


## Quantum Monte Carlo methods and stabilizer states

Bhilahari Jeevanesan 

Remote Sensing Technology Institute, German Aerospace Center DLR, 82234 Wessling, Germany

 (Received 1 November 2024; revised 5 May 2025; accepted 13 June 2025; published 2 July 2025)

The quantum Monte Carlo technique known as the stochastic series expansion (SSE) relies on a crucial no-branching condition: the SSE sampling is carried out in the computational basis, and the no-branching assumption ensures that superpositions of basis states do not appear when operators are applied. Without this proviso, the number of complex amplitudes would grow exponentially with the number of qubits and would eventually overwhelm the memory and processing power of a classical computer. However, the action of Clifford group elements on stabilizer states can be very efficiently described without resorting to an amplitude description. We explore how stabilizer states allow an extension of the SSE technique, and we give an example of a toy model that can be studied in this way. The method is also illustrated using the transverse-field Ising model.

DOI: [10.1103/PhysRevA.112.012603](https://doi.org/10.1103/PhysRevA.112.012603)

### I. INTRODUCTION

The difficulty of classically simulating quantum systems stems from the typically enormous Hilbert spaces and the need to sum complex probability amplitudes over all possible paths from the initial to the final state. The latter summation can lead to delicate cancellations that can be difficult to keep track of. Nevertheless, over the past decades, clever computational tools have been developed for certain physical systems. Examples of these are the various types of quantum Monte Carlo simulations based on Euclidean path integrals [1–3]. They all rely on the fact that occasionally, it is possible to rewrite a system’s thermal or ground state in terms of purely real and non-negative probability amplitudes. Then the latter can be interpreted as weights of states that can be sampled using the Markov chain Monte Carlo approach [4]. However, this approach can fail when the Hamiltonian has a sign problem that persists under local basis change (i.e., after acting with a low-depth unitary circuit). Usually, this is interpreted as a case where quantum mechanics is an essential component of the problem and obstructs efficient classical sampling [5,6]. Contrast this with the situation in quantum computing, where it was shown by Lloyd that a universal quantum system can simulate any other local quantum system efficiently [7], proving a statement that was conjectured by Feynman in [8].

Despite much work, the limits of classical simulability are not well understood. Further progress may be made by pushing the boundaries on what is classically simulable. In this paper, we contribute to this discussion by exploring an extension of the stochastic series expansion (SSE) Monte

Carlo technique [9–12]. The SSE technique relies on a high-temperature expansion of the partition function. When the expansion terms can be interpreted as non-negative weights, a Monte Carlo sampling is often possible. During the simulation, powers of the Hamiltonian act on quantum states. In order to make this tractable on a classical computer, the SSE technique uses a *no-branching* provision that avoids the formation of quantum superpositions. As a consequence of this, it is not possible to arbitrarily modify an operator string by inserting or removing an (off-diagonal) operator since this will quickly result in a zero-weight state. Thus, inspired by the algorithm of Swendsen and Wang [13], many clever loop

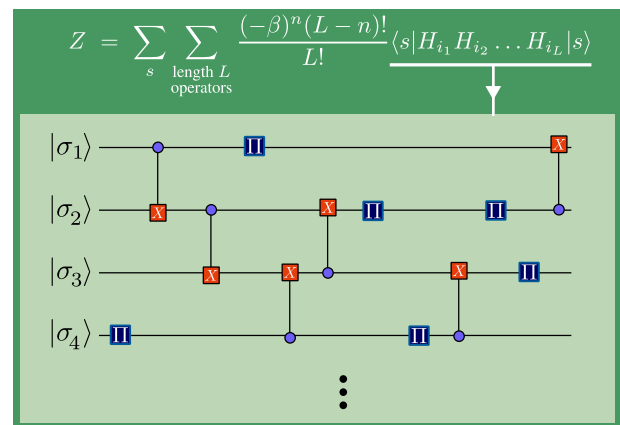


FIG. 1. When the partition function  $Z$  is expanded as a stochastic series, the occurring operator string can be interpreted as a Clifford circuit. We sample the terms in the partition function using a Markov chain Monte Carlo procedure and evaluate the matrix elements by handing them off to our Clifford circuit subroutine. The algorithm applies the circuit to the state  $|s\rangle = |\sigma_1\rangle \otimes \dots \otimes |\sigma_N\rangle$  and computes the inner product with  $|s\rangle$  by exploiting rules for stabilizer states. The example shows controlled-NOT and  $\Pi$  projection operations as they appear in the series expansion for the controlled-NOT (CNOT) toy model of Sec. III.

\*Contact author: [billa.j@gmail.com](mailto:billa.j@gmail.com)

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International license](https://creativecommons.org/licenses/by/4.0/). Further distribution of this work must maintain attribution to the author(s) and the published article’s title, journal citation, and DOI.

and cluster algorithms have been proposed over the years to efficiently sample the state and operator configurations [11,14,15]. The SSE technique has also been extended to quantum computers in the form of a quantum SSE algorithm [16]; see, also, [17] for a review.

The present paper shows that the no-branching condition can be removed even on a classical computer for some models by making use of *stabilizer states*. These states have been extensively used in the construction of quantum stabilizer codes [18–21]. They are distinguished within Hilbert space by having an elegant formalism for their description that enables rapid manipulation [21] under the action of the *Clifford group*. Despite their classical tractability, they can have high degrees of quantum entanglement [22].

We start in Sec. II with a brief summary of the stochastic series expansion and introduce the idea of stabilizer states and their compact description. In Sec. III, we demonstrate the use of stabilizer states in quantum Monte Carlo simulations by picking a toy Hamiltonian and illustrating the algorithm in detail. Finally, Sec. IV presents a simulation of the transverse-field Ising model using this method and sketches how it may be extended to certain  $\mathbb{Z}_2$  gauge theories. The GitHub repository [23] contains the C++ implementations of the stabilizer Monte Carlo algorithm and exact diagonalization programs used in this paper.

## II. A BRIEF REVIEW OF SSE QUANTUM MONTE CARLO AND THE STABILIZER FORMALISM

The SSE Monte Carlo method was pioneered by Handscomb [24,25] in the 1960s and later turned into a powerful tool by Sandvik in seminal publications [10,11,14,15]. It allows the study of interacting quantum-spin systems and has been successfully applied to investigate exotic phases of matter.

