Quantum Monte Carlo algorithm for arbitrary spin-1/2 Hamiltonians

Lev Barash [1,*] Arman Babakhani [1,2] and Itay Hen [1,2]

[1]*Information Sciences Institute, University of Southern California, Marina del Rey, California 90292, USA*
[2]*Department of Physics and Astronomy, University of Southern California, Los Angeles, California 90089, USA*

We present a universal parameter-free quantum Monte Carlo (QMC) algorithm designed to simulate arbitrary spin-1/2 Hamiltonians. To ensure the convergence of the Markov chain to equilibrium for every conceivable case, we devise a clear and simple automated protocol that produces QMC updates that are provably ergodic and satisfy detailed balance. We demonstrate the applicability and versatility of our method by considering several illustrative examples, including the simulation of the XY model on a triangular lattice, the toric code, and random *k*-local Hamiltonians.

I. INTRODUCTION

Quantum Monte Carlo (QMC) algorithms [1–3] are an indispensable tool in the study of the equilibrium properties of large quantum many-body systems, with applications ranging from superconductivity and novel quantum materials [4–6] through the physics of neutron stars [7] and quantum chromodynamics [8,9]. The algorithmic development of QMC remains an active area of research, with continual efforts being made to extend the scope of QMC applicability and to improve convergence rates of existing algorithms with the goal of expanding our understanding of quantum systems and facilitating the discovery of novel quantum phenomena [10–14].

While QMC algorithms have been adapted to the simulation of a wide variety of physical systems, different physical models typically require the development of distinct model-specific update rules and measurement schemes, making the simulation of many large-scale quantum many-body systems prohibitively time consuming.

In this paper, we introduce a universal parameter-free QMC algorithm designed to reliably simulate *arbitrary* spin-1/2 Hamiltonians. To achieve such capabilities, we devise a clear and simple automated protocol for generating the necessary set of updates to ensure the ergodic Markov chain Monte Carlo sampling of any conceivable input system. The generated QMC updates are shown to be ergodic as well as satisfying detailed balance, thereby guaranteeing the proper convergence of the QMC Markov chain. We demonstrate the validity and flexibility of our code by studying in detail a number of models, including the XY model on a triangular lattice,

the toric code, random *k*-local Hamiltonians, and more. We have also made our program code freely accessible on GitHub [15].

The technique we propose here builds on the recently introduced permutation matrix representation (PMR) QMC [16]—an abstract Trotter-error-free technique wherein the quantum dimension consists of products of elements of permutation groups and which as a result allows for the general treatment of entire classes of Hamiltonians. In PMR QMC, the partition function is expanded in a power series in the off-diagonal strength of the Hamiltonian about the partition function of the classical (diagonal) component of the Hamiltonian [17,18].

The paper is organized as follows. In Sec. II, we provide a brief overview of the off-diagonal series expansion for quantum partition functions, on which our approach is founded. In Sec. III, we analyze spin-1/2 Hamiltonians in the context of PMR QMC. In the following section, Sec. IV, we discuss our method to generate all the necessary QMC updates, proving that these ensure both ergodicity and detailed balance. Section V is devoted to the implementation of comprehensive measurement schemes. In Sec. VI we showcase the power of our technique by detailing the simulation results of the physical models mentioned above. We conclude in Sec. VII with an additional discussion of future directions of research.

II. THE OFF-DIAGONAL PARTITION FUNCTION EXPANSION

The PMR protocol [16] begins by first casting the to-be-simulated Hamiltonian in PMR form, i.e., as a sum:

$$H = \sum_{j=0}^{M} \tilde{P}_j = \sum_{j=0}^{M} D_j P_j = D_0 + \sum_{j=1}^{M} D_j P_j, \quad (1)$$

where $\{\tilde{P}_j\}$ is a set of $M+1$ distinct generalized permutation matrices [19], i.e., matrices with at most one nonzero element in each row and each column. Each operator $\tilde{P}_j$ can be written, without loss of generality, as $\tilde{P}_j = D_j P_j$, where D_j is a

---

*lbarash@usc.edu

diagonal matrix and P_j is a permutation matrix with no fixed points (equivalently, no nonzero diagonal elements) except for the identity matrix $P_0 = \mathbb{1}$. We will refer to the basis in which the operators $\{D_j\}$ are diagonal as the computational basis and denote its states by $\{|z\rangle\}$. We will call the diagonal matrix D_0 the "classical Hamiltonian." The permutation matrices appearing in H will be treated as a subset of a permutation group, wherein P_0 is the identity element.

The $\{D_j P_j\}$ off-diagonal operators (in the computational basis) give the system its "quantum dimension." For $j > 0$, each term $D_j P_j$ obeys $D_j P_j |z\rangle = d_j(z')|z'\rangle$, where $d_j(z')$ is a possibly complex-valued coefficient and $|z'\rangle \neq |z\rangle$ is a basis state.

Upon casting the Hamiltonian in PMR form, one can show [16] that the partition function $Z = \text{Tr}\,[e^{-\beta H}]$ can be written as

$$Z = \sum_z \sum_{q=0}^{\infty} \sum_{S_{\mathbf{i}_q}} D_{(z,S_{\mathbf{i}_q})} e^{-\beta[E_{z_0},...,E_{z_q}]} \langle z|S_{\mathbf{i}_q}|z\rangle. \quad (2)$$

The sum above is a double sum: over the set of all basis states z and over all products of q off-diagonal permutation operators $S_{\mathbf{i}_q} = P_{i_q} \cdots P_{i_2} P_{i_1}$ with q running from zero to infinity. The multi-index $\mathbf{i}_q = (i_1, \ldots, i_q)$ covers all products of permutation operators, where each index i_j runs from 1 to M.

In the above sum, each summand is a product of three terms. The first is $D_{(z,S_{\mathbf{i}_q})} \equiv \prod_{j=1}^{q} d_{z_j}^{(i_j)}$ consisting of a product of the matrix elements:

$$d_{z_j}^{(i_j)} = \langle z_j|D_{i_j}|z_j\rangle. \quad (3)$$

The various $\{|z_j\rangle\}_{j=0}^{q}$ states are the states obtained from the action of the ordered P_{i_j} operators in the product $S_{\mathbf{i}_q}$ on $|z_0\rangle$, then on $|z_1\rangle$, and so forth. For $S_{\mathbf{i}_q} = P_{i_q} \cdots P_{i_2} P_{i_1}$, we have $|z_0\rangle = |z\rangle$, $P_{i_j}|z_{j-1}\rangle = |z_j\rangle$ for $j = 1, 2, \ldots, q$. The proper indexing of the states $|z_j\rangle$ is $|z_{(i_1,i_2,\ldots,i_j)}\rangle$ to indicate that the state at the jth step depends on all $P_{i_1} \cdots P_{i_j}$. We will, however, use the shorthand $|z_j\rangle$ to simplify the notation. The sequence of basis states $\{|z_j\rangle\}$ may be viewed as a "walk" [20] on the Hamiltonian graph where every matrix element H_{ij} corresponds to an edge between the two basis states, or nodes, i and j.

The second term in each summand, $e^{-\beta[E_{z_0},...,E_{z_q}]}$, is called the divided differences of the function $f(\cdot) = e^{-\beta(\cdot)}$ with respect to the inputs $[E_{z_0}, \ldots, E_{z_q}]$, where $E_{z_i} = \langle z_i|H|z_i\rangle = \langle z_i|D_0|z_i\rangle$. The divided differences [21,22] of the function f are defined as[1]

$$f[E_{z_0}, \ldots, E_{z_q}] \equiv \sum_{j=0}^{q} \frac{f(E_{z_j})}{\prod_{k \neq j} (E_{z_j} - E_{z_k})}. \quad (4)$$

A useful property is that the divided differences are invariant under rearrangements of the input values.

Now, the term $\langle z|S_{\mathbf{i}_q}|z\rangle$ in Eq. (2) evaluates either to 1 or to zero. Moreover, since the permutation matrices with the exception of P_0 have no fixed points, the condition $\langle z|S_{\mathbf{i}_q}|z\rangle = 1$

implies $S_{\mathbf{i}_q} = \mathbb{1}$; i.e., $S_{\mathbf{i}_q}$ must evaluate to the identity element P_0 (note that the identity element does not appear in the sequences $S_{\mathbf{i}_q}$). The expansion in Eq. (2) can thus be more succinctly rewritten as

$$Z = \sum_z \sum_{S_{\mathbf{i}_q} = \mathbb{1}} D_{(z,S_{\mathbf{i}_q})} e^{-\beta[E_{z_0},...,E_{z_q}]}, \quad (5)$$

i.e., as a sum over all basis states and permutation matrix products that evaluate to the identity matrix.

Having derived the expansion in Eq. (5) for any Hamiltonian cast in the form of Eq. (1), we are now in a position to interpret the partition function expansion as a sum of weights, i.e., $Z = \sum_{\mathcal{C}} w_{\mathcal{C}}$, where the set of configurations $\{\mathcal{C}\}$ consists of all the distinct pairs $\{|z\rangle, S_{\mathbf{i}_q}\}$. We refer to

$$w_{\mathcal{C}} = D_{(z,S_{\mathbf{i}_q})} e^{-\beta[E_{z_0},...,E_{z_q}]} \quad (6)$$

as the configuration weight. We note that, as written, the weights $w_{\mathcal{C}}$ can in general be complex valued, despite the partition function $Z = \sum_{\mathcal{C}} w_{\mathcal{C}}$ being real (and positive). Hence, the imaginary portions of the complex-valued weights do not contribute to the partition function and may be disregarded altogether.[2] We may therefore ignore the imaginary components of the weights to obtain the strictly real-valued weights,

$$W_{\mathcal{C}} = \text{Re}\left[D_{(z,S_{\mathbf{i}_q})}\right] e^{-\beta[E_{z_0},...,E_{z_q}]}. \quad (7)$$

III. PERMUTATION MATRIX REPRESENTATION OF SPIN-1/2 HAMILTONIANS

We next discuss the PMR formulation of general spin-1/2 Hamiltonians, which are the main focus of this work. Hamiltonians of spin-1/2 systems are often, and can always be, represented using Pauli matrices (X, Y, and Z). In order to represent this class of Hamiltonians in PMR form, Eq. (1), we choose our group of permutations to be the set of all tensor products of Pauli-X operators, G_X, while the diagonal matrices D_j will be expressed in terms of tensor products of Pauli-Z matrices. As we will see, any permutation operator for spin-1/2 systems belongs to G_X.

Consider a Hamiltonian H given as a linear combination of Pauli strings $S^{(i)} = \bigotimes_j s_j^{(i)}$. Here, $s_j^{(i)}$ represents a Pauli matrix $s \in \{X, Y, Z\}$ in the ith Pauli string, which acts on the jth spin, where $j \in \{1, \ldots, n\}$ with n being the number of spins in the system (we assume that each Pauli string $S^{(i)}$ contains no more than one Pauli matrix $s_j^{(i)}$ for each spin index j). Explicitly, the Hamiltonian has the form

$$H = \sum_i c_i S^{(i)}, \quad (8)$$

where the c_i's are real-valued coefficients. To cast the Hamiltonian in PMR form, we first write each Pauli string as a product of a diagonal operator and a permutation operator; i.e., we will write $S^{(i)}$ as a string of Pauli Z's multiplied by a

---

[1]The expression, Eq. (4), is ill defined if two (or more) of the inputs are repeated, in which case $f[E_{z_0}, \ldots, E_{z_q}]$ can be properly evaluated using limits.

[2]In fact, it can be shown that for every configuration $\mathcal{C} = \{|z\rangle, P_{i_q} \cdots P_{i_2} P_{i_1}\}$ the configuration with reverse order of operators $\overline{\mathcal{C}} = \{|z\rangle, P_{i_1}^{-1} P_{i_2}^{-1} \cdots P_{i_q}^{-1}\}$ produces the conjugate weight $w_{\overline{\mathcal{C}}} = \overline{w}_{\mathcal{C}}$.

string of Pauli X's using the fact that $Y = -iZX$, giving

$$H = \sum_i \tilde{c}_i Z^{(i)} X^{(i)}, \tag{9}$$

where $\tilde{c}_i = (-i)^{n_y^{(i)}} c_i$ with $n_y^{(i)}$ being the number of Y operators appearing in the ith Pauli string. Here, $X^{(i)}$ represents the ith string (or product) of Pauli-X operators. The notation $Z^{(i)}$ similarly represents a string of Pauli-Z operators.

Noting that Pauli-X strings are permutation operators and that Pauli-Z strings are diagonal in our chosen basis, we next group together all terms that have the same Pauli-X component, ending up with a Hamiltonian of the form

$$H = \sum_i \left(\sum_j \tilde{c}_j Z_j^{(i)} \right) X^{(i)}. \tag{10}$$

The permutation operators $X^{(i)} \in G_X$ in the above expression are now *distinct* products of Pauli-X operators, and we identify $D_i = \sum_j \tilde{c}_j Z_j^{(i)}$ as the accompanying diagonal operators, as desired.

IV. QMC UPDATES FOR ARBITRARY SPIN-1/2 HAMILTONIANS

A. QMC configurations

As was discussed above, for any Hamiltonian cast in PMR form, the partition function $Z = \text{Tr}[e^{-\beta H}]$ can be written as a sum of configuration weights, where a configuration $\mathcal{C} = \{|z\rangle, S_{\mathbf{i}_q}\}$ is a pair of a "classical" basis state $|z\rangle$ (an eigenstate of diagonal operators) and a product $S_{\mathbf{i}_q}$ of permutation operators that must evaluate to the identity element $P_0 = \mathbb{1}$. The configuration $\mathcal{C}$ induces a list of states $\{|z_0\rangle = |z\rangle, |z_1\rangle, \ldots, |z_q\rangle = |z\rangle\}$, which in turn generates a corresponding multiset of energies $E_{\mathcal{C}} = \{E_{z_0}, E_{z_1}, \ldots, E_{z_q}\}$ for the configuration.

We can now consider a QMC algorithm, i.e., a Markov chain Monte Carlo routine, that samples these configurations with probabilities proportional to their weights $W_{\mathcal{C}}$ [Eq. (7)]. The Markov process would start with some initial configuration, and a set of (probabilistic) rules, or QMC updates, will dictate transitions from one configuration to the next.

We will take the initial state of the chain to be $\mathcal{C}_0 = \{|z\rangle, S_0 = \mathbb{1}\}$, where $|z\rangle$ is a randomly generated initial classical state. The weight of this initial configuration is

$$W_{\mathcal{C}_0} = e^{-\beta[E_z]} = e^{-\beta E_z}, \tag{11}$$

i.e., the classical Boltzmann weight of the initial randomly generated basis state $|z\rangle$.

The set of QMC updates will be discussed in the next sections. Before doing so, we note that to make certain that configurations are sampled properly, i.e., in proportion to their weight, we will ensure that the Markov chain (i) is ergodic, i.e., that the QMC updates are capable of generating all basis states $|z\rangle$ as well as all sequences $S_{\mathbf{i}_q}$ evaluating to the identity, and (ii) satisfies detailed balance, a sufficient (but not strictly necessary) condition dictating that the ratio of transition probabilities from one configuration to another and the transition in the opposite direction equals to the ratio of

their respective weights [23,24]. In what follows, we show how both conditions are made to be satisfied.

B. Fundamental cycles

As noted above, for spin-1/2 Hamiltonians, the permutation operators P_i are Pauli-X strings. As such, they obey (i) $[P_i, P_j] = 0$ for every i, j, i.e., they all commute, and (ii) $P_i^2 = \mathbb{1}$. Moreover, each Pauli-X string can be represented as a bit string of the form $[b_1 b_2 \cdots b_n]$ where the ith bit $b_i \in \{0, 1\}$ indicates whether X_i is in the string ($b_i = 1$) or not ($b_i = 0$). (For example, $[1\,0\,1\,0\,0\,1]$ would correspond to the string $X_1 X_3 X_6$.) Denoting by p_i the bit string corresponding to the operator P_i, one can easily verify that the product of two permutation operators P_i and P_j would likewise correspond to the addition modulo 2 (the XOR operation) of the two respective bit strings, $p_i \oplus p_j$. Specifically, a sequence of operators evaluating to the identity, $S_{\mathbf{i}_q} = P_{i_q} \cdots P_{i_2} P_{i_1} = \mathbb{1}$, can be written as $\oplus_{j=1}^q p_{i_j} = \mathbf{0}$, where $\mathbf{0}$ is the zero bit string—the bit string consisting of only zeros.

Moreover, the following observations may be made:

(i) Any given sequence of permutation operators $S_{\mathbf{i}_q}$ that evaluates to the identity is a permutation of a multiset of operators, the product of which evaluates to the identity.

(ii) A multiset of operators obtained from another by the removal of a pair of identical operators will evaluate to the identity if and only if the original multiset evaluates to the identity (this follows from the commutativity of permutation operators and the fact that $P_i^2 = \mathbb{1}$).

(iii) Upon removing all pairs of identical operators from a multiset, one is left with a (proper) set of permutation operators, the product of which evaluates to the identity. This set consists of the permutation operators that are contained an odd number of times in the original sequence $S_{\mathbf{i}_q}$.

We will call a set of permutation operators that multiply to the identity a "cycle." In terms of binary strings, cycles may be represented by bit strings $[a_1 a_2 \cdots a_M]$ indicating which of the permutation operators $P_1, \ldots, P_M$ belong to the set. The length of a cycle would be the number of permutation operators in it (equivalently, the number of 1's in its binary vector representation). As pointed out above, any sequence of operators $S_{\mathbf{i}_q}$ has a "core" cycle from which $S_{\mathbf{i}_q}$ can be generated via insertions of operator pairs followed by reordering.

The question of how one can generate all possible sequences of operators (which evaluate to the identity) is thus reduced to the question of how one can generate all possible cycles (as the former can be derived from the latter). In terms of binary strings, the above question translates to generating the bit strings $[a_1 a_2 \cdots a_M]$ ($a_j \in \{0, 1\}$) that obey the following system of linear equations over mod 2 addition: $\oplus_{j=1}^M a_j p_j = \mathbf{0}$. This question can be answered by finding the null space over mod 2 addition of the matrix whose columns are the bit strings p_j. Any bit string $[a_1 a_2 \cdots a_M]$ obeying the condition $\oplus_{j=1}^M a_j p_j = \mathbf{0}$ can be written as mod 2 addition of bit strings from the null space basis.

Now, the null space basis can be found efficiently via Gaussian elimination over mod 2 addition [25]. We refer to cycles represented by the bit strings from the null space basis as fundamental cycles. Any cycle is therefore a combination of fundamental cycles.

Generally, the null space basis states can be chosen in many different ways, and so the choice of the set of fundamental cycles is not unique. From a practical point of view, however, we find that obtaining a "minimal cycle basis," i.e., a basis that minimizes the lengths of all basis cycles, is advantageous from the QMC standpoint. This follows from the fact that the probability of a QMC update to be accepted is a decreasing function of the cycle length (see Sec. IV D). To reduce the cycle lengths, we therefore find a basis using Gaussian elimination and then proceed to replace long-cycle basis states with shorter basis states by performing mod 2 additions between the bit strings of pairs of cycles, accepting the changes each time a new cycle with a shorter length is found. The process ends when a pass through all pairs of cycles does not result in an improvement.

It is worth noting that minimizing cycle lengths of Hamiltonians with nontrivial topologies will not always lead to a cycle basis whose fundamental cycle lengths are all of the order $O(1)$, i.e., do not grow with system size. In the presence of a nontrivial topology, e.g., where periodic boundary conditions are imposed or for other nonzero genus models, there may exist fundamental cycles becoming longer with the system size such as the cycles "wrapping around the system," consisting of an extensive number of permutation operators.

C. Ergodicity

We are now in a position to utilize the insights from the previous section to establish a set of updates to the sequence of operators $S_{\mathbf{i}_q}$ that would allow for the possibility of generating all possible operator sequences that evaluate to the identity.

As a preliminary step, prior to the simulation taking place, we produce a list of fundamental cycles $\mathcal{C}_1, \ldots, \mathcal{C}_T$, where T is the dimension of the null space, for the to-be-simulated Hamiltonian cast in PMR form.

We next prove that all possible sequences can be generated, i.e., that the Markov chain is ergodic, via a combination of insertion and removal of fundamental cycles, changing the order of the permutation operators in the sequence (permutation operator swapping), and the insertion and removal of pairs of identical permutation operators.

Claim. Every permutation operator sequence evaluating to the identity can be generated by (i) insertion and removal of fundamental cycles, (ii) insertion and removal of pairs of identical permutation operators, and (iii) swapping the order of two adjacent permutation operators.

Proof. For each permutation operator sequence $S_{\mathbf{i}_q}$ evaluating to the identity, one can consider the cycle $\mathcal{C}_{\mathbf{i}_q}$ consisting of permutation operators that appear an odd number of times in $S_{\mathbf{i}_q}$. In other words, by removal of pairs of identical operators, the multiset of operators that are contained in $S_{\mathbf{i}_q}$ can be reduced to the cycle $\mathcal{C}_{\mathbf{i}_q}$ consisting of permutation operators that appear only once.

As shown in Sec. IV B, the cycle $\mathcal{C}_{\mathbf{i}_q}$ can be expressed through fundamental cycles, since any cycle can be expressed in terms of the mod 2 null space of the matrix containing bit strings of permutation operators P_i from Eq. (1). Specifically, $c_{\mathbf{i}_q} = \oplus_{j=1}^{T} \alpha_j \mu_j$, where $c_{\mathbf{i}_q}$ is the bit-string representation of the cycle $\mathcal{C}_{\mathbf{i}_q}$, μ_j is a bit-string representation of the fundamental cycle $\mathcal{C}_j$, and $\alpha_j \in \{0, 1\}$, $j = 1, 2, \ldots, T$.

The latter observation implies that the cycle $\mathcal{C}_{\mathbf{i}_q}$ can be obtained by inserting those fundamental cycles $\mathcal{C}_j$ for which the condition $\alpha_j = 1$ is satisfied, accompanied by any necessary reordering of permutation operators and the removal of pairs of identical permutation operators. In turn, the sequence $S_{\mathbf{i}_q}$ can be obtained from $\mathcal{C}_{\mathbf{i}_q}$ via the addition of pairs of permutation operators together with reordering of permutation operators. This concludes the proof of our claim. ■

The ability to generate all permutation operator sequences that evaluate to the identity, $S_{\mathbf{i}_q}$, implies ergodicity along the quantum (or imaginary-time) dimension. In addition, the generation of all possible classical basis states $|z\rangle$ can independently be achieved by considering moves of the like of single- or multispin flips of the basis states. Moves of this nature guarantee ergodicity along the "classical" dimension.

In the next section we will use the above observations to devise QMC updates that ensure ergodicity in the entire configuration space, which is the direct product of the classical and quantum configuration spaces, i.e., the pairs $\mathcal{C} = \{|z\rangle, S_{\mathbf{i}_q}\}$.

D. QMC updates

We next describe the basic update moves for our QMC algorithm. To ensure both performance and high accuracy of evaluating the weight of any given configuration $\mathcal{C} = \{|z\rangle, S_{\mathbf{i}_q}\}$, we employ a divided differences calculation by means of addition and removal of items from the input multiset of classical energies [26]. We recall that the weight of a configuration requires the calculation of the divided differences $e^{-\beta[E_{z_0}, \ldots, E_{z_q}]}$ with inputs $[E_{z_0}, \ldots, E_{z_q}]$, where $E_{z_i} = \langle z_i | H | z_i \rangle = \langle z_i | D_0 | z_i \rangle$. For $S_{\mathbf{i}_q} = P_{i_q} \cdots P_{i_2} P_{i_1}$, we have $|z_0\rangle = |z\rangle$, $P_{i_j} |z_{j-1}\rangle = |z_j\rangle$ for $j = 1, 2, \ldots, q$. As it was shown in Ref. [26], upon adding an item to or removing an item from the input list of an already evaluated divided differences $e^{-\beta[E_0, \ldots, E_q]}$, the reevaluation can be done with only $O(sq)$ floating-point operations and $O(sq)$ bytes of memory, where $[E_0, \ldots, E_q]$ are the inputs and $s \propto \max_{i,j} |\beta E_i - \beta E_j|$.

Following the analysis of the previous section, to ensure the ergodicity of the Markov chain Monte Carlo, we find that employing the following QMC updates suffices.

1. Simple (local) swap

Simple swap is an update that consists of selecting a random integer m from $\{1, \ldots, q-1\}$, and then attempting to swap the permutation operators P_{i_m} and $P_{i_{m+1}}$ in $S_{\mathbf{i}_q}$, namely:

$$P_{i_m} P_{i_{m+1}} \rightarrow P_{i_{m+1}} P_{i_m}. \tag{12}$$

The updated sequence also evaluates to the identity. Since the internal classical state $|z_m\rangle$ may change, implementing the swap involves adding a new energy $E_{z'_m}$ and removing the old one E_{z_m} from the energy multiset. The acceptance probability for the update satisfying the detailed balance condition is

$$p = \min\left(1, \frac{W_{\mathcal{C}'}}{W_{\mathcal{C}}}\right), \tag{13}$$

where it follows from Eq. (7) that $W_{\mathcal{C}} = \mathrm{Re}[D_{\mathcal{C}}] e^{-\beta[E_{\mathcal{C}}]}$, $W_{\mathcal{C}'} = \mathrm{Re}[D_{\mathcal{C}'}] e^{-\beta[E_{\mathcal{C}'}]}$, and the energy multiset $E_{\mathcal{C}'} = E_{\mathcal{C}} + \{E_{z'_m}\} - \{E_{z_m}\}$. Here, $e^{-\beta[E_{\mathcal{C}}]}$ is a shorthand for $e^{-\beta[E_{z_0}, E_{z_1}, \ldots, E_{z_q}]}$ for the configuration $\mathcal{C}$ and likewise for $\mathcal{C}'$.

2. Pair insertion and deletion

Pair insertion is an update consisting of selecting random integers $m \in \{1, \ldots, q+1\}$ and $j \in \{1, \ldots, M\}$, and then attempting an insertion of the pair of permutation operators P_j into the $S_{\mathbf{i}_q}$ sequence in such a way that the updated operators P_{i_m} and $P_{i_{m+1}}$ are P_j. As a result, the length of the sequences increases: $q \to q+2$.

The updated sequence also evaluates to the identity. The input multiset of energies of the new configuration differs from that of the current configuration by having two additional energies $E_{z'_m}$ and $E_{z'_{m+1}}$. The weight of the new configuration is then $W_{\mathcal{C}'} = \mathrm{Re}[D_{\mathcal{C}'}]e^{-\beta[E_{\mathcal{C}'}]}$, where the energy multiset $E_{\mathcal{C}'} = E_{\mathcal{C}} + \{E_{z'_m}\} + \{E_{z'_{m+1}}\}$.

The reverse update of pair insertion is that of pair deletion. This update can be implemented only for $q \geqslant 2$. It consists of selecting a random integer $m \in \{1, \ldots, q-1\}$ checking the validity of the condition $P_{i_m} = P_{i_{m+1}}$, and attempting the removal of the pair of operators P_{i_m} and $P_{i_{m+1}}$ if this condition is true. The updated sequence also evaluates to the identity. The input multiset of energies of the new configuration differs from that of the current configuration by having one fewer E_{z_m} value and one fewer $E_{z_{m+1}}$ value. The weight of the new configuration is then proportional to $e^{-\beta[E_{\mathcal{C}'}]}$, where $E_{\mathcal{C}'} = E_{\mathcal{C}} - \{E_{z_m}\} - \{E_{z_{m+1}}\}$. The acceptance probabilities p_+ for pair insertion and p_- for pair deletion, satisfying the detailed balance condition, are

$$p_{\pm} = \min\left(1, M^{\pm 1}\frac{W_{\mathcal{C}'}}{W_{\mathcal{C}}}\right). \tag{14}$$

3. Block swap

Block swap is an update that involves a change of the classical state z. Here, a random position $k \in \{1, \ldots, q-1\}$ in the product $S_{\mathbf{i}_q}$ is picked such that the product is split into two (nonempty) subsequences, $S_{\mathbf{i}_q} = S_2 S_1$, with $S_1 = P_{i_k} \cdots P_{i_1}$ and $S_2 = P_{i_q} \cdots P_{i_{k+1}}$. The classical state $|z'\rangle$ at position k in the product is given by

$$|z'\rangle = S_1|z\rangle = P_{i_k} \cdots P_{i_1}|z\rangle, \tag{15}$$

where $|z\rangle$ is the classical state of the current configuration. The state $|z'\rangle$ has energy $E_{z'}$, and the state $|z\rangle$ has energy E_z. The new block-swapped configuration is $\mathcal{C}' = \{|z'\rangle, S_1 S_2\}$. The input multiset of energies of this configuration differs from that of the current configuration by having one fewer E_z value and one additional $E_{z'}$ value. The weight of the new configuration is then proportional to $e^{-\beta[E_{\mathcal{C}'}]}$ where the energy multiset $E_{\mathcal{C}'} = E_{\mathcal{C}} + \{E_{z'}\} - \{E_z\}$. The acceptance probability is as in Eq. (13) with the aforementioned $E_{\mathcal{C}'}$.

4. Classical updates

Classical moves are moves that involve a manipulation of the classical state $|z\rangle$ while leaving $S_{\mathbf{i}_q}$ unchanged. In a single bit-flip classical move, a spin from the classical bit-string state $|z\rangle$ of $\mathcal{C}$ is picked at random and is subsequently flipped, generating a state $|z'\rangle$ and hence a new configuration $\mathcal{C}'$. Calculating the weight of $\mathcal{C}'$ requires the calculation of the new energy multiset $E_{\mathcal{C}'}$ and recalculation of the divided differences, so it can become computationally intensive if q is large. Classical moves should therefore be attempted with relatively

low probabilities if q is large. Simply enough, the acceptance probability for a classical move satisfying the detailed balance condition is Eq. (13).

In the absence of a quantum part of the Hamiltonian ($M = 0$), not only are classical moves the only moves necessary, but they are also the only moves that have nonzero acceptance probabilities. Since the initial configuration of the QMC algorithm is a random classical configuration $|z\rangle$ and an empty operator sequence $S_0 = \mathbb{1}$, for a purely classical Hamiltonian, the algorithm automatically reduces to a classical thermal algorithm keeping the size of the imaginary-time dimension at zero ($q = 0$) for the duration of the simulation.

5. Fundamental cycle completion

Fundamental cycle completion is an update that consists of choosing a subsequence S from $S_{\mathbf{i}_q}$, choosing a fundamental cycle containing all operators of the subsequence S, and attempting to replace the subsequence S with the remaining operators from the selected cycle.

As is discussed in Sec. IV C, for the ergodicity condition to be fulfilled, it is necessary to be able to perform the insertion (or, equivalently, completion) of any of the fundamental cycles. For the probability of accepting an update to be non-negligible it is preferable that the change in the value of q will be minimal. The above implies that to complete a fundamental cycle of length l it is most advantageous to replace a subsequence of length r from $S_{\mathbf{i}_q}$ by the remaining $l - r$ permutation operators, where $r \approx l/2$.

We find that in some cases, there are fundamental cycles such that a simple cycle completion update, where the subsequence S comprises consecutive elements of $S_{\mathbf{i}_q}$, is always rejected for these cycles due to the zero weight of the resulting configuration [see Eq. (7)]. Thus, the fundamental cycle completion routine may never accept some of the fundamental cycles during the Markov process if inserted "as is." To resolve this issue, we have developed a subroutine that we refer to as "cycle completion with gaps," which does not require the elements of the sequence S to be consecutive within $S_{\mathbf{i}_q}$. Specific details of this protocol can be found in Appendix A.

6. Composite update

The role of the composite update is to ensure that nonfundamental cycles have the chance of being incorporated into the sequence of operators directly rather than via the concatenation of fundamental cycles which are inserted through the cycle completion move. The composite update is required for situations where fundamental cycle insertions may have zero weight [due to the vanishing of one or more of the matrix elements in the product $D_{(z, S_{\mathbf{i}_q})}$; see Eq. (7)], whereas nonfundamental cycles may not.

The update consists of a combination of several basic updates and is described as follows.

(i) Perform one of the basic QMC updates equally likely: either a simple swap, a pair insertion, a pair deletion, or a fundamental cycle completion.

(ii) If the resulting weight is zero, reject the entire update with a probability of $1/2$, and with the remaining probability, return to step (i).

(iii) Finalize the update $\mathcal{C}_1 \to \mathcal{C}_2 \to \cdots \to \mathcal{C}_k$ with the following acceptance probability, which satisfies the detailed balance condition:

$$P_{\text{accept}}(\mathcal{C}_1 \to \cdots \to \mathcal{C}_k)$$
$$= \min\left(1, \frac{W_{\mathcal{C}_k}}{W_{\mathcal{C}_1}} R(\mathcal{C}_1, \mathcal{C}_2) \cdots R(\mathcal{C}_{k-1}, \mathcal{C}_k)\right). \quad (16)$$

Here, $R(\mathcal{C}, \mathcal{C}') = 1$ when the update $\mathcal{C} \to \mathcal{C}'$ is a simple swap, $R(\mathcal{C}, \mathcal{C}') = M$ when it is a pair insertion, $R(\mathcal{C}, \mathcal{C}') = 1/M$ when it is a pair deletion, and $R(\mathcal{C}, \mathcal{C}') = (p_r(q') \cdot n_c \cdot r'!)/(p_r(q) \cdot n'_c \cdot r!)$ when it is a cycle completion (see Appendix A).

To prove that the QMC updates ensure ergodicity in the entire configuration space, consider two arbitrary configurations $\mathcal{C} = \{|z\rangle, S_{\mathbf{i}_q}\}$ and $\mathcal{C}' = \{|z'\rangle, S_{\mathbf{i}'_{q'}}\}$ such that $W_{\mathcal{C}} \neq 0$ and $W_{\mathcal{C}'} \neq 0$. It follows from Sec. IV C that the above QMC updates allow in particular the following sequence of transformations: $\mathcal{C} \to \mathcal{C}_0 \to \mathcal{C}'_0 \to \mathcal{C}'$, where $\mathcal{C}_0 = \{|z\rangle, \mathbb{1}\}$ and $\mathcal{C}'_0 = \{|z'\rangle, \mathbb{1}\}$. Hence, the transformation from $\mathcal{C}$ to $\mathcal{C}'$ is possible, and the ergodicity holds.

7. Worm update

An alternative to the composite update, also capable of incorporating nonfundamental cycles, is a worm-type global update for PMR QMC. This update involves introducing a "disturbance" (or a "worm head") into the sequence of operators $S_{\mathbf{i}_q}$ by either appending $S_{\mathbf{i}_q}$ with a single operator or removing one from it (we will refer to this addition or removal of an operator as "single operator moves"). Insertion or removal of a single permutation operator causes the sequence to evaluate to a nonidentity permutation, thus resulting in a zero-weight configuration. Consequently, the disturbed sequence must be "healed" back to an identity-forming sequence. The healing process involves introducing further moves, either by employing the basic updates described above such as simple swap, fundamental cycle completion, pair insertion, and pair deletion or by applying additional single-operator moves. These single-operator moves have the power to heal the sequence. After each such move, the instantaneous sequence $S_{\mathbf{i}_q}$ is checked to determine if it evaluates to the identity. If it does, the worm update ends; if not, additional moves are required.

To make sure that detailed balance is conserved, and that the acceptance rates of intermediate worm moves are sufficiently high, we assign nonidentity intermediate configurations (sequences of operators that do not evaluate to the identity) their "natural" weight $W_{\mathcal{C}}$ as per Eq. (7). This allows intermediate moves to be accepted or rejected with probabilities obeying detailed balance. Additionally, to prevent the worm from straying too far from being healed (in other words, to ensure that the sequence of operators is not too far from the identity), we introduce a small probability p_f to reject the entire worm update at each intermediate state.

The worm update is specified as follows:

(i) Start with a sequence of operators $S_{\mathbf{i}_q}$ that evaluates to the identity, and store $S_{\mathbf{i}_q}$.

(ii) Generate a modified sequence of operators by applying one of the following updates with equal probabilities: either a local swap, fundamental cycle completion, pair insertion or

deletion, or a single-operator move. Accept or reject the new configuration with probabilities obeying detailed balance.

(iii) The worm update ends if the new $S_{\mathbf{i}_q}$ evaluates to the identity. If it does not evaluate to the identity, revert to the stored $S_{\mathbf{i}_q}$ and exit with probability p_f; with the remaining probability, return to step (ii).

V. MEASUREMENTS

The PMR formulation allows one to measure a wide range of static operators and additional dynamical quantities [27]. The key to being able to do so is to write for any given operator A its thermal average as

$$\langle A \rangle = \frac{\text{Tr}[A e^{-\beta H}]}{\text{Tr}[e^{-\beta H}]} = \frac{\sum_{\mathcal{C}} A_{\mathcal{C}} w_{\mathcal{C}}}{\sum_{\mathcal{C}} w_{\mathcal{C}}}. \quad (17)$$

Although, generally, both $w_{\mathcal{C}}$ and $A_{\mathcal{C}}$ are complex valued, both the sums $\sum_{\mathcal{C}} A_{\mathcal{C}} w_{\mathcal{C}}$ and $\sum_{\mathcal{C}} w_{\mathcal{C}}$ are real valued since both H and A are Hermitian operators. Therefore, we have

$$\langle A \rangle = \frac{\sum_{\mathcal{C}} \text{Re}[A_{\mathcal{C}} w_{\mathcal{C}}] / \text{Re}[w_{\mathcal{C}}] \cdot W_{\mathcal{C}}}{\sum_{\mathcal{C}} W_{\mathcal{C}}}, \quad (18)$$

where $W_{\mathcal{C}} = \text{Re}[w_{\mathcal{C}}]$. The quantity $\text{Re}[A_{\mathcal{C}} w_{\mathcal{C}}] / \text{Re}[w_{\mathcal{C}}]$ is therefore the instantaneous quantity associated with the configuration $\mathcal{C} = \{z, S_{\mathbf{i}_q}\}$ that will be gathered during the simulation.

A. Measurements of standard observables

We next provide the instantaneous quantities to be collected throughout the simulation for the following operators: (i) the Hamiltonian H, (ii) the Hamiltonian squared, H^2, (iii) the diagonal component of the Hamiltonian, $H_{\text{diag}} = D_0$, (iv) the diagonal component squared, H^2_{diag}, (v) the off-diagonal component of the Hamiltonian, $H_{\text{offdiag}} = \sum_{j=1}^{M} D_j P_j$, and (vi) the off-diagonal component of the Hamiltonian squared, H^2_{offdiag}.

We have $-\partial e^{-\beta[E_{z_0}, \ldots, E_{z_q}]} / \partial \beta = h[E_{z_0}, \ldots, E_{z_q}]$, where $h(E) = E e^{-\beta E}$. Using the Leibniz rule for divided differences [21,22], we obtain

$$-\frac{\partial w_{\mathcal{C}}}{\partial \beta} = H_{\mathcal{C}} w_{\mathcal{C}}, \quad (19)$$

where

$$H_{\mathcal{C}} = \begin{cases} E_{z_0} & \text{for } q = 0 \\ E_{z_0} + \dfrac{e^{-\beta[E_{z_1}, \ldots, E_{z_q}]}}{e^{-\beta[E_{z_0}, \ldots, E_{z_q}]}} & \text{for } q > 0. \end{cases} \quad (20)$$

Therefore,

$$\langle H \rangle = -\frac{1}{Z} \frac{\partial Z}{\partial \beta} = \frac{\sum_{\mathcal{C}} H_{\mathcal{C}} w_{\mathcal{C}}}{\sum_{\mathcal{C}} w_{\mathcal{C}}}, \quad (21)$$

which coincides with Eq. (17) for $A = H$. Similarly, we have $\langle H^2 \rangle = \langle H \rangle^2 - \partial \langle H \rangle / \partial \beta$, where

$$\frac{\partial}{\partial \beta} \langle H \rangle = \frac{1}{Z} \frac{\partial}{\partial \beta} \left(\sum_{\mathcal{C}} H_{\mathcal{C}} w_{\mathcal{C}} \right) + \langle H \rangle^2. \quad (22)$$

It follows that $(H^2)_\mathcal{C} = (H_\mathcal{C})^2 - \partial H_\mathcal{C}/\partial\beta$, and

$$(H^2)_\mathcal{C} = \begin{cases} E_{z_0}^2 & \text{for } q = 0 \\ E_{z_0}^2 + \dfrac{(E_{z_0}+E_{z_1})e^{-\beta E_{z_1}}}{e^{-\beta[E_{z_0},E_{z_1}]}} & \text{for } q = 1 \\ E_{z_0}^2 + (E_{z_0} + E_{z_1})\dfrac{e^{-\beta[E_{z_1},...,E_{z_q}]}}{e^{-\beta[E_{z_0},...,E_{z_q}]}} + \dfrac{e^{-\beta[E_{z_2},...,E_{z_q}]}}{e^{-\beta[E_{z_0},...,E_{z_q}]}} & \text{for } q > 1. \end{cases} \tag{23}$$

It is straightforward to obtain the remaining expressions:

$$(H_{\text{diag}})_\mathcal{C} = E_{z_0}, \tag{24}$$

$$\left(H_{\text{diag}}^2\right)_\mathcal{C} = E_{z_0}^2, \tag{25}$$

$$(H_{\text{offdiag}})_\mathcal{C} = H_\mathcal{C} - E_{z_0}, \tag{26}$$

$$\left(H_{\text{offdiag}}^2\right)_\mathcal{C} = H_\mathcal{C}^2 + E_{z_0}\left(E_{z_0} - 2H_\mathcal{C}\right). \tag{27}$$

B. Measurements of custom static operators

We next consider the measurement of a general static operator A. We proceed by casting it in PMR form, i.e., as $A = \sum_i \tilde{A}_i \tilde{P}_i$, where each $\tilde{A}_i$ is diagonal and the $\tilde{P}_i$'s are permutation operators. The operators $\tilde{P}_i$ belong to the group G_X containing all possible Pauli-X tensor products, whose elements appear in the PMR representation of the Hamiltonian (see Sec. III). We note that in the most general case, the $\tilde{P}_i$ operators in the representation of A may not all appear in the Hamiltonian's PMR decomposition, Eq. (1). Nonetheless, we can always write

$$\langle A \rangle = \frac{\text{Tr}\,[Ae^{-\beta H}]}{\text{Tr}\,[e^{-\beta H}]} = \sum_i \frac{\text{Tr}\,[\tilde{A}_i\tilde{P}_ie^{-\beta H}]}{\text{Tr}\,[e^{-\beta H}]}, \tag{28}$$

and we may focus on a single $\tilde{A}\tilde{P}$ term at a time. Carrying out the off-diagonal expansion [27], we end up with

$$\text{Tr}\,[\tilde{A}\tilde{P}e^{-\beta H}] = \sum_z \tilde{A}(z) \sum_{q=0}^\infty \sum_{S_{\mathbf{i}_q}} D_{(z,S_{\mathbf{i}_q})} \\ \times e^{-\beta[E_{z_0},...,E_{z_q}]}\langle z|\tilde{P}S_{\mathbf{i}_q}|z\rangle, \tag{29}$$

where $D_{(z,S_{\mathbf{i}_q})}e^{-\beta[E_{z_0},...,E_{z_q}]}$ is the weight of the configuration $(z, S_{\mathbf{i}_q})$.

We differentiate between two cases: (i) the operator $\tilde{P}$ appears in the Hamiltonian or can be written as a product of permutation operators that appear in the Hamiltonian, and (ii) the operator $\tilde{P}$ does not appear in the Hamiltonian and cannot be written as a product of permutation operators that do.

In the latter case, since every $S_{\mathbf{i}_q}$ sequence consists of the permutation operators that appear in the Hamiltonian, we always have $\langle z|\tilde{P}S_{\mathbf{i}_q}|z\rangle = 0$. Then, it follows from Eq. (29) that $\langle\tilde{A}\tilde{P}\rangle = 0$.

In the former case, the operator to be measured has the form $A = \tilde{A}\tilde{P}$, where $\tilde{A}$ is diagonal and $\tilde{P} = P_{i_1}P_{i_2}\cdots P_{i_k}$. We modify Eq. (29) so that $(z, \tilde{S}_{\mathbf{i}_q})$ with $\tilde{S}_{\mathbf{i}_q} = \tilde{P}S_{\mathbf{i}_q}$ is seen as a configuration instead of $(z, S_{\mathbf{i}_q})$. Thus, we arrive at

$$\langle A \rangle = \frac{\sum_{(z,\tilde{S}_{\mathbf{i}_q})} w_{(z,\tilde{S}_{\mathbf{i}_q})}M_{\tilde{A}\tilde{P}}(z, \tilde{S}_{\mathbf{i}_q})}{\sum_{(z,\tilde{S}_{\mathbf{i}_q})} w_{(z,\tilde{S}_{\mathbf{i}_q})}}, \tag{30}$$

where

$$M_{\tilde{A}\tilde{P}}(z, \tilde{S}_{\mathbf{i}_q}) = \delta_{\tilde{P}}\tilde{A}(z)\frac{1}{D_{(z,\tilde{P})}}\frac{e^{-\beta[E_{z_0},...,E_{z_{q-k}}]}}{e^{-\beta[E_{z_0},...,E_{z_q}]}}. \tag{31}$$

In the above, $\delta_{\tilde{P}} = 1$ if the leftmost operators of $\tilde{S}_{\mathbf{i}_q}$ are $P_{i_1}P_{i_2}\cdots P_{i_k}$ and is zero otherwise, and

$$D_{(z,\tilde{P})} = \frac{D_{(z,\tilde{S}_{\mathbf{i}_q})}}{D_{(z,S_{\mathbf{i}_q})}} = \prod_{m=1}^k \langle z_{q-m+1}|D_{i_m}|z_{q-m+1}\rangle. \tag{32}$$

The above formulation allows automatic measurements of static operators as long as these are set up as linear combinations of Pauli strings as in Eq. (8). For an arbitrary static operator A given as a linear combination of Pauli strings, one is required to cast the expression in the form $\sum_i \tilde{A}_i\tilde{P}_i$. As a next step, the $\tilde{P}_i$ operators are expressed in terms of the permutations P_i of the Hamiltonian. Here, the Gaussian elimination over mod 2 addition can be used [25]. Finally, Eqs. (30) and (31) are employed to compute the thermal average $\langle A \rangle$.

However, for the thermal average of a custom observable to be computed correctly, it is necessary to account for the following restrictions on its structure. We find that employing Eqs. (30) and (31) leads to obtaining the correct value of the thermal average $\langle A \rangle$ under the condition that $\tilde{A}(z) = 0$ for all basis states $|z\rangle$ for which $D_{(z,\tilde{P})} = 0$. Otherwise, it is possible that the following conditions hold for some of the configurations: $D_{(z,\tilde{P})} = 0$ and $D_{(z,\tilde{S}_{\mathbf{i}_q})} = 0$ while $D_{(z,S_{\mathbf{i}_q})} \neq 0$ and $\tilde{A}(z) \neq 0$, where $\tilde{S}_{\mathbf{i}_q} = \tilde{P}S_{\mathbf{i}_q}$. Such configurations $(z, S_{\mathbf{i}_q})$ have a nonzero contribution in Eq. (29), but the Markov chain does not generate the corresponding configurations $(z, \tilde{S}_{\mathbf{i}_q})$ because they have zero weights: $W_{(z,\tilde{S}_{\mathbf{i}_q})} = 0$.

In particular, it is possible to obtain the thermal average of any observable of the form $A = \sum_i A_iD_iP_i$, where each A_i is an arbitrary diagonal matrix and each D_iP_i is a generalized permutation matrix appearing in Eq. (1). Any operator A satisfying $\langle z|A|z'\rangle = 0$ for all basis states $|z\rangle$ and $|z'\rangle$ such that $\langle z|H|z'\rangle = 0$ can be written in such a form and hence its thermal average (30) will be computed correctly.

In addition to the above measurement protocol, there is also a way to correctly obtain a thermal average of a static operator A which can be written in the form $A = \sum_i \tilde{P}^{(i)}$, where each $\tilde{P}^{(i)}$ is a matrix of the general form

$$\tilde{P}^{(i)} = A_k^{(i)}D_{j_k}P_{j_k} \cdots A_2^{(i)}D_{j_2}P_{j_2} \cdot A_1^{(i)}D_{j_1}P_{j_1} \cdot A_0^{(i)}, \tag{33}$$

where $A_0^{(i)}, \ldots, A_k^{(i)}$ are arbitrary diagonal matrices and $D_{j_1}P_{j_1}, \ldots, D_{j_k}P_{j_k}$ are the generalized permutation matrices appearing in Eq. (1).

Focusing on a single such term, we calculate the expectation value for an observable of the form

$$\bar{P} = A_k D_{j_k} P_{j_k} \cdots A_2 D_{j_2} P_{j_2} A_1 D_{j_1} P_{j_1} A_0, \qquad (34)$$

where each of the $D_{j_k}P_{j_k}$ appear in the PMR decomposition of the Hamiltonian and the A_k matrices are arbitrary diagonal operators. Carrying out the off-diagonal expansion for $\text{Tr}[\bar{P}e^{-\beta H}]$, we first obtain

$$\text{Tr}[\bar{P}e^{-\beta H}] = \sum_z M_{\bar{P}}(z, S_{\mathbf{i}_q}) \sum_{q=0}^{\infty} \sum_{S_{\mathbf{i}_q}} D_{(z,S_{\mathbf{i}_q})}$$
$$\times e^{-\beta[E_{z_0}, \ldots, E_{z_q}]} \langle z | \tilde{P} S_{\mathbf{i}_q} | z \rangle, \qquad (35)$$

where

$$M_{\bar{P}}(z, S_{\mathbf{i}_q}) = A_k(z_{q+k}) d_{z_{q+k}}^{(j_k)} \cdots A_1(z_{q+1}) d_{z_{q+1}}^{(j_1)} A_0(z_q), \qquad (36)$$

with $A_i(z_j) \equiv \langle z_j | A_i | z_j \rangle$. Note that there is a one-to-one correspondence between nonvanishing terms $\langle z | \tilde{P} S_{\mathbf{i}_q} | z \rangle = \langle z | P_{j_k} \cdots P_{j_2} P_{j_1} S_{\mathbf{i}_q} | z \rangle$ and nonvanishing terms $\langle z | S_{\mathbf{i}_q} | z \rangle$ which appear in the partition function expansion. We can therefore rewrite the above as

$$\text{Tr}[\bar{P}e^{-\beta H}]$$
$$= \sum_z \left(\prod_{j=0}^k A_j(z_{q+j}) \right) \sum_{q=0}^{\infty} \sum_{S_{\mathbf{i}}} D_{(z,S_{\tilde{\mathbf{i}}})}$$
$$\times e^{-\beta[E_{z_0}, \ldots, E_{z_q}]} \frac{D_{(z,S_{\mathbf{i}})} e^{-\beta[E_{z_0}, \ldots, E_{z_{q+k}}]}}{D_{(z,S_{\tilde{\mathbf{i}}})} e^{-\beta[E_{z_0}, \ldots, E_{z_{q+k}}]}} \langle z | S_{\tilde{\mathbf{i}}} | z \rangle, \quad (37)$$

where $D_{(z,S_{\tilde{\mathbf{i}}})} e^{-\beta[E_{z_0}, \ldots, E_{z_{q+k}}]}$ is the weight of configuration $(z, S_{\tilde{\mathbf{i}}})$ with $S_{\tilde{\mathbf{i}}} = P_{j_k} \cdots P_{j_2} P_{j_1} S_{\mathbf{i}_q}$. This gives

$$\text{Tr}[\bar{P}e^{-\beta H}] = \sum_z \sum_{q=0}^{\infty} \sum_{S_{\mathbf{i}_q}} D_{(z,S_{\mathbf{i}_q})} e^{-\beta[E_{z_0}, \ldots, E_{z_q}]}$$
$$\times M_{\bar{P}}(z, S_{\mathbf{i}_q}) \langle z | S_{\mathbf{i}_q} | z \rangle, \qquad (38)$$

where $M_{\bar{P}}(z, S_{\mathbf{i}_q})$ is redefined as

$$M_{\bar{P}}(z, S_{\mathbf{i}_q}) = \delta_{\bar{P}} \left(\prod_{j=0}^k A_j(z_{q-k+j}) \right) \frac{e^{-\beta[E_{z_0}, \ldots, E_{z_{q-k}}]}}{e^{-\beta[E_z, \ldots, E_{z_q}]}}. \qquad (39)$$

In the above expression, $\delta_{\bar{P}} = 1$ if the leftmost operators of $S_{\mathbf{i}_q}$ are $P_{j_1}P_{j_2}\cdots P_{j_k}$ and is zero otherwise.

Denoting $\tilde{A}(z, S_{\mathbf{i}_q}) = \sum_{i=0}^q M_{\bar{P}_i}(z, S_{\mathbf{i}_q})$, we can thus write $\langle A \rangle$ as

$$\langle A \rangle = \left\langle \sum_i \bar{P}^{(i)} \right\rangle = \frac{\sum_{(z,S_{\mathbf{i}_q})} w_{(z,S_{\mathbf{i}_q})} \tilde{A}(z, S_{\mathbf{i}_q})}{\sum_{(z,S_{\mathbf{i}_q})} w_{(z,S_{\mathbf{i}_q})}}. \qquad (40)$$

C. Monitoring the average sign

A necessary condition for the proper importance sampling of partition function weights is that all weights are nonnegative. Whenever the partition function expansion produces negative weights [see Eq. (7)], the system is said to possess a

sign problem [28]. In the presence of a sign problem, configuration weights cannot be treated as unnormalized probabilities as they should in Markov chain Monte Carlo simulations. In such cases, a common workaround is to take the configuration weights to be the absolute values of the original ones, $|W_{\mathcal{C}}|$ [28–30]. By doing so, a thermal average of an observable A is rewritten as

$$\langle A \rangle = \frac{\langle A \cdot \text{sgn}(W) \rangle_{|W|}}{\langle \text{sgn}(W) \rangle_{|W|}}, \qquad (41)$$

where the average sign

$$\langle \text{sgn} \rangle = \langle \text{sgn}(W) \rangle_{|W|} = \frac{\sum_{\mathcal{C}} W_{\mathcal{C}}}{\sum_{\mathcal{C}} |W_{\mathcal{C}}|} \qquad (42)$$

is monitored throughout the simulation. The average sign is usually considered a figure of merit of how adverse the sign problem is. For models that do not have a sign problem, all weights are positive, $\langle \text{sgn} \rangle = 1$, and the expression for $\langle A \rangle$ naturally reduces to its original form, Eq. (18). For sign-problematic systems, $\langle \text{sgn} \rangle \approx 0$, and one would expect to obtain extremely large error bars for $\langle A \rangle$ that would as a result require an exponentially long simulation time for an accurate computation of observables.

As we demonstrate in the next section, in some cases accurate calculations are achievable even when $\langle \text{sgn} \rangle$ is relatively small. In Appendix B we provide an improved approximation of $\langle A \rangle$ in the presence of a sign problem and an approximation for the statistical error $\sigma(A)$, which can be determined during the simulation.

VI. RESULTS AND DISCUSSION

In this section, we demonstrate the success of our method in simulating a variety of large-scale quantum many-body systems, taking as test cases models that would help highlight the extensive scope of the algorithm.

Wherever exact calculations were possible, we have verified the correctness and accuracy of our technique by ensuring that the calculated values agree with exact values.

A. The XY model on a triangular lattice

We consider the prototypical quantum anisotropic XY (XZ) model [31–33] on a triangular lattice with open boundary conditions (see Fig. 1). This model and variants thereof have been used extensively as simplified models for a variety of physical systems such as liquid helium, high-T_c superconductors, anisotropic magnets, and more. The Hamiltonian we study is

$$H = J \sum_{\langle jk \rangle} Z_j Z_k + \Gamma \sum_{\langle jk \rangle} X_j X_k, \qquad (43)$$

where $\langle jk \rangle$ denotes neighbors on a triangular lattice with $n = L^2$ spins containing L sites on each side. The above XY model gives rise to a severe sign problem, preventing a true quantitative understanding of the phase diagram of the model if $J > 0$ and $\Gamma > 0$, which will be our region of interest. Specifically, we will consider a system with $L = 8$ and $J = 1$ and allow the parameter Γ, which serves as the strength of the quantum component of the Hamiltonian, to vary.
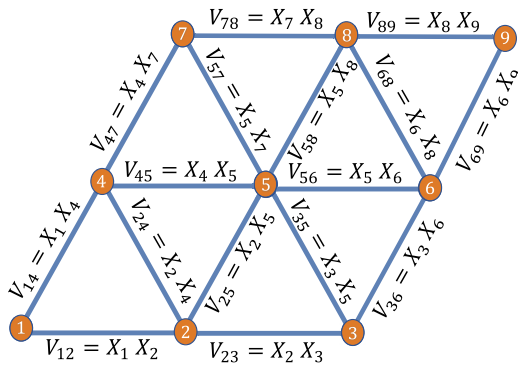
FIG. 1. A triangular lattice with open boundary conditions. Here the side length is $L = 3$.

The PMR form of the Hamiltonian, Eq. (43), is

$$H = D_0 + \Gamma \sum_{\langle jk \rangle} V_{jk}. \qquad (44)$$

Here, $D_0 = J \sum_{\langle jk \rangle} Z_j Z_k$ is the "classical" component of the Hamiltonian that is diagonal in the computational basis. The set $\{V_{jk} = X_j X_k\}$ consists of off-diagonal permutation operators that give the system its "quantum dimension" and obey $V_{ij}|z\rangle = |z'\rangle$ for every basis state $|z\rangle$, where $|z'\rangle \neq |z\rangle$ is also a basis state differing from $|z\rangle$ by two spin flips. There are $M = 3L^2 - 4L + 1$ off-diagonal operators V_{jk}, one for each edge of the lattice (see Fig. 1).

Since $V_{ij} V_{jk} V_{ki} = \mathbb{1}$ for every triplet of spins with indices i, j, and k that form a basic triangle of the lattice (or a triangular plaquette), we conclude that the model admits $2(L - 1)^2$ fundamental cycles, one for each basic triangle (see Sec. IV B).

Figure 2 shows the computed mean energy, its error bar, and $\langle sgn \rangle$ as a function of β and Γ. Here, we have used 2×10^9 Monte Carlo (MC) updates. As is shown in Fig. 2, the observables can be accurately calculated using our method even in cases where the values of $\langle sgn \rangle$ are as small as 10^{-3}. However, as expected, the values of $\langle sgn \rangle$ decrease exponentially with both β and Γ.

Table I shows the increase in complexity of the calculation with the severity of the sign problem in more detail. In particular, one can see that the wall-clock time is roughly proportional to $\langle q \rangle$ for $\langle q \rangle > 1$ (in agreement with prior results pertaining to divided-differences calculations [26]). This is

TABLE I. Dependence of the expansion order and average sign on the off-diagonal strength Γ for the antiferromagnetic XY model on a triangular lattice. Calculations are shown for $L = 8$, $J = 1$, and $\beta = 1$.

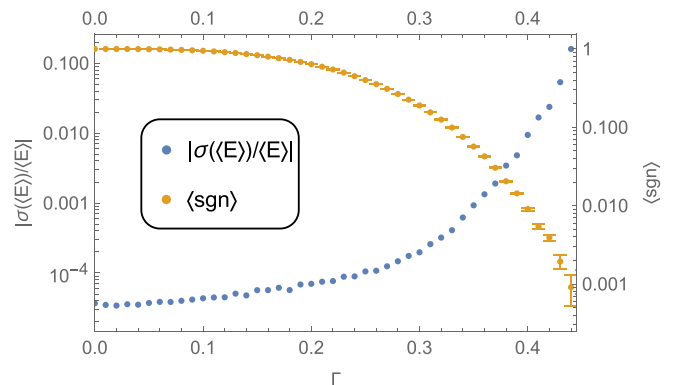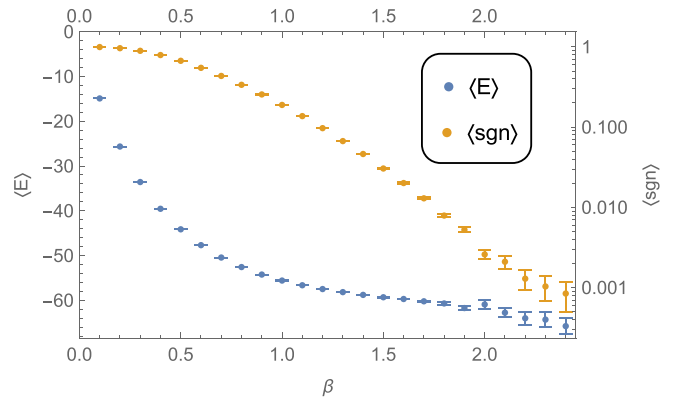| Γ	$\langle sgn \rangle$	$\langle q \rangle$	max(q)	Time per MC update (s)
0.05	0.994	0.2	9	2.0×10^{-6}
0.1	0.95	0.84	13	2.7×10^{-6}
0.2	0.64	4	30	5.4×10^{-6}
0.3	0.19	11	46	1.2×10^{-5}
0.35	0.06	17	59	1.8×10^{-5}
0.4	0.01	27	75	3.5×10^{-5}
0.45	0.001	41	93	5.7×10^{-5}

FIG. 2. Top: Dependence of mean energy and $\langle sgn \rangle$ on β for $L = 8$ and $\Gamma = 0.3$. Bottom: $\langle sgn \rangle$ and relative error of the mean energy as a function of Γ for $L = 8$ and $\beta = 1$.

because most of the computing time is spent on the calculation and reevaluation of the divided differences.

B. The XY model on a square lattice: Dependence of convergence time on temperature

We next investigate the dependence of simulation runtimes on the inverse-temperature β, using as a test case the XY model on a square lattice with periodic boundary conditions imposed. Similar to the XY model on the triangular lattice discussed in the previous section, the permutations of the XY model on the square lattice are two-body XX interactions. The fundamental cycles are therefore length-four products of permutations corresponding to edges surrounding each plaquette. On top of these, the periodic boundary conditions induce additional cycles that wrap around the lattice. For an 8×8 square lattice, the null space consists of 65 cycles, of which 56 are of length four and 9 of length eight wrapping around the lattice either horizontally or vertically.

Figure 3 shows the wall-clock time of 10^7 QMC updates as a function of β, fitted by the dashed curve with the optimal fit of $3.14 \times \beta^{2.23}$. The inset shows the estimated error $\sigma(\langle E \rangle)$ of 10^7 QMC updates versus β. The statistical error turns out to be of the same order of magnitude across the entire tested temperature range.

We thus find that the convergence time of the algorithm grows relatively slowly with β, obeying a modest power law, indicating that low-temperature simulations are readily attainable.
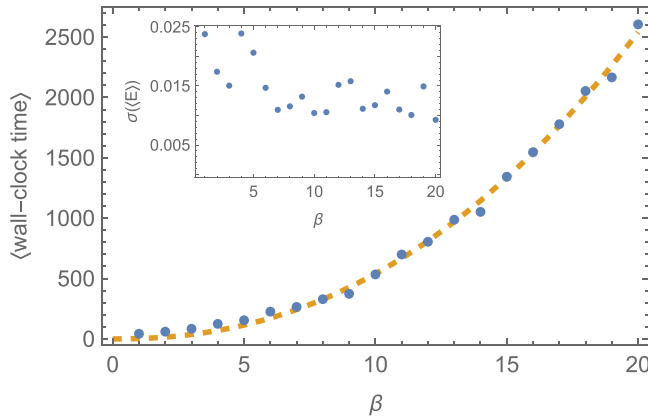
FIG. 3. Calculations of the XY model on a square lattice with periodic boundary conditions: wall-clock time in seconds of 10^7 QMC updates as a function of β. Parameters: $L = 8$, $J = 1$, and $\Gamma = -0.25$. The dashed line is $3.14 \times \beta^{2.23}$. Inset: Estimated statistical error of 10^7 QMC updates as a function of β.

C. Topological models

As mentioned above, in the presence of a nontrivial topology, e.g., where periodic boundary conditions are imposed or for other nonzero genus models, there may exist fundamental cycles that grow with the system size such as the cycles "wrapping around the system," consisting of an extensive number of permutation operators (as was the case in the XY model on a square lattice with periodic boundary conditions discussed above). For such models, some fundamental cycle lengths will be of the order $O(N)$, i.e., grow with system size [as opposed to being $O(1)$]. For that reason, these models are usually exceptionally difficult to study as they require so-called global rather than local moves.

To demonstrate that our proposed method can successfully solve these models as well, we next present some results pertaining to the well-known toric code model. The toric code is defined on a periodic two-dimensional lattice (a torus), usually chosen to be the square lattice, with a spin-1/2 particle located on each edge. The Hamiltonian of the toric code is given by [34,35]

$$H_{\text{toric}} = J \left(\sum_v A_v + \sum_p B_p \right), \quad (45)$$

where $J > 0$ and $A_v = \prod_{i \in v} X_i$ and $B_p = \prod_{i \in p} Z_i$ with $i \in v$ denoting the edges touching the vertex v, and $i \in p$ denoting the edges surrounding the plaquette p. In PMR, the Hamiltonian is rewritten as

$$H_{\text{toric}} = D_0 + \sum_v D_v P_v, \quad (46)$$

where $D_0 = \sum_p B_p$, $P_v = A_v$, and $D_v = J \cdot \mathbb{1}$. The only fundamental cycle is equal to the product of all plaquette terms $P \equiv \prod_v P_v$.

The local moves of insertion and deletion of pairs of permutation operators are, however, not ergodic on their own. This can be seen by noticing that the annihilation and creation local updates imply that plaquette terms P_v appear in even numbers. On the other hand, the product of all plaquette terms
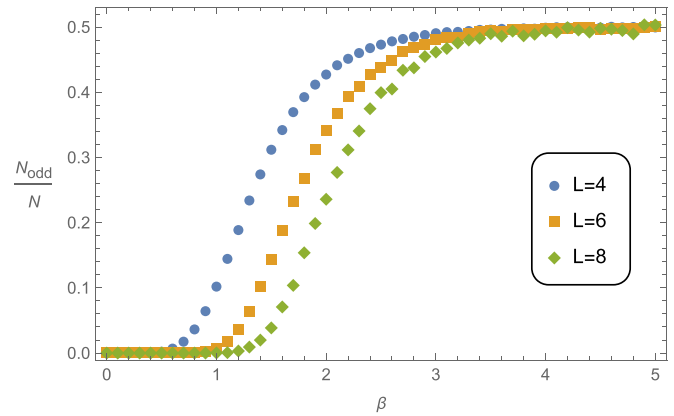

FIG. 4. Toric code: The fraction of time in the odd sector as a function of β for $L = 4$, $L = 6$, and $L = 8$.

$P \equiv \prod_v P_v$ also evaluates to the identity and so sequences where all plaquette terms are odd numbered should also be accounted for. There are thus two topological sectors: an even one and an odd one. For the simulation to sample configurations in both, there must be a global move that jumps between the two and which changes the parity of all plaquette operators. As follows from Sec. IV C, the fundamental cycle completion accomplishes this move.

Figure 4 shows N_{odd}/N versus β, where N_{odd} is the number of visited odd-sector configurations, and N is the total number of visited configurations. As can be seen, the Markov chain switches easily between the even sector and the odd one for sufficiently low temperatures, indicating that the Markov chain is ergodic and the algorithm works properly. The above results also agree with the expected behavior of toric code, where transitions between ground states are generated by pairs of excitations at sufficiently low temperatures [36].

D. Random spin-1/2 Hamiltonians

To illustrate the versatility of our algorithm, we next present simulation results for randomly generated spin-1/2 Hamiltonians. We produce random n-spin m-term k-local spin-1/2 Hamiltonians by adding together m randomly generated Pauli strings. To create a k-local Pauli string, we first sample k spin indices $i_1, \ldots, i_k$ from the set of spin indices $\{1, \ldots, n\}$. For each chosen index, we pick at random an operator from the set $\{X, Y, Z\}$, thereby creating a Pauli string with locality k. Our final random Hamiltonian attains the form $\sum_i c^{(i)} S^{(i)}$, where each randomly generated Pauli string $S^{(i)}$ is multiplied by a real-valued coefficient $c^{(i)}$ randomly drawn from the interval $[-1, 1]$.

To demonstrate the ease with which our approach allows the simulation of such systems, we have generated random 40-spin m-term Hamiltonians with m varying from $m = 1$ to $m = 50$, simulating 200 randomly generated instances per each value of m at $\beta = 1$ for three choices of locality $k = 3$, $k = 5$, and $k = 8$. Figure 5 (top) shows the average of the energy $\langle E \rangle$ over the 200 instances as a function of m. The error bars indicate magnitude of fluctuations of the averaged energy. The bottom panel of Fig. 5 depicts the average sign $\langle \text{sgn} \rangle$, averaged over the 200 instances per each choice of m
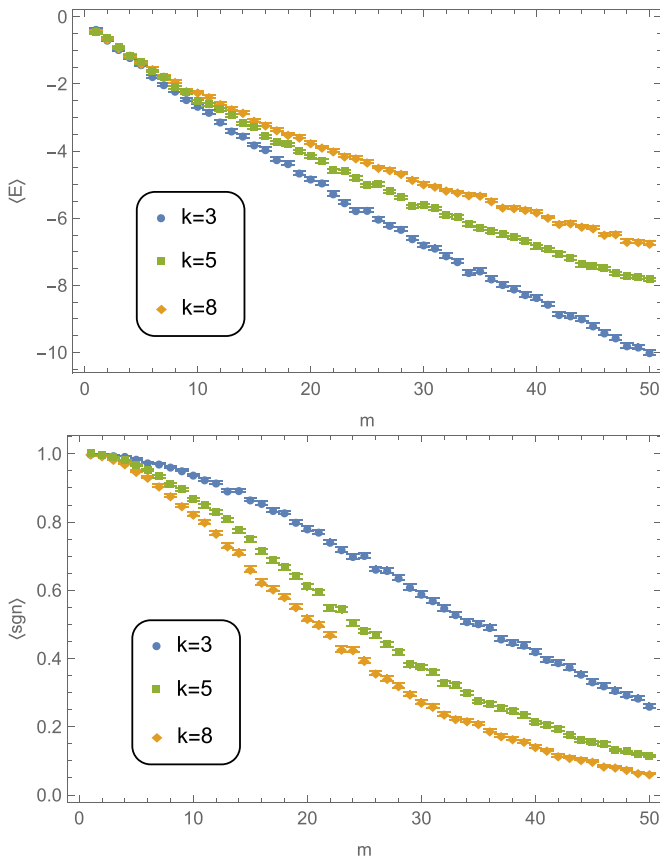
FIG. 5. Top: Average energy $\langle E \rangle$ over 200 randomly generated Hamiltonian instances as a function of m for random k-local 40-spin Hamiltonians for $k = 3$, $k = 5$, and $k = 8$. Bottom: A similar plot for $\langle \text{sgn} \rangle$, averaged over the 200 Hamiltonian instances. Here, $\beta = 1$.

FIG. 6. Top: Average energy $\langle E \rangle$ over 200 randomly generated Hamiltonian instances as a function of m for random k-local 40-spin Hamiltonians for $k = 3$, $k = 5$, and $k = 8$. Bottom: A similar plot for $\langle \text{sgn} \rangle$, averaged over the 200 Hamiltonian instances. Here, $\beta = 5$.

and k. Similarly, Fig. 6 depicts $\langle E \rangle$ and $\langle \text{sgn} \rangle$ as a function of m for $\beta = 5$.

E. Classically frustrated spin models

It is important to note that while our approach guarantees a correct equilibrium distribution of the Markov chain for any input spin-1/2 Hamiltonian, a universal rapid mixing of the Markov chain cannot be ensured in general.

One class of many-body systems that is known to considerably hinder the convergence of Monte Carlo algorithms—classical or quantum—is that of strongly frustrated spin models. For these, there exist competing terms in the Hamiltonians; the minimization of one directly conflicts with the minimization of others, creating "frustration." For such Hamiltonians, employing a Markov chain whose classical updates are based on single spin-flip Metropolis moves will result in slow mixing. This is because single spin-flip moves may cause the simulation to get trapped in metastable regions of configuration space (i.e., local minima).

In many cases, the slowdown caused by frustration can be significantly mitigated (but not fully cured in general due to the NP-hardness of the underlying problem) by replacing the classical Metropolis updates with suitable cluster updates if such exist [37–41]. Replacing classical Metropolis updates (see Sec. IV D 4) with relevant cluster updates often results
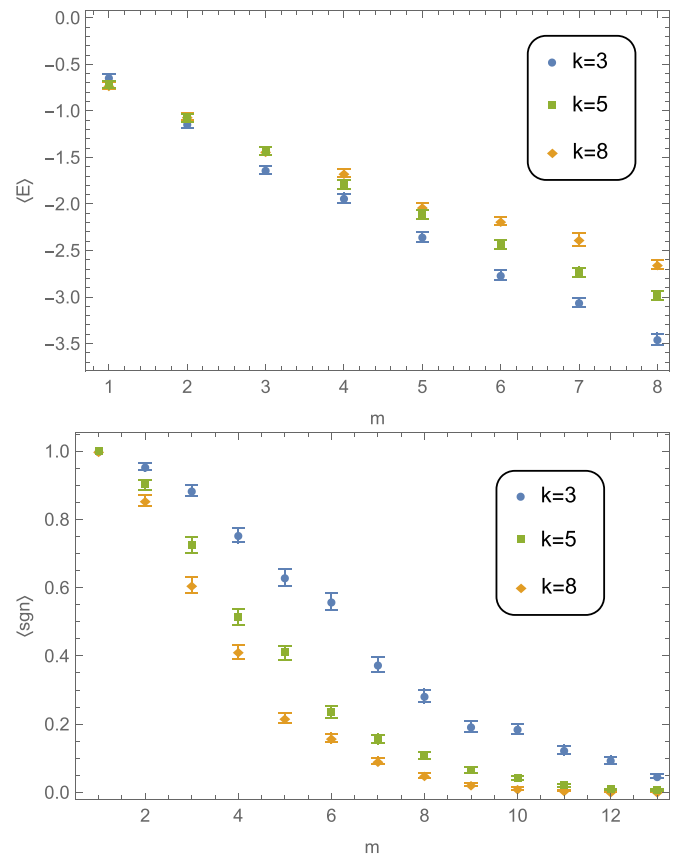
in a much faster converging algorithm.[3] Another efficient approach to some frustrated systems is to combine the QMC algorithm with parallel tempering [42,43] or population annealing [44–46].

A simple way to detect whether geometric frustration leads to a computational inefficiency in a Monte Carlo calculation is to compare the statistical error obtained from binning analysis (see Appendix B) with the statistical error estimated from multiple completely independent, and hence uncorrelated, runs of the algorithm. If the system is thermalized throughout the course of the simulation, these two estimates should roughly agree. For a highly frustrated system at a low temperature, for which the autocorrelation time is longer than the timescale of the simulation, the error obtained from the binning analysis is significantly underestimated due to the measurements being not fully decorrelