.

3.3 Variational quantum eigensolver

So far we focused on the time evolution problem. In this section we tackle the ground state problem. As already laid out in the introduction, the ground state problem is QMA-complete and we do not expect a general quantum speedup in contrast to the time evolution problem. However such exact statements may affect practical approaches only marginally. Here we will show that the variational principle can also be applied for states prepared on quantum computers. We want to find a state $|\psi\rangle$ with low energy

$$E[\psi] = \frac{\langle\psi| H |\psi\rangle}{\langle\psi|\psi\rangle}. \tag{61}$$

The variational principle gives as an upper bound $E[\psi] \geq E_0$ for every normalized $|\psi\rangle$, with equality only in the ground-state subspace. This property is used in an algorithm called the *Variational Quantum Eigensolver* (VQE). The VQE uses a parameterized circuit $U(\boldsymbol{\vartheta})$ to prepare a quantum state $|\psi(\boldsymbol{\vartheta})\rangle$ and determine a cost function $C(\boldsymbol{\vartheta})$ that is the energy expectation value of the state,

$$|\psi(\boldsymbol{\vartheta})\rangle = U(\boldsymbol{\vartheta}) |0\rangle^{\otimes n}, \quad C(\boldsymbol{\vartheta}) = \langle\psi(\boldsymbol{\vartheta})| H |\psi(\boldsymbol{\vartheta})\rangle . \tag{62}$$

The idea is now to minimize $C(\boldsymbol{\vartheta})$ using the parameters $\boldsymbol{\vartheta}$ in the unitary $U(\boldsymbol{\vartheta})$. One might ask, how to realize such a complicated subroutine like minimization on a quantum computer. The answer is: one does not. Instead we rather use the quantum computer to sample the energy expectation value and then give that task to a classical computer, which executes a minimization algorithm, such as steepest descent. More precisely, the quantum processor estimates $C(\boldsymbol{\vartheta})$ by measuring Pauli strings in a decomposition $H = \sum_a h_a P_a$. The classical optimizer then proposes new parameters which are again used by the quantum processor to get a new estimate of the Pauli strings. This hybrid quantum-classical loop is closed until the cost no longer decreases, see Fig. 4.

The main open question we have not yet addressed is the choice of the Ansatz, i.e., the parameterized circuit $U(\boldsymbol{\vartheta})$. Various strategies have been proposed to design different ansatz schemes [32]. So called *Hardware-efficient ansätze* [33] are tailored to the native gate sets and

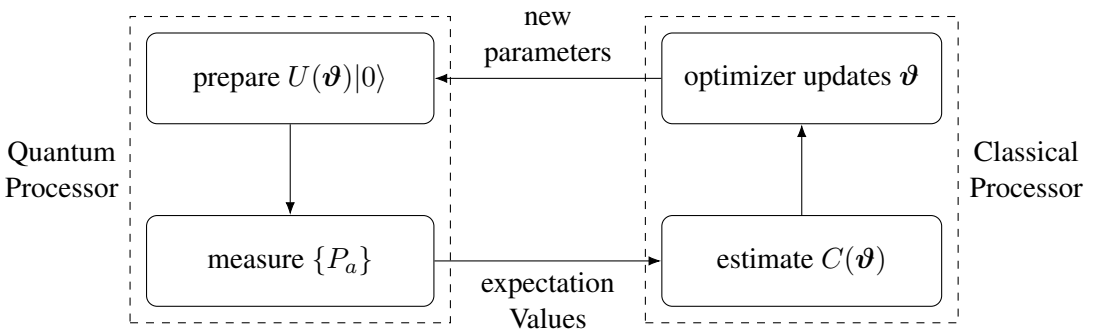


Fig. 4: *Hybrid structure of the Variational Quantum Eigensolver. The quantum device prepares and measures states. The classical computer constructs the cost function and chooses the next parameters.*

connectivity of quantum devices, thereby reducing circuit depth, but they often suffer from limited expressivity. In contrast, problem-inspired approaches such as the *Variational Hamiltonian Ansatz* [34] aim to mimic adiabatic state preparation, potentially improving convergence, although at the cost of reduced flexibility or increased circuit depth. More recently, approaches such as the *Quantum-optimal-control-inspired ansatz* [35] have explored symmetry breaking and control-theoretic concepts to enhance expressivity and optimization performance. Despite these advances, identifying ansätze that simultaneously balance expressivity, trainability, and noise resilience remains an open problem [36, 37].

We want to explore what is currently possible to achieve with a useful example, the one-dimensional axial next nearest neighbor Ising model, or ANNNI model [38]. The Hamiltonian is given by

$$H(J_2, B_x) = -J_1 \sum_i Z_i Z_{i+1} + J_2 \sum_i Z_i Z_{i+2} + B_x \sum_i X_i. \tag{63}$$

Here $J_1 > 0$ favors ferromagnetic order, while $J_2 > 0$ favors antiferromagnetic alignment at distance two. The transverse field B_x introduces quantum fluctuations. We set $J_1 = 1$ as the unit of energy and use the ratio J_2/J_1 as the control parameter. For $B_x = 0$ the Hamiltonian is classical. If $J_2/J_1 < 1/2$, the ferromagnetic term dominates and the two classical states $|00\cdots0\rangle$ and $|11\cdots1\rangle$ span the ground state manifold. If $J_2/J_1 > 1/2$, the next nearest neighbor term enforces a period four pattern such as

$$|\psi_{22}\rangle \simeq |00110011\cdots\rangle\,, \tag{64}$$

up to translations and global spin flip. The point $J_2/J_1 = 1/2$ is highly degenerate at zero field. A small transverse field lifts this degeneracy and reveals a richer finite size phase diagram. Besides the ferromagnetic and period four regions one finds modulated states and a paramagnetic region at sufficiently large field. Strictly speaking these are true phases only in the thermodynamic limit. On a finite quantum processor they appear as phaselike regions, i.e., regions in which the optimized ground state has a characteristic structure and changes rapidly at special parameter values. This makes the model a good benchmark. The Hamiltonian is simple in terms of Pauli operators, but its ground states are sensitive to frustration and to quantum fluctuations.

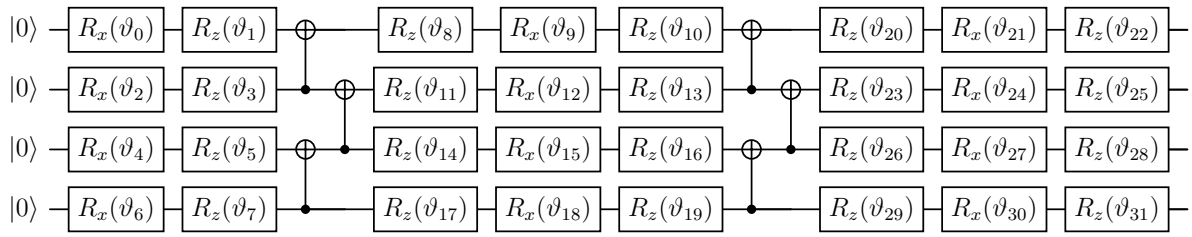


Fig. 5: *One layer of the VQE ansatz for the ANNNI model for four qubits. Each single qubit rotation carries an independent variational parameter. The layer starts with rotations around the x and z axes, followed by a staggered pattern of* CNOT *gates. The same entangling pattern is then repeated after a second block of single qubit rotations. More layers repeat this structure with new variational parameters.*

For this model we use a hardware efficient circuit assuming a one-dimensional device connectivity. One layer for four qubits is shown in Fig. 5. It starts from local rotations, then applies controlled not gates. Local rotations between and after the entangling gates provide the variational parameters. The same pattern can be tiled to larger systems. For L layers the circuit is obtained by repeating this block L times with independent angles. Increasing L enlarges the set of states that can be represented. In particular, correlations can spread over longer distances and the circuit can better interpolate between the ferromagnetic state, the period four state, and states in between. At the same time, every extra layer adds two qubit gates, and these gates usually dominate the noise. The useful depth is therefore not set only by expressivity but by a compromise between the variational error of a shallow circuit and the hardware error of a deep circuit.

The energy alone is not always the most reliable observable for detecting changes in the wave function on noisy hardware. A more sensitive quantity is the fidelity susceptibility. Let $p = J_2/J_1$ and let $|\psi(p)\rangle$ denote the ground state or the optimized VQE state. The fidelity between neighboring points is

$$F(p,\delta) = \big|\, \langle \psi(p-\delta) | \psi(p+\delta) \rangle \,\big|. \tag{65}$$

For small δ one defines the fidelity susceptibility through

$$F(p,\delta) = 1 - \chi_F(p)\delta^2 + \mathcal{O}(\delta^4). \tag{66}$$

It measures how quickly the ground state changes when the Hamiltonian parameter is varied. Away from a transition the state changes smoothly and χ_F is small. Near a level crossing or an avoided crossing the state changes strongly and χ_F develops a peak. We can determine the fidelity susceptibility also from perturbation theory. If $H(p) = H_0 + pH_A$ and $|n(p)\rangle$ are exact eigenstates, then

$$\chi_F(p) = 2 \sum_{n \neq 0} \frac{\big|\, \langle n(p)|\, H_A\, |0(p)\rangle \,\big|^2}{\big(E_n(p) - E_0(p)\big)^2}\,. \tag{67}$$

The spectral gap therefore enters in the denominator. This is why the fidelity susceptibility can capture a gap closure. It is also largely agnostic to the order parameter, which is useful for frustrated models where the relevant order may be modulated or hard to guess.

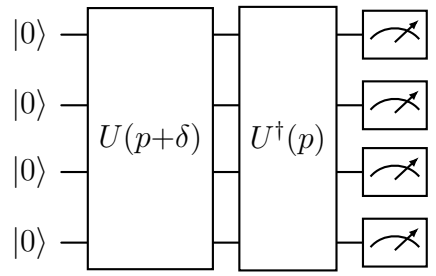


Fig. 6: *Overlap circuit for the fidelity susceptibility. The probability of measuring* 0000 *gives the squared overlap between two variational states. Repeating the circuit for nearby values of* $p = J_2/J_1$ *yields the finite difference estimate of* χ_F.

On a quantum computer the overlap in Eq. (65) can be measured with a simple inverse circuit. Suppose that $U(p)$ prepares the optimized state at p and that $U(p+\delta)$ prepares the optimized state at the nearby point. The circuit in Fig. 6 applies $U(p+\delta)$ and then the inverse of $U(p)$. The probability to measure the all zero bit string is

$$P_0 = \left| \langle 0|^{\otimes N} U^\dagger(p) U(p+\delta) |0\rangle^{\otimes N} \right|^2 = \left| \langle \psi(p) | \psi(p+\delta) \rangle \right|^2. \tag{68}$$

Thus the fidelity is obtained as $F = \sqrt{P_0}$. In a VQE calculation there is one further subtlety. The optimizer can choose symmetry broken representatives of the same degenerate subspace at neighboring values of p. Then the direct overlap may be small even though both states describe the same phase. One can partially correct this by maximizing the overlap over symmetry operations of the Hamiltonian, for example the global bit flip $X^{\otimes N}$ and, for periodic boundary conditions, lattice translations. This gives a fidelity measure that is less sensitive to arbitrary choices made by the variational optimizer.

In Fig. 7 we see the results of the VQE approach for the ANNNI model. On the top the correlation functions demonstrate the transitions of the ground state due to spatial fluctuations, e.g. the change of the oscillation frequency in the intermediate J_2/J_1 ratio. In the bottom panel the ground state energy is compared to an ideal VQE simulation and to the VQE obtained from the quantum computer. The large noise in the energy prohibits any statements on phase transitions in the system. In contrast the fidelity susceptibility clearly identifies strong changes in the ground state wave function. Thus the VQE should not be judged only by the absolute accuracy of the energy. The same measured states can contain robust information about changes in the wave function, and this information can be easier to extract on noisy devices.

4 Quantum error correction

4.1 Introduction to error correction

Physical qubits are never perfectly isolated from their environment. Typical errors are qubit relaxation, dephasing and coherent errors of the gates, e.g., overrotations. If such errors accumulate with circuit depth, then the simulated time evolution can become dominated by errors

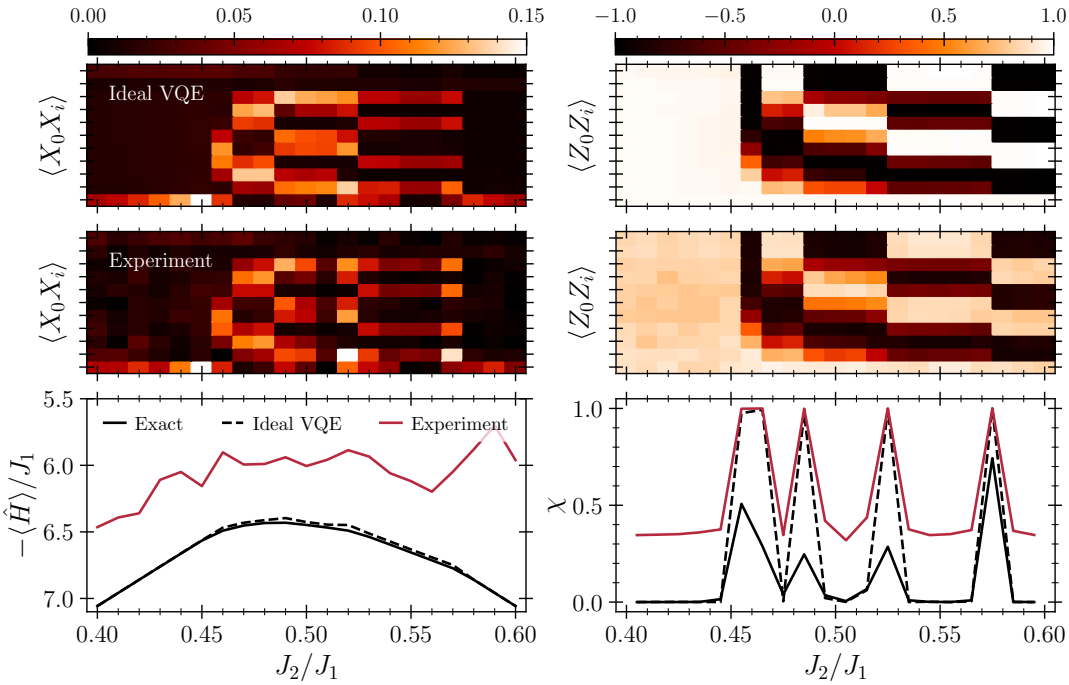


Fig. 7: Variational Quantum Eigensolver solution of the ANNNI model. *The correlation functions $\langle X_0X_i\rangle$ and $\langle Z_0Z_i\rangle$ alongside energy and fidelity susceptibility χ for $N = 12$ at $B_x = 0.1$ with open boundary conditions. The top row shows the ideal noise-free VQE correlation functions, while the middle row shows the experimental results on* `ibm_kyoto` *quantum processor, starting from site $i = 1$ on the y-axis. The J_2/J_1 axes are shared with the energy and fidelity susceptibility plotted on the bottom rows.*

rather than by the Hamiltonian we wanted to study. Quantum error correction tries to overcome this problem. The core idea is that quantum information can be stored nonlocally in a larger Hilbert space and we can use targeted measurements, that do not perturb this information, to diagnose errors without measuring the encoded state itself.

We begin with a classical example. Suppose that we want to protect one classical bit against a bit flip. The simplest code stores the logical bit redundantly in three physical bits,

$$0 \mapsto 000, \qquad 1 \mapsto 111. \tag{69}$$

If at most one bit flips, the corrupted strings are still closer to the intended codeword than to the other one. For example, the string 010 is interpreted as 000, while 101 is interpreted as 111. This is the majority vote principle. The redundancy has not removed errors, but it has converted a small number of errors into a detectable pattern.

The states 000 and 111 are then what define the *code space*.

The same approach can be expressed in terms of parity checks. For three bits b_1, b_2, b_3 one measures whether neighboring bits agree. The two parity checks are

$$s_1 = b_1 \oplus b_2, \qquad s_2 = b_2 \oplus b_3. \tag{70}$$

For the codewords 000 and 111 both parities vanish. If the first bit flips, one obtains $s_1 = 1$ and $s_2 = 0$. If the second bit flips, both parities are one. If the third bit flips, one obtains $s_1 = 0$ and

$s_2 = 1$. Thus the two parity checks reveal the position of the error. They do not reveal whether the stored bit was zero or one. This distinction is essential. The measurement extracts only the error *syndrome*, not the logical information.
The quantum version of the same idea encodes one logical qubit into three physical qubits as

$$|0_L\rangle = |000\rangle\,, \qquad |1_L\rangle = |111\rangle\,, \tag{71}$$

so that an arbitrary logical state is

$$|\psi_L\rangle = \alpha\,|000\rangle + \beta\,|111\rangle\,. \tag{72}$$

Note that this is different from copying a quantum state, e.g., $|\phi\rangle \rightarrow |\phi\rangle\,|\phi\rangle$, which the no cloning theorem forbids. The state is encoded into a two dimensional subspace of the eight dimensional Hilbert space of three qubits. A bit flip on the first, second, or third physical qubit gives

$$X_1\,|\psi_L\rangle = \alpha\,|100\rangle + \beta\,|011\rangle\,, \tag{73}$$
$$X_2\,|\psi_L\rangle = \alpha\,|010\rangle + \beta\,|101\rangle\,, \tag{74}$$
$$X_3\,|\psi_L\rangle = \alpha\,|001\rangle + \beta\,|110\rangle\,. \tag{75}$$

These three error spaces are orthogonal to the code space and to each other. We can therefore identify which error occurred without collapsing the phases α and β.
The quantum parity checks are measurements of the two commuting operators

$$S_1 = Z_1 Z_2, \qquad S_2 = Z_2 Z_3. \tag{76}$$

Both logical codewords are eigenstates of S_1 and S_2 with eigenvalue $+1$. This remains true for every superposition in the code space. If a physical bit flip X_i occurs, it anticommutes with the parity checks that contain qubit i. The measured eigenvalues of S_1 and S_2 therefore provide the syndrome. The outcomes $(+1, +1)$ mean no bit flip, $(-1, +1)$ means a flip on qubit one, $(-1, -1)$ means a flip on qubit two, and $(+1, -1)$ means a flip on qubit three. After this diagnosis one applies X_i to the identified qubit. The correction maps the state back to $|\psi_L\rangle$.
The operators S_1 and S_2 are called *stabilizers* because they leave the code space invariant. The code space is the common $+1$ eigenspace of all stabilizers. Error correction is therefore a repeated sequence of syndrome measurements and conditional operations. The syndrome measurements are allowed because they commute with the logical information. They ask whether the state is still inside the code space, and if not, in which orthogonal error space it lies.

4.2 Bit and phase flip errors

A qubit has more possible errors than a classical bit. The Pauli matrix X exchanges $|0\rangle$ and $|1\rangle$ and is therefore the quantum analogue of a bit flip. The Pauli matrix Z leaves $|0\rangle$ unchanged and changes the sign of $|1\rangle$,

$$Z\,|0\rangle = |0\rangle\,, \qquad Z\,|1\rangle = -\,|1\rangle\,. \tag{77}$$

This is a relative phase flip. It has no classical analogue for a single computational basis state, but it changes coherent superpositions and is therefore just as damaging for a quantum computation. The Pauli matrix Y is, up to a global phase, the product of both errors, i.e., $Y = iZX$. For this reason it is enough to understand how to detect and correct X and Z errors.
The three qubit code above corrects one bit flip. It does not correct one phase flip, since

$$Z_1 |\psi_L\rangle = \alpha |000\rangle - \beta |111\rangle . \tag{78}$$

This state is still in the common $+1$ eigenspace of Z_1Z_2 and Z_2Z_3. The parity checks therefore cannot see the error. To correct phase flips one can use the same code after a change of basis from the Z to the X basis. This basis transformation is realized through the Hadamard gate,

$$HZH = X. \tag{79}$$

Thus a phase flip in the computational basis is a bit flip in the X basis. A three qubit phase flip code is obtained from the bit flip code by applying Hadamard gates to all qubits. With

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \qquad |-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}, \tag{80}$$

one may encode

$$|0_L\rangle = |+++\rangle , \qquad |1_L\rangle = |---\rangle . \tag{81}$$

The stabilizers are now

$$S_1 = X_1X_2, \qquad S_2 = X_2X_3. \tag{82}$$

Their measurement detects a single Z error in exactly the same way that Eq. (76) detects a single X error.
A full quantum error correcting code combines both ideas. The smallest historical example is the nine qubit code developed by Shor [39]. Modern stabilizer codes use the same principles but try to minimize the use large stabilizer measurements and large number of qubits.
There is one important conceptual point that allows quantum error correction to be also applied to continuous errors. For example, a small unwanted rotation around the x axis on the first qubit has the form

$$R_x(\epsilon) = \exp\left(-\mathrm{i}\frac{\epsilon}{2}X_1\right) = \cos\left(\frac{\epsilon}{2}\right) I - \mathrm{i}\sin\left(\frac{\epsilon}{2}\right) X_1. \tag{83}$$

Acting on an encoded state this gives a coherent superposition of no error and a full bit flip,

$$R_x(\epsilon) |\psi_L\rangle = \cos\left(\frac{\epsilon}{2}\right) |\psi_L\rangle - \mathrm{i}\sin\left(\frac{\epsilon}{2}\right) X_1 |\psi_L\rangle . \tag{84}$$

Before the syndrome measurement this is not a classical random error. It is a quantum superposition of two orthogonal syndrome sectors. The parity measurement projects the state into one of them. With probability $\cos^2(\epsilon/2)$ the syndrome is trivial and the state is projected back to the code space, thus no error occurred. With probability $\sin^2(\epsilon/2)$ the syndrome is that of a full X_1 error. In that branch we apply X_1 and again recover $|\psi_L\rangle$. The measurement has discretized the continuous error into a finite set of correctable alternatives.
The same reasoning applies to general errors, since any operator can be expanded in the Pauli string basis. Thus stabilizer error correction focuses on a discrete set of Pauli errors even though the microscopic dynamics of noise is continuous.

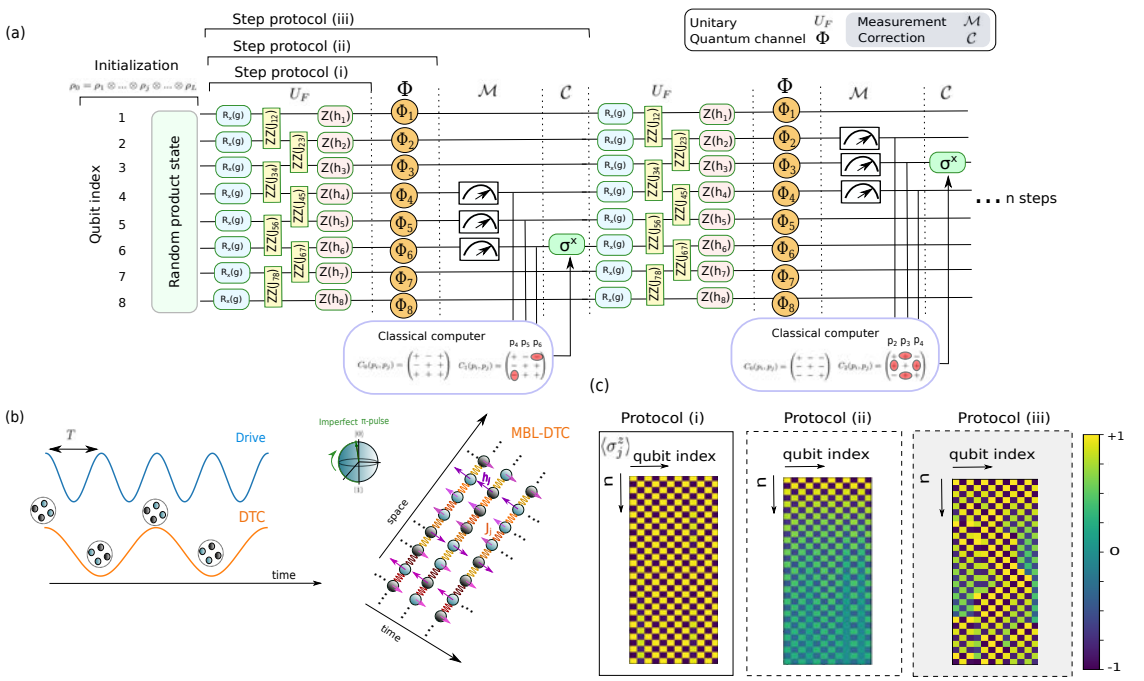


Fig. 8: *Quantum-classical feedback scheme for the stabilization of a discrete time crystal on a quantum computer. a) The time evolution of the system is subject to a noise channel Φ_i. Afterwards a parity measurements determines possibly flipped spins and corrects them. b) Shows the mechanism behind the many-body localized time crystal phase. c) Demonstrates the effect of exact unitarity, noise and noise plus correction scheme. Adopted from [41].*

4.3 Partial error correction for non-equilibrium phases of matter

Quantum error correction aims to preserve an arbitrary logical state for an arbitrarily long circuit. This is a very demanding goal. For many quantum simulation tasks this is not necessary, as we do not need to protect all quantum information equally well. Here we discuss an example that protects only the part of the dynamics that defines a phase of matter. One measures and corrects selected stabilizers or selected observables that are tied to the order of interest, while accepting that other information is still lost.

We focus on a many body localized discrete time crystal (MBL-DTC). Such a system is characterized by a robust subharmonic response to a periodic drive. That means if the Hamiltonian has period T, local observables oscillate with period $nT, n > 1$, in our case $n = 2$. In an ideal closed system this period doubling can persist indefinitely in the thermodynamic limit because disorder and interactions prevent the system from heating to a featureless state. On a noisy quantum processor the situation is different since the oscillations decay due to decoherence after a limited number of Floquet cycles [40].

The most studied model is a one dimensional periodically kicked Ising chain with Floquet unitary

$$U_F = \exp\left(-\mathrm{i}\frac{T}{4}\sum_{j=1}^{L-1} J_j Z_j Z_{j+1}\right)\exp\left(-\mathrm{i}\frac{T}{2}\sum_{j=1}^{L} h_j Z_j\right)\exp\left(-\mathrm{i}\frac{\pi g}{2}\sum_{j=1}^{L} X_j\right). \tag{85}$$

The couplings J_j and fields h_j are disordered. The last factor is close to a global spin flip when g is close to one. Therefore a product state in the Z basis is approximately mapped to its spin reversed state after one period, and back again after two periods. The interactions and disorder stabilize this response against small imperfections in the pulse, i.e., $g \approx 1$. This is the many body localized discrete time crystal regime.

In Fig. 8 the circuit for the MBL-DTC is shown, including a noise channel Φ_i and a correction scheme [41]. Here the logical space are the classical bit strings that we want to protect from local flips. The error correction code measures a small contiguous region of M qubits in the Z basis after each Floquet unitary. For a product state in the computational basis the relevant local information is contained in the local Z parity data. This parity is constant in the ideal MBL-DTC phase. Thus, if one physical qubit has flipped, all correlations involving that qubit change sign. The qubit with the largest number of such sign changes is identified as the most likely error location. A corrective X operation is then applied to that qubit. We see the effect of the stabilization in comparison to a perfect quantum computer (unitary) and a noisy one in Fig. 8 c).

Such partial correction schemes can stabilize a phase even in the absence of fully error corrected quantum processor. For quantum simulation this is a promising viewpoint. Many target problems do not require preserving an unknown quantum state with arbitrary accuracy. They require preserving a physically meaningful structure, such as a symmetry sector, a conserved quantity, a topological constraint, or a pattern of spatiotemporal order.

References

[1] H. De Raedt, J. Kraus, A. Herten, V. Mehta, M. Bode, M. Hrywniak, K. Michielsen, and T. Lippert, Future Gener. Comput. Syst. **183**, 108592 (2026)

[2] Y.I. Manin: *Vychislimoe i Nevychislimoe [Computable and Uncomputable]* (Sovetskoye Radio, Moscow, 1980) In Russian; English translation in *Mathematics as Metaphor: Selected Essays* (AMS, 2007)

[3] P. Benioff, J. Stat. Phys. **22**, 563 (1980)

[4] R.P. Feynman, Int. J. Theor. Phys. **21**, 467 (1982)

[5] D. Deutsch, Proceedings of the Royal Society of London A **400**, 97 (1985)

[6] P.W. Shor, pp. 124–134 in *Proceedings of the 35th Annual Symposium on Foundations of Computer Science (FOCS)* (IEEE Computer Society Press, 1994)

[7] P.W. Shor, SIAM J. Comput. **26**, 1484 (1997)

[8] S. Lloyd, Science **273**, 1073 (1996)

[9] S. Aaronson: *Quantum Computing Since Democritus* (Cambridge University Press, 2013)

[10] A.M. Gleason, J. Math. Mech. **6**, 885 (1957)

[11] S. Aaronson, arXiv:quant-ph/0401062 (2004)

[12] P. Barrios Hita, A. Trushechkin, H. Kampermann, M. Epping, and D. Bruß, Phys. Rev. Lett. **136**, 240202 (2026)

[13] A.Y. Kitaev, A.H. Shen, and M.N. Vyalyi: *Classical and Quantum Computation*, Graduate Studies in Mathematics, Vol. 47 (American Mathematical Society, 2002)

[14] J. Kempe, A. Kitaev, and O. Regev, SIAM J. Comput. **35**, 1070 (2006)

[15] H.F. Trotter, Proc. Am. Math. Soc. **10**, 545 (1959)

[16] A.M. Childs, Y. Su, M.C. Tran, N. Wiebe, and S. Zhu, Phys. Rev. X **11**, 011020 (2021)

[17] M. Suzuki, J. Math. Phys. **32**, 400 (1991)

[18] D.W. Berry, G. Ahokas, R. Cleve, and B.C. Sanders, Commun. Math. Phys. **270**, 359 (2007)

[19] G.H. Low and I.L. Chuang, Phys. Rev. Lett. **118**, 010501 (2017)

[20] I.M. Georgescu, S. Ashhab, and F. Nori, Rev. Mod. Phys. **86**, 153 (2014)

[21] J.I. Cirac and P. Zoller, Nat. Phys. **8**, 264 (2012)

[22] M. Greiner, O. Mandel, T. Esslinger, T.W. Hänsch, and I. Bloch, Nature **415**, 39 (2002)

[23] I. Bloch, J. Dalibard, and S. Nascimbène, Nat. Phys. **8**, 267 (2012)

[24] R. Blatt and C.F. Roos, Nat. Phys. **8**, 277 (2012)

[25] M.A. Nielsen and I.L. Chuang: *Quantum Computation and Quantum Information* (Cambridge University Press, 2010), 10th anniversary ed.

[26] A. Barenco, C.H. Bennett, R. Cleve, D.P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J.A. Smolin, and H. Weinfurter, Phys. Rev. A **52**, 3457 (1995)

[27] D.P. DiVincenzo, Phys. Rev. A **51**, 1015 (1995)

[28] A.Y. Kitaev, Russ. Math. Surv. **52**, 1191 (1997)

[29] C.M. Dawson and M.A. Nielsen, Quantum Inf. Comput. **6**, 81 (2006)

[30] B. Fauseweh and J.-X. Zhu, Quantum Inf. Process. **20**, 138 (2021)

[31] Z. Jiang, A. Kalev, W. Mruczkiewicz, and H. Neven, Quantum **4**, 276 (2020)

[32] J. Tilly, H. Chen, S. Cao, D. Picozzi, K. Setia, Y. Li, E. Grant, L. Wossnig, I. Rungger, G.H. Booth, and J. Tennyson, Phys. Rep. **986**, 1 (2022)

[33] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J.M. Chow, and J.M. Gambetta, Nature **549**, 242 (2017)

[34] D. Wecker, M.B. Hastings, and M. Troyer, Phys. Rev. A **92**, 042303 (2015)

[35] A. Choquette, A. Di Paolo, P.K. Barkoutsos, D. Sénéchal, I. Tavernelli, and A. Blais, Phys. Rev. Res. **3**, 023092 (2021)

[36] S. Liu, S.-X. Zhang, C.-Y. Hsieh, S. Zhang, and H. Yao, Phys. Rev. B **107**, 024204 (2023)

[37] C. Cao, Y. Zhou, S. Tannu, N. Shannon, and R. Joynt, Quantum **9**, 1942 (2025)

[38] K. Lively, T. Bode, J. Szangolies, J.-X. Zhu, and B. Fauseweh, Phys. Rev. Res. **6**, 043254 (2024)

[39] A.R. Calderbank and P.W. Shor, Phys. Rev. A **54**, 1098 (1996)

[40] X. Mi, *et al.*, Nature **601**, 531 (2022)

[41] G. Camacho and B. Fauseweh, Phys. Rev. Res. **6**, 033092 (2024)