Consider a Hamiltonian  $H$  on  $N$  qubits. In a first step, the partition function is expanded in powers of  $\beta$ ,

$$\begin{aligned} Z(\beta) &= \text{Tr}[e^{-\beta H}] = \sum_s \langle s | e^{-\beta H} | s \rangle \\ &= \sum_s \sum_{n=0}^{\infty} \frac{(-\beta)^n}{n!} \langle s | H^n | s \rangle, \end{aligned} \quad (1)$$

where the  $|s\rangle$  are the  $2^N$  computational basis states, e.g., in the  $Z$  basis. We first explain the SSE formalism of Sandvik before we discuss how stabilizer states can be used. The Hamiltonian is split up into a sum of terms  $H = \sum_i H_i$  for convenience. It yields a string of operators upon expansion,

$$Z = \sum_s \sum_{i_1, \dots, i_n} \sum_{n=0}^{\infty} \frac{(-\beta)^n}{n!} \langle s | H_{i_1} H_{i_2} \dots H_{i_n} | s \rangle. \quad (2)$$

The SSE algorithm [15] works by imposing a cutoff  $L$  on the maximum order of this series such that the error remains negligible; see discussion in Sec. III C. Then, by padding the operator strings with identity operators  $\mathbf{1}$ , all of them end up having length  $L$ . A string of length  $n$  can be padded in  $\binom{L}{L-n}$

ways with identity operators to reach length  $L$ . Thus one can rewrite the series as

$$\begin{aligned} Z &= \sum_s \sum_{\substack{\text{length } L \\ \text{operators}}} \frac{(-\beta)^n (L-n)!}{L!} \langle s | H_{i_1} H_{i_2} \dots H_{i_n} | s \rangle \quad (3) \\ &\equiv \sum_s \sum_{\substack{\text{length } L \\ \text{operators}}} W(|s\rangle, [H_{i_1}, \dots, H_{i_n}]), \end{aligned} \quad (4)$$

where the factor  $\binom{L}{L-n}^{-1}$  compensates for overcounting identical terms. The sum is extended over all combinations of length- $L$  operator strings and  $n$  refers to the number of non-identity operators in it. In other words,  $L-n$  of the  $H_i$  are now identity operators. In the second line, we introduced the notation  $W$  for the weight of each configuration.

To summarize, the partition function (1) has been transformed into a sum of matrix elements involving  $L$  operators. Each term can be interpreted as the weight of a physical process where the system begins and ends in state  $|s\rangle$  after the action of  $L$  operators. Thus, if the Hamiltonian  $H$  describes a lattice system of dimension  $d$ , then the representation (4) describes a system of dimension  $d+1$  where the additional dimension is periodic with  $L$  lattice sites [26].

The crucial step in SSE Monte Carlo is that the decomposition of  $H$  into terms  $\sum_i H_i$  is done in such a way that the action of any  $H_i$  on a computational basis state  $|s\rangle$  yields another computational basis state  $|s'\rangle$ . This can usually be accomplished by breaking  $H$  into sufficiently small pieces and is called the *no-branching* provision. It guarantees that no superpositions over basis states are formed during the simulation process and thereby ensures easy bookkeeping. General superpositions of basis states are practically impossible to keep track of on a classical computer, except when system sizes are small.

The sampling of the partition function now proceeds by setting up a Markov chain [4,27] with each element of the chain being a state  $|s\rangle$  together with a string of operators. The weight of the Markov chain element is given by the corresponding term in Eq. (3). An efficient simulation is only possible if the weights are all non-negative. Occasionally, one can manipulate the Hamiltonian terms  $H_i$  by the addition of constants and render all weights non-negative. If this is not possible, one faces the Monte Carlo sign problem. It can be dealt with, in a brute-force manner, by absorbing the sign of the weight into the observable. But, generally, this leads to long convergence times when calculating observable averages and the variances diverge exponentially with system size.

A Markov chain update occurs by proposing a change of the state  $|s\rangle$  or the insertion or deletion of an operator in the string. The proposal is accepted or rejected to satisfy the detailed balance condition. The latter involves the ratios of the weights; an example appears in Sec. III B. Note that if only one operator is inserted or removed, most factors in the weights are unchanged and will therefore cancel out. This leads to simple expressions for the update rules. However, since the application of the  $H_{i_k}$  in Eq. (3) always yields basis states, randomly inserting or rejecting an operator will often result in a state with zero weight. Instead, the insertion and removal of operators have to be carefully crafted. Furthermore, local updates of the operator string correspond to local

changes of the periodic word line. Thus the system cannot explore all winding number sectors. For certain Hamiltonians, smart algorithms have been discovered that insert and delete a large number of operators simultaneously in order to achieve fast changes of the sampled configurations [11,14,15].

In this paper, we ask whether the no-branching condition is essential or if there is an efficient way to track the branching of the states. To this end, we return to the expression in Eq. (3) and consider an interpretation of the sequence of operators as a quantum circuit. This is possible if each  $H_i$  is either proportional to a unitary or to a projection operator, with the latter describing the outcomes of partial projective measurements. This observation by itself would not be particularly useful since an arbitrary quantum circuit still takes an exponential time to simulate classically. It has been proposed to evaluate the matrix element in (3) by means of quantum computers, resulting in a quantum SSE algorithm [16]; see [17] for a helpful review.

However, in the absence of large-scale quantum computers, progress can still be made through classical computation if the unitary operator is a *Clifford circuit*, see Fig. 1. In this case, efficient classical simulation is possible, as formalized by the Gottesman-Knill theorem [28].

We next give a brief summary of some results on stabilizer groups and minimize the discussion to what will be needed later. For broader expositions on this topic, see [28–30]. We will confine ourselves to the case where the stabilizer operators are elements of the Pauli group  $\mathcal{P}_N$  on  $N$  qubits. The latter group is defined as an  $N$ -fold tensor product of the Pauli group on 1 qubit  $\mathcal{P}_1$ , i.e.,  $\mathcal{P}_N = \mathcal{P}_1^{\otimes N}$ . The group  $\mathcal{P}_1$  contains 16 elements, which are the identity operator  $\mathbf{1}$  and the Pauli operators  $X, Y, Z$  with particular phases,

$$\mathcal{P}_1 = \{i^n \cdot \mathcal{O} | n \in \{0, 1, 2, 3\}, \mathcal{O} \in \{\mathbf{1}, X, Y, Z\}\}. \quad (5)$$

Consider the Hilbert space of  $N$  qubits. An operator  $G \in \mathcal{P}_N$  is said to stabilize a state  $|\psi\rangle$  if  $G|\psi\rangle = |\psi\rangle$ . In other words,  $|\psi\rangle$  is a +1 eigenstate of  $G$ . If two operators  $G_1, G_2$  stabilize  $|\psi\rangle$ , then so does their product  $G_1 G_2$ . Thus the stabilizer operators form a finite subgroup of  $\mathcal{P}_N$ .

Below, we will use the stabilizer formalism to evaluate matrix elements as they appear in Eq. (3). Thus the operator string will be applied to computational basis states in the  $Z$  basis. These are states of the form (making the usual identification of Pauli  $Z$  eigenstates  $|\uparrow\rangle \equiv |\sigma = +1\rangle \equiv |0\rangle$  and  $|\downarrow\rangle \equiv |\sigma = -1\rangle \equiv |1\rangle$ )

$$|s\rangle = |\sigma_1\rangle \otimes \cdots \otimes |\sigma_N\rangle, \quad (6)$$

where  $\sigma_i \in \{-1, 1\}$ . This state is stabilized by the collection of operators

$$\sigma_1 Z_1, \sigma_2 Z_2, \dots, \sigma_N Z_N. \quad (7)$$

Any product of these stabilizers is, of course, again a stabilizer. In fact, the full stabilizer group of the state in Eq. (6) contains  $2^N$  elements and is finitely generated by the  $N$  generators in (7). This is an example of a general fact about finite groups: A group  $G$  with  $|G|$  elements is finitely generated by, at most,  $\log_2 |G|$  generators. This is the reason for the efficiency of the stabilizer description. Instead of using  $2^N$  complex amplitudes to describe a state in Hilbert space, we only list the  $\log_2(2^N) = N$  generators of the stabilizer group.

The drawback is, of course, that only a finite number of states in Hilbert space can be described in this way. In fact, as computed in [30], there are roughly  $2^{N(N+1)/2}$  stabilizer states. Nevertheless, it turns out that for the purposes of quantum Monte Carlo simulations of certain Hamiltonians, such as the toy model below, this is all one needs: The operator string applied to  $|s\rangle$  in Eq. (3) never leaves the stabilizer subspace.

Let  $|\psi\rangle$  be a stabilizer state with a stabilizer group  $\langle G_1, G_2, \dots, G_N \rangle$ . If a unitary  $U$  acts on  $|\psi\rangle$ , the new state is described by different stabilizers, in general. They can be worked out as follows:

$$G_i |\psi\rangle = |\psi\rangle \rightarrow (UG_i U^\dagger)U|\psi\rangle = U|\psi\rangle. \quad (8)$$

Thus, if  $G_i$  is a stabilizer of  $|\psi\rangle$ , then  $UG_i U^\dagger$  is a stabilizer of  $U|\psi\rangle$ . In other words, after the action of  $U$  on  $|\psi\rangle$ , the generators get updated as

$$\langle G_1, G_2, \dots, G_N \rangle \rightarrow \langle UG_1 U^\dagger, UG_2 U^\dagger, \dots, UG_N U^\dagger \rangle. \quad (9)$$

Thus, acting with  $U$  on  $|\psi\rangle$  is equivalent to conjugating all the generators by  $U$ .

For general  $U$ , the resulting generators will lie outside the Pauli group and their description will be just as cumbersome as describing  $|\psi\rangle$  in the computational basis. However, if  $U$  maps all Pauli group elements to (possibly different) Pauli group elements under conjugation, then an update according to (9) is straightforward. Since conjugation by  $U$  followed by conjugation by  $V$  is the same as conjugation by  $VU$ , these elements form a group, the *Clifford group*. It was shown in Ref. [31] that the Clifford group is finitely generated by the Hadamard-, phase- and controlled-NOT gates. This set of gates is, of course, not universal. In fact, circuits made up of only these elements are simulable with classical algorithms in polynomial time. Adding  $T$  unitaries to this gate set would make the circuit universal and generally intractable with classical simulations.

### III. A CONTROLLED-NOT TOY MODEL

We illustrate the use of stabilizer circuits by applying the SSE procedure to the Hamiltonian

$$H = \sum_{i=1}^N H_i^{(1)} + \sum_{i=1}^N H_i^{(2)}, \quad (10)$$

$$H_i^{(1)} = -J[\text{CX}]_{i,i+1}, \quad (11)$$

$$H_i^{(2)} = -\frac{\hbar}{2}(X_i + \mathbf{1}). \quad (12)$$

The two sums do not commute with each other. The controlled- $X$  (CX) term is a *controlled*-NOT operation with qubit  $i$  being the control and qubit  $i + 1$  the target. In the  $Z$  basis, this operator flips qubit  $i + 1$ , conditioned on the state of qubit  $i$  being  $|1\rangle$ . The  $H^{(2)}$  term represents an external field acting on each qubit individually. We work throughout this paper with periodic boundary conditions such that site  $N + 1$  is identified with site 1. We also measure energy in units of  $J$ , i.e., we set  $J = 1$ . Conditional interactions, similar to the CX operator, also appear in constrained statistical mechanics systems and lead to interesting dynamical effects. A recent example is the PXP model that describes Rydberg blockade

physics [32]: an atom can only be excited if the neighboring atoms are in the ground state. The system shows interesting many-body dynamics [33] and quantum scarring [34] that can be traced back to the constrained dynamics.

The Hamiltonian in Eq. (10) is physically sensible because the unitary CX is also Hermitian. Thus, in terms of Pauli operators, the Hamiltonian given by Eq. (10) reads

$$H = -\frac{1}{2} \sum_{i=1}^N (\mathbf{1} + Z_i + X_{i+1} - Z_i X_{i+1}) - \frac{h}{2} \sum_{i=1}^N (X_i + \mathbf{1}). \quad (13)$$

We recognize the operator

$$\Pi_i \equiv \frac{\mathbf{1} + X_i}{2} \quad (14)$$

as the projector onto the  $|+\rangle_i = (|0\rangle + |1\rangle)/\sqrt{2}$  state, i.e.,  $\Pi_i \Pi_i = \Pi_i$ .

We will now apply the SSE sampling algorithm to the partition function of the Hamiltonian (10). The sign problem is avoided if we select  $h \geq 0$ . In the course of the algorithm, we have to efficiently evaluate matrix elements of an operator string of the form

$$\mathcal{M} = \langle s | \mathcal{O}_i \mathcal{O}_{i_2} \dots \mathcal{O}_{i_L} | s \rangle, \quad (15)$$

where  $|s\rangle$  is a computational basis state of  $N$  qubits. Each  $\mathcal{O}_i$  is either a  $\Pi$ , a CX, or an identity operator. Consider, as an example, the two-qubit matrix element,

$$\mathcal{M}_0 = \langle 00 | [\text{CX}]_{1,2} \Pi_1 | 00 \rangle. \quad (16)$$

The action of  $\Pi_1$  on the  $|00\rangle$  ket is to put the first qubit into the  $|+\rangle$  state. The effect of  $[\text{CX}]_{1,2}$  is to make a Bell state out of this,

$$[\text{CX}]_{1,2} \Pi_1 | 00 \rangle = \frac{|00\rangle + |11\rangle}{2}. \quad (17)$$

Thus, one obtains the final result,

$$\mathcal{M}_0 = \frac{1}{2}. \quad (18)$$

This simple example already illustrates how the consecutive application of the sequence of operators in Eq. (15) results in superpositions of computational basis states with nonzero entanglement. The fact that the Bell state appears in Eq. (17) also demonstrates that in general, these superpositions *cannot* be rewritten as product states through a local basis change. Nevertheless, the stabilizer formalism allows us to keep track of such states.

If the operators in Eq. (15) were all general, then the effort to keep track of all the complex amplitudes would grow exponentially since the number of amplitudes grows as  $2^N$ . However, as explained above, the action of the Clifford group can be efficiently computed in polynomial time.

In our case, the evaluation has one more complication since the operator string in Eq. (15) contains not only the Clifford operators  $[\text{CX}]_{i,i+1}$ , but also the non-Clifford projection operators  $\Pi_i$ . Nevertheless, a fast evaluation of  $\mathcal{M}$  in Eq. (15) is possible, as we explain in the following.

## A. Calculation of $\mathcal{M}$

In the algorithm, the matrix element  $\mathcal{M}$  in Eq. (15) is evaluated in two steps. First, the state  $|s'\rangle \equiv \mathcal{O}_{i_1} \dots \mathcal{O}_{i_L} |s\rangle$  is determined in terms of its stabilizer group generators by beginning with  $|s\rangle$  and consecutively applying the  $\mathcal{O}$  operators in the order shown. In the second step, the overlap  $\langle s | s' \rangle$  is calculated.

### 1. Calculation of $|s'\rangle$

For the first step, we keep track of the stabilizer group elements and update them after each operator multiplication. The state  $|s\rangle$  is a computational basis element in the  $Z$  basis. As discussed above, its stabilizer group is  $\langle \sigma_1 Z_1, \dots, \sigma_N Z_N \rangle$  with  $\sigma_i \in \{-1, +1\}$ . When we now apply the  $[\text{CX}]_{i,i+1}$  and  $\Pi_i$  operations, this list of generators will have to be updated. In general, there will be  $N$  stabilizer operators. The  $m$ th stabilizer (where  $1 \leq m \leq N$ ) will have the form

$$G_m \equiv \gamma_m X_1^{P_{m1}} Z_1^{Q_{m1}} \otimes \dots \otimes X_N^{P_{mN}} Z_N^{Q_{mN}} = \gamma_m \prod_{n=1}^N X_n^{P_{mn}} Z_n^{Q_{mn}}, \quad (19)$$

where  $\gamma_m \in \{-1, +1\}$  denotes the signs, and  $P$  and  $Q$  are binary matrices with entries  $P_{mn}, Q_{mn} \in \{0, 1\}$ . By specifying all stabilizers  $G_1, \dots, G_N$ , a state is fixed that is unique up to a global phase (i.e., the generators fix a ray). We identify the normalized stabilizer state with the stabilizer group by writing  $|\psi\rangle = \langle G_1, \dots, G_N \rangle$ . We note that this description of the state  $|\psi\rangle$  only requires the storage of  $(2N + 1) \times N$  bits, instead of the usual  $2^N$  complex numbers.

Consider, first, the action of the operator  $[\text{CX}]_{i,i+1}$  on the generators  $G_1, \dots, G_N$ . Since CX is a unitary operation, we merely have to update the  $G_i$  by conjugation according to the rule in (8). Moreover, conjugation of  $X_j$  and  $Z_j$  by  $[\text{CX}]_{i,i+1}$  leaves the former invariant when  $j \neq i$  and  $j \neq i + 1$ . The only nontrivial updates upon conjugation involve  $j = i$  and  $i + 1$  (see [28]):

$$X_i \rightarrow X_i X_{i+1}, \quad (20)$$

$$X_{i+1} \rightarrow X_{i+1}, \quad (21)$$

$$Z_i \rightarrow Z_i, \quad (22)$$

$$Z_{i+1} \rightarrow Z_i Z_{i+1}. \quad (23)$$

Since there are no sign changes, this update does not modify the  $\gamma$  vector. The  $P$  and  $Q$  matrices are updated by

$$P'_{ni+1} = P_{ni} \oplus P_{ni+1}, \quad (24)$$

$$Q'_{ni} = Q_{ni} \oplus Q_{ni+1}, \quad (25)$$

for all  $1 \leq n \leq N$ . Here,  $\oplus$  denotes the *exclusive-or* bit operation that can also be interpreted as addition modulo 2.

We next turn to the  $\Pi_i$  projection operator. Its nonunitarity implies that it can potentially change the normalization of  $|\psi\rangle$ . We account for this by keeping track of a factor  $F$  that we

update as follows:

$$|\psi\rangle = \langle G_1, \dots, G_N \rangle \rightarrow |\psi'\rangle = \langle G'_1, \dots, G'_N \rangle, \quad (26)$$

$$F \rightarrow F', \quad (27)$$

such that

$$\Pi_i F |\psi\rangle = F' |\psi'\rangle. \quad (28)$$

Thus we let  $|\psi\rangle$  and  $|\psi'\rangle$  be states normalized to 1 and let  $F$  and  $F'$  keep track of the actual normalization. This step is necessary because the stabilizers do not determine the norm of a state.

After projector  $\Pi_i$  acts on state  $|\psi\rangle$ , the new stabilizers can be worked out as follows. Since the generators are products of Pauli operators, the  $X_i$  operator will either commute or anticommute with them. Retain all the generators that commute with  $X_i$ . If *all* generators commute with  $X_i$ , we are done and set  $F' = F$ .

Otherwise, let  $G_{m_1}, \dots, G_{m_l}$  be all the generators that anticommute with  $X_i$ . If  $l > 1$ , first replace  $G_{m_2}, \dots, G_{m_l}$  by

$$G'_{m_a} = G_{m_1} G_{m_a} \quad (29)$$

for all  $2 \leq a \leq l$ . Now replace  $G_{m_1}$  by  $X_i$ , i.e.,

$$G'_{m_1} = X_i. \quad (30)$$

All generators in this new collection commute with  $X_i$ . We update the factor to  $F' = F/\sqrt{2}$ .

In terms of the  $\gamma$  vector and  $P, Q$  matrices, Eq. (29) becomes

$$\gamma'_{m_a} = (-1)^\rho \gamma_{m_1} \gamma_{m_a}, \quad (31)$$

$$P'_{m_a s} = P_{m_1 s} \oplus P_{m_a s}, \quad (32)$$

$$Q'_{m_a s} = Q_{m_1 s} \oplus Q_{m_a s}, \quad (33)$$

$$\rho = \sum_s P_{m_a s} \oplus Q_{m_1 s}, \quad (34)$$

while the consequence of Eq. (30) is

$$\gamma'_{m_1} = 1, \quad (35)$$

$$P'_{m_1 s} = \delta_{si}, \quad (36)$$

$$Q'_{m_1 s} = 0. \quad (37)$$

The sign  $(-1)^\rho$  stems from moving the  $X$  operators in  $G_{m_a}$  past the  $Z$  operators in  $G_{m_1}$ . All other components of  $\gamma$ ,  $P$ , and  $Q$  are left unmodified.

Finally, to detect which generators  $G_m$  anticommute with  $X_i$ , we simply check if a generator contains the factor  $Z_i$ . Thus, if  $Q_{mi} = 1$ , then  $G_m$  anticommutes with  $X_i$ , else it commutes. These tools furnish us with a way to complete the first step, i.e., to compute  $|s'\rangle = O_{i_1} \dots O_{i_l} |s\rangle$  by finding its stabilizer group generators. In the source code, the derived equations are implemented in the function APPLY, which modifies the stabilizers.

## 2. Calculation of overlap between $|s'\rangle$ and $|s\rangle$

We first observe that  $\langle s|s'\rangle$  is non-negative. The reason for this is that both  $\Pi_i$  and  $[CX]_{i,i+1}$  are non-negative matrices in

the  $Z$  basis, i.e., all the entries are non-negative. The product of non-negative matrices is non-negative in that same basis, thus  $\langle s|s'\rangle$  is non-negative. This is necessary below in order to avoid the sign problem.

If  $|s\rangle$  has a generator  $g$  and  $|s'\rangle$  has a generator  $-g$ , then  $\langle s|s'\rangle = 0$  since

$$\langle s|s'\rangle = \langle s|g|s'\rangle = -\langle s|s'\rangle. \quad (38)$$

Next, we calculate the overlap in cases where it does not vanish. If we had to convert  $|s'\rangle$  back into the computational basis representation, we would lose all the efficiency gained from the stabilizer formalism. A much more efficient way to compute such overlaps was found in [30]. In our case, the state  $|s\rangle$  has the stabilizer group  $\langle \sigma_1 Z_1, \dots, \sigma_N Z_N \rangle$ . Let the generators of  $|s'\rangle$  be  $\langle g_1, \dots, g_N \rangle$  with factor  $F$ . By operator multiplication, we may be able to make some of the generators of the two states identical. Let the maximum number of such equal generators be  $M$ ; then the overlap is

$$\langle s|s'\rangle = \frac{1}{2^{(F+N-M)/2}}. \quad (39)$$

This finishes the calculation of the matrix element  $\mathcal{M}$ . The reason why this formula differs from [30] is that we apply not only unitaries, but also projectors.

## B. Operator string updating

To sample from the partition function given by Eq. (3), a Markov chain is constructed with each element of the chain being a state  $|s\rangle$  together with the list of  $L$  operators  $[H_{i_1}, H_{i_2}, \dots, H_{i_L}]$ . The goal of this is to generate a chain of elements that appear with frequencies proportional to their weights in Eq. (3). Once the Markov chain reaches equilibrium, it is straightforward to compute observable averages from it; see the discussion in Sec. III C. Our algorithm alternates between proposing a change to a different state  $|s'\rangle$ , chosen at random, and  $L$  consecutive proposals to modify an element of the operator string. It is clear that any combination of state and operator string can, in principle, be reached by a sequence of such steps, and thus the sampling is ergodic.

The state of the simulation, i.e., the position in the Markov chain, is fully characterized by giving the state  $|s\rangle$  and the operator string  $[\mathcal{H}_{i_1} \dots \mathcal{H}_{i_L}]$ . We denote this configuration by  $\mathcal{C}$ . In order for the Markov chain to reach the correct equilibrium distribution, the processes described above have to occur with the correct transition rates. This is achieved by imposing the condition of detailed balance that relates the transition rate between neighboring configurations  $\mathcal{C}$  and  $\mathcal{C}'$  by

$$W(\mathcal{C})P(\mathcal{C} \rightarrow \mathcal{C}') = W(\mathcal{C}')P(\mathcal{C}' \rightarrow \mathcal{C}). \quad (40)$$

The update rule for states is straightforward:

*State updates.* A computational basis  $|s'\rangle$  is chosen uniformly at random and a switch to this state is made with probability

$$\begin{aligned} P(|s\rangle \rightarrow |s'\rangle) &= \min \left( 1, \frac{W(|s'\rangle, [H_{i_1}, \dots, H_{i_L}])}{W(|s\rangle, [H_{i_1}, \dots, H_{i_L}])} \right) \\ &= \min \left( 1, \frac{\langle s'|H_{i_1} \dots H_{i_L}|s'\rangle}{\langle s|H_{i_1} \dots H_{i_L}|s\rangle} \right). \end{aligned} \quad (41)$$

*Operator updates.* We loop over all elements in the operator list. If the  $i$ th operator is the identity  $\mathbf{1}_i$ , then a proposal is made to change it into either  $\Pi_a$  or  $[\text{CX}]_{a,a+1}$ . Here,  $a$  is a site index that is randomly chosen. The decision whether to insert  $\Pi$  or CX is made randomly. With probability

$$P_{\Pi} = \frac{h}{h+J}, \quad (42)$$

the proposal is made to insert a  $\Pi$  operator and, with complementary probability

$$P_{\text{CX}} = 1 - P_{\Pi} = \frac{J}{h+J}, \quad (43)$$

a CX operator insertion is proposed (we temporarily restore the coupling constant  $J$  for clarity).

Each of these proposals is now accepted according to the probabilities

$$P(\mathbf{1} \rightarrow \mathcal{O}) = \min\left(1, \frac{N\beta(h+J)W_{\mathcal{O}}}{L-n} \frac{W_{\mathbf{1}}}{W_{\mathcal{O}}}\right), \quad (44)$$

$$\mathcal{O} = \Pi_a \text{ or } [\text{CX}]_{a,a+1}, \quad (45)$$

$$W_{\mathcal{O}} = \langle s | H_{i_1} \dots \mathcal{O} \dots H_{i_L} | s \rangle, \quad (46)$$

$$W_{\mathbf{1}} = \langle s | H_{i_1} \dots \mathbf{1}_i \dots H_{i_L} | s \rangle. \quad (47)$$

If instead the  $i$ th operator is not the identity, then the operator is replaced by the identity  $\mathbf{1}_i$  with probability

$$P(\mathcal{O} \rightarrow \mathbf{1}) = \min\left(1, \frac{L-n+1}{N\beta(h+J)} \frac{W_{\mathbf{1}}}{W_{\mathcal{O}}}\right). \quad (48)$$

In all these expressions, the variable  $n$  denotes the number of nonidentity operators *before* the transition is made. The expressions themselves follow from Eq. (3), together with the detailed balance condition given by Eq. (40). The factors of  $N$  in Eqs. (44) and (48) are due to the fact that the insertion of operators can happen at one of the  $N$  sites, while the removal of an operator is tied to a specific site. In the ratio of weights in Eq. (3), factors of  $h$  or  $J$  appear. But since we already preselected by the probabilities in Eqs. (42) and (43), now the factor  $h+J$  must appear in Eqs. (44) and (48) to compensate for the denominators in Eqs. (42) and (43).

### C. Simulation results

We can gauge the quality of the simulation by computing the thermal average of the energy as a function of the temperature  $T$ . We compare the simulations with exact diagonalization results. The mean thermal energy can be obtained from the partition function [14],

$$\langle H \rangle = -\partial_{\beta} \ln Z(\beta) = -\frac{\langle n \rangle}{\beta}, \quad (49)$$

where Eq. (3) was used in the last step. In other words, by calculating the average number of nonidentity operators encountered in our random Markov process, we obtain the mean of the energy.

As discussed above, the stochastic series expansion has a cutoff parameter  $L$  that denotes the order of the expansion of the exponential  $\exp(-\beta H)$ . When  $\langle n \rangle$  is computed during the

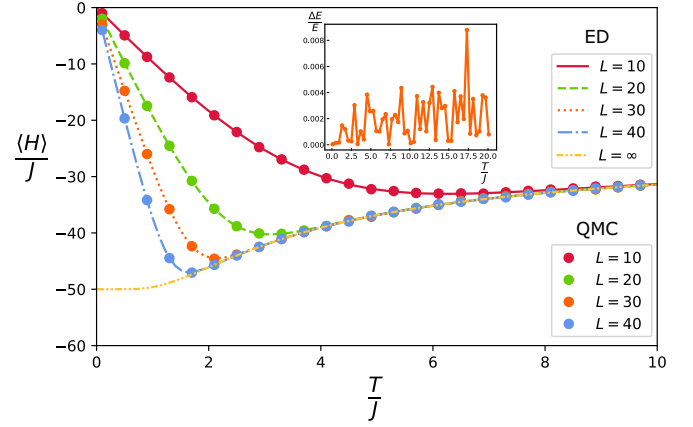


FIG. 2. Thermal average energy of the system in Eq. (10) with  $N = 10$  qubits and external field  $h/J = 4.0$  for various values of the expansion order  $L$ . The continuous lines are exact results obtained by calculating the mean energy with the truncated partition function in Eq. (50) using exact diagonalization. The points are the results of the quantum Monte Carlo simulation. The relative error at each data point is always less than 1%; an example for  $L = 30$  is shown in the inset.

simulation and we find that  $\langle n \rangle \approx L$ , this is an indication that the chosen  $L$  is not sufficiently large and must be increased (for a pedagogical discussion of this, see Sec. 5.2.3. of [9]). Instead of comparing the simulation to the infinite  $L$  result, we define a *truncated* partition function,

$$Z_L(\beta) \equiv \sum_{n=0}^L \frac{(-\beta)^n}{n!} \text{Tr}[H^n]. \quad (50)$$

In this way, we separate the discussion of sampling errors from the issue of choosing too small an  $L$ . If we denote by  $e_0$  the ground-state energy per qubit, it is clear from Eq. (49) that as  $T \rightarrow 0$ , the expansion order has to be chosen as  $L \gtrsim e_0 N/T$ .

As a check of our simulation, we choose a system of  $N = 10$  and set the external field first to a value of  $h/J = 4.0$ . The quantum Monte Carlo results are obtained after  $0.5 \times 10^5$  thermalization cycles followed by  $0.5 \times 10^5$  cycles to compute the average  $\langle n \rangle$ . Each cycle consists of an attempt to change the state  $|s\rangle$  followed by  $L$  proposals to update the operator string. The simulation proceeds by successively cooling down from a temperature of  $T/J = 10$  to 0 in steps of 0.4. The calculation is repeated for several expansion orders  $L = 10, 20, 30, 40$ .

We also performed an exact diagonalization calculation using the truncated partition function of Eq. (50) and the full partition function ( $L = \infty$ ); see the Appendix A for details. The data are shown in Fig. 2. The Monte Carlo results are in good agreement with the exact results, the relative error being smaller than 1%; see inset. The author's C++ implementation of the Monte Carlo algorithm and the PYTHON implementation of the exact diagonalization can be found in the GitHub repository [23].

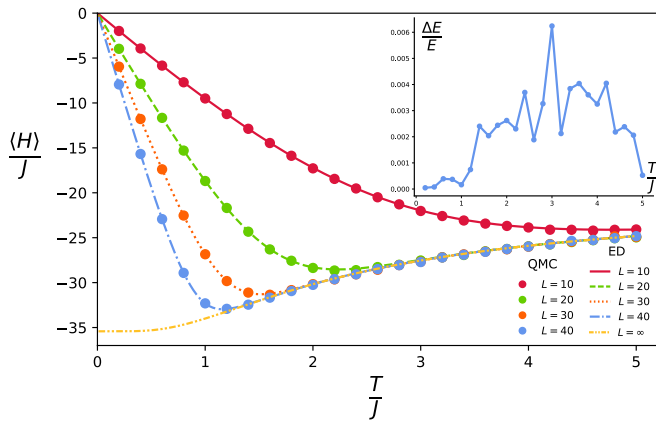


FIG. 3. Thermal average energy of the transverse-field Ising model, defined in Eq. (52). We simulate the system for  $N = 10$  qubits and an external field of  $h/J = 3.0$  for various values of the expansion order  $L$ . The continuous lines are exact results obtained by calculating the mean energy with the truncated partition function in Eq. (50) using exact diagonalization. The points are the results of the quantum Monte Carlo simulation. The relative error at each data point is always less than 1%; an example for  $L = 40$  is shown in the inset.

#### IV. STABILIZER UPDATES FOR THE TRANSVERSE-FIELD ISING MODEL AND $\mathbb{Z}_2$ GAUGE THEORIES

The stabilizer scheme that we have described is not limited to the toy model above. Instead, a large number of spin-1/2 models can be reformulated in this way. As another example, we apply the technique to the transverse-field Ising model defined by

$$H = -J \sum_{\langle ij \rangle} \frac{Z_i Z_j + \mathbf{1}}{2} - h \sum_i \frac{X_i + \mathbf{1}}{2} \quad (51)$$

$$= -J \sum_{\langle ij \rangle} \tilde{\Pi}_{ij} - h \sum_i \Pi_i. \quad (52)$$

We have added constants to the operators to turn them into projectors. The second term is the  $\Pi_i$  operator that we have already encountered, while the first term is a new kind of projector that we denote by  $\tilde{\Pi}_{ij}$ . Above, we have given the update rules when  $\Pi_i$  acts. When  $\tilde{\Pi}_{ij}$  acts, the rules are nearly as simple. First, retain all the generators that commute with  $Z_i Z_j$ . Then list all operators that anticommute with  $Z_i Z_j$ , replace the first of these by  $Z_i Z_j$ , and make new generators by taking products as before. Since  $\Pi_i$  and  $\tilde{\Pi}_{ij}$  are projection operators, we need to keep track of the overall factor  $F$  to account for potential changes of the norm of a state. The update rule is the same as before.

Since  $\Pi_i$  and  $\tilde{\Pi}_{ij}$  have all non-negative entries in the  $Z$  basis, the matrix elements  $\mathcal{M}$  are all non-negative for  $h > 0$ , thereby avoiding the sign problem. The author's stabilizer Monte Carlo code for the transverse-field Ising model is available in the GitHub repository [23], together with the corresponding exact diagonalization code that was used to compare the numerics. As an example, we show in Fig. 3 the average thermal energy of the system as a function of temperature. To compare with exact diagonalization, we have

used  $N = 10$  Ising spins. The system is again thermalized for  $0.5 \times 10^5$  cycles and measurements are taken in the next  $0.5 \times 10^5$  cycles. The Monte Carlo data (dots) agree well with the exact diagonalization results, the relative error being less than 1%, as shown in the inset of Fig. 3.

Finally, we note that an extension of the presented approach to  $\mathbb{Z}_2$  lattice gauge theories is straightforward. In such models, it is usual to have products of multiple spin operators appearing in the Hamiltonian. An example is Kitaev's toric-code Hamiltonian [35], which contains products of four spin-1/2 operators as star and plaquette terms. Such combinations of operators can be rewritten in terms of projection operators  $\Pi_1 \equiv (\mathbf{1} + X_1 X_2 X_3 X_4)/2$  and  $\Pi_2 \equiv (\mathbf{1} + Z_1 Z_2 Z_3 Z_4)/2$  that are non-negative in the  $Z$  basis. Their update rules are similar to the ones for  $\tilde{\Pi}_{ij}$ . In this way, the thermodynamics of certain  $\mathbb{Z}_2$  lattice gauge theories will be simulable without sign problems.

#### V. CONCLUSION

To summarize, we have demonstrated a scheme to remove the no-branching condition in the SSE quantum Monte Carlo algorithm by utilizing Clifford stabilizer states. In the language of word-line trajectories, our algorithm evaluates multiple branches of the wave function simultaneously, instead of averaging over trajectories one by one. This allowed us to straightforwardly deal with the CNOT toy model. Next, it was shown that the simulation of the transverse-field Ising model is similarly effortless, requiring only minor modifications of the technique. Finally, we indicated how the approach may also be leveraged in the simulation of  $\mathbb{Z}_2$  gauge theories that usually involve terms of multispin operators. The present paper focused on thermal averages, but the SSE technique can also be adapted to compute ground-state properties [9]. The stabilizer approach illustrated here readily extends to these calculations.

There are several interesting directions to pursue. The simulation [23] of Clifford circuits in this paper, although requiring only polynomial time and space complexity, was performed in the simplest possible way. Therefore, it seems likely that more sophisticated approaches, such as the tableau-based simulation of [30] or the graph-state simulation of [36], will provide significant speedups.

In terms of applications, it is worth mentioning that lattice gauge theories, once the domain of high-energy physics [37], have recently garnered increasing attention from the cold-atoms community as candidates to investigate quantum simulators [38–41]. It is worth exploring whether stabilizer states could unlock the SSE technique for these models, especially in cases where an efficient updating scheme is difficult to design or otherwise unavailable. Although this approach does not mitigate the sign problem generally, the class of simulable models might nevertheless be expanded. In this way, the stabilizer-SSE algorithm may provide a generic tool to simulate models and make it unnecessary to tailor update algorithms specifically to fit a particular model.

Finally, we note that Hamiltonians of interacting qubits with controlled-X and controlled-Z operations, somewhat resembling Eq. (10), have been explored in the literature as

models for gapped phases of matter [42,43]. This is a further potential playground for the stabilizer-SSE approach.

### ACKNOWLEDGMENT

It is a pleasure to thank Pok Man Tam and Pak Kau Lim for an interesting afternoon of discussion. I thank Abiramjee Jeevanesan for helpful technical advice.

### APPENDIX: DETAILS ON THE EXACT DIAGONALIZATION CALCULATION

In the main text, we gauged the quality of the Monte Carlo algorithm by comparing the average thermal energy to the exact diagonalization (ED) results. The ED implementation is simple and can be found in the author's GitHub repository [23]. The code uses NUMPY [44] to construct the Hamiltonian matrix. For  $N$  qubits, the latter has dimension  $2^N \times 2^N$ . The implementation starts by constructing the  $CX_i$  and  $\Pi_i$  operators for each site  $i$  via Kronecker products of Pauli matrices and identity operators. This yields the Hamiltonian  $H$  in matrix form and eigenvalues  $E_i$  are straightforwardly computed. As explained in the main text, we construct the mean energy.

$$\langle H \rangle = -\partial_\beta \ln Z(\beta) \quad (\text{A1})$$

from the partition function  $Z = \text{Tr}[\exp(-\beta H)]$ . We use the symbolic manipulation package SYMPY [45] to compute the

partition function,

$$Z(\beta) = \sum_{i=1}^{2^N} e^{-\beta E_i}, \quad (\text{A2})$$

using the known eigenvalues  $E_i$ . From this, we compute the average energy  $\langle H \rangle = -\partial_\beta \ln Z(\beta)$  by symbolical differentiation with respect to  $\beta$  and thereby avoid ill-conditioned finite-difference calculations.

Similarly, we are also interested in the truncated partition function,

$$Z_L(\beta) \equiv \sum_{n=0}^L \frac{(-\beta)^n}{n!} \text{Tr}[H^n]. \quad (\text{A3})$$

We tabulate the values of  $\text{Tr}[H^n]$  and use SYMPY to construct the polynomials  $Z_L(\beta)$  that we again differentiate, according to  $\langle H \rangle_L = -\partial_\beta \ln Z_L(\beta)$ , to obtain the mean energy. Figure 2 in the main text shows the resulting curves for  $L = 10, 20, 30, 40$  and infinite order.

Our Hamiltonian matrix has dimension  $D = 2^N$ . Since a general  $D \times D$  matrix requires  $O(D^3)$  steps for the diagonalization and there are  $2^N$  basis states, the computational complexity of the full diagonalization increases exponentially as  $O(8^N)$ .

- 
- [1] J. E. Hirsch, R. L. Sugar, D. J. Scalapino, and R. Blankenbecler, Monte Carlo simulations of one-dimensional fermion systems, *Phys. Rev. B* **26**, 5033 (1982).
  - [2] B. B. Beard and U.-J. Wiese, Simulations of discrete quantum systems in continuous euclidean time, *Phys. Rev. Lett.* **77**, 5130 (1996).
  - [3] N. V. Prokof'ev, B. V. Svistunov, and I. S. Tupitsyn, Exact, complete, and universal continuous-time worldline Monte Carlo approach to the statistics of discrete quantum systems, *J. Expt. Theor. Phys.* **87**, 310 (1998).
  - [4] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, Equation of state calculations by fast computing machines, *J. Chem. Phys.* **21**, 1087 (1953).
  - [5] Z.-X. Li and H. Yao, Sign-problem-free fermionic quantum Monte Carlo: Developments and applications, *Annu. Rev. Condens. Matter Phys.* **10**, 337 (2019).
  - [6] P. Henelius and A. W. Sandvik, Sign problem in Monte Carlo simulations of frustrated quantum spin systems, *Phys. Rev. B* **62**, 1102 (2000).
  - [7] S. Lloyd, Universal quantum simulators, *Science* **273**, 1073 (1996).
  - [8] R. P. Feynman, Simulating physics with computers, in *Feynman and Computation* (CRC Press, Boca Raton, FL, 2018), pp. 133–153.
  - [9] A. W. Sandvik, Computational studies of quantum spin systems, in *AIP Conference Proceedings* (American Institute of Physics, Melville, New York, 2010), Vol. 1297, pp. 135–338.
  - [10] A. W. Sandvik and J. Kurkijärvi, Quantum Monte Carlo simulation method for spin systems, *Phys. Rev. B* **43**, 5950 (1991).
  - [11] A. W. Sandvik, Stochastic series expansion method with operator-loop update, *Phys. Rev. B* **59**, R14157 (1999).
  - [12] R. K. Kaul, R. G. Melko, and A. W. Sandvik, Bridging lattice-scale physics and continuum field theory with quantum Monte Carlo simulations, *Annu. Rev. Condens. Matter Phys.* **4**, 179 (2013).
  - [13] R. H. Swendsen and J.-S. Wang, Nonuniversal critical dynamics in Monte Carlo simulations, *Phys. Rev. Lett.* **58**, 96 (1987).
  - [14] A. W. Sandvik, Stochastic series expansion method for quantum Ising models with arbitrary interactions, *Phys. Rev. E* **68**, 056701 (2003).
  - [15] O. F. Syljuåsen and A. W. Sandvik, Quantum Monte Carlo with directed loops, *Phys. Rev. E* **66**, 046701 (2002).
  - [16] K. C. Tan, D. Bhowmick, and P. Sengupta, Sign-problem free quantum stochastic series expansion algorithm on a quantum computer, *npj Quantum Inf.* **8**, 44 (2022).
  - [17] T. Jiang, J. Zhang, M. K. A. Baumgarten, M.-F. Chen, H. Q. Dinh, A. Ganeshram, N. Maskara, A. Ni, and J. Lee, Walking through Hilbert space with quantum computers, [arXiv:2407.11672](https://arxiv.org/abs/2407.11672).
  - [18] P. W. Shor, Scheme for reducing decoherence in quantum computer memory, *Phys. Rev. A* **52**, R2493 (1995).
  - [19] A. R. Calderbank and P. W. Shor, Good quantum error-correcting codes exist, *Phys. Rev. A* **54**, 1098 (1996).
  - [20] A. M. Steane, Error correcting codes in quantum theory, *Phys. Rev. Lett.* **77**, 793 (1996).
  - [21] D. Gottesman, *Stabilizer Codes and Quantum Error Correction* (California Institute of Technology, Pasadena California, 1997).

- [22] D. Fattal, T. S. Cubitt, Y. Yamamoto, S. Bravyi, and I. L. Chuang, Entanglement in the stabilizer formalism, [arXiv:quant-ph/0406168](https://arxiv.org/abs/quant-ph/0406168).
- [23] B. Jeevanesan, SSE stabilizer Monte Carlo, 2025, <https://github.com/BillyJay/SSE-Stabilizer-Monte-Carlo>.
- [24] D. C. Handscomb, A Monte Carlo method applied to the Heisenberg ferromagnet, in *Mathematical Proceedings of the Cambridge Philosophical Society* (Cambridge University Press, Cambridge, 1964), Vol. 60, pp. 115–122.
- [25] D. C. Handscomb, The Monte Carlo method in quantum statistical mechanics, in *Mathematical Proceedings of the Cambridge Philosophical Society* (Cambridge University Press, Cambridge, 1962), Vol. 58, pp. 594–598.
- [26] M. Suzuki, Relationship between  $d$ -dimensional quantum spin systems and  $(d + 1)$ -dimensional Ising systems: Equivalence, critical exponents and systematic approximants of the partition function and spin correlations, *Prog. Theor. Phys.* **56**, 1454 (1976).
- [27] P. Diaconis, The Markov chain Monte Carlo revolution, *Bull. Am. Math. Soc.* **46**, 179 (2009).
- [28] D. Gottesman, The Heisenberg representation of quantum computers, [arXiv:quant-ph/9807006](https://arxiv.org/abs/quant-ph/9807006).
- [29] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, 2010).
- [30] S. Aaronson and D. Gottesman, Improved simulation of stabilizer circuits, *Phys. Rev. A* **70**, 052328 (2004).
- [31] D. Gottesman, Theory of fault-tolerant quantum computation, *Phys. Rev. A* **57**, 127 (1998).
- [32] I. Lesanovsky and H. Katsura, Interacting Fibonacci anyons in a Rydberg gas, *Phys. Rev. A* **86**, 041601 (2012).
- [33] H. Bernien, S. Schwartz, A. Keesling, H. Levine, A. Omran, H. Pichler, S. Choi, A. S. Zibrov, M. Endres, M. Greiner *et al.*, Probing many-body dynamics on a 51-atom quantum simulator, *Nature (London)* **551**, 579 (2017).
- [34] M. Serbyn, D. A. Abanin, and Z. Papić, Quantum many-body scars and weak breaking of ergodicity, *Nat. Phys.* **17**, 675 (2021).
- [35] A. Yu. Kitaev, Fault-tolerant quantum computation by anyons, *Ann. Phys.* **303**, 2 (2003).
- [36] S. Anders and H. J. Briegel, Fast simulation of stabilizer circuits using a graph-state representation, *Phys. Rev. A* **73**, 022334 (2006).
- [37] M. Creutz, *Quarks, Gluons and Lattices* (Cambridge University Press, Cambridge, 1983).
- [38] U.-J. Wiese, Ultracold quantum gases and lattice systems: Quantum simulation of lattice gauge theories, *Ann. Phys.* **525**, 777 (2013).
- [39] C. W. Bauer, Z. Davoudi, A. B. Balantekin, T. Bhattacharya, M. Carena, W. A. De Jong, P. Draper, A. El-Khadra, N. Gemelke, M. Hanada *et al.*, Quantum simulation for high-energy physics, *PRX Quantum* **4**, 027001 (2023).
- [40] M. Aidelsburger, L. Barbiero, A. Bermudez, T. Chanda, A. Dauphin, D. González-Cuadra, P. R. Grzybowski, S. Hands, F. Jendrzejewski, J. Jünemann *et al.*, Cold atoms meet lattice gauge theory, *Phil. Trans. R. Soc. A* **380**, 20210064 (2022).
- [41] M. C. Banuls, R. Blatt, J. Catani, A. Celi, J. I. Cirac, M. Dalmonte, L. Fallani, K. Jansen, M. Lewenstein, S. Montangero *et al.*, Simulating lattice gauge theories within quantum technologies, *Eur. Phys. J. D* **74**, 165 (2020).
- [42] X. Chen, Z.-X. Liu, and X.-G. Wen, Two-dimensional symmetry-protected topological orders and their protected gapless edge excitations, *Phys. Rev. B* **84**, 235141 (2011).
- [43] D. T. Stephen and X. Chen, Fusion of one-dimensional gapped phases and their domain walls, *Phys. Rev. B* **110**, 165144 (2024).
- [44] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant *et al.*, Array programming with NumPy, *Nature (London)* **585**, 357 (2020).
- [45] A. Meurer, C. P. Smith, M. Paprocki, O. Čertík, S. B. Kirpichev, M. Rocklin, A. Kumar, S. Ivanov, J. K. Moore, S. Singh *et al.*, Sympy: Symbolic computing in Python, *PeerJ Comput. Sci.* **3**, e103 (2017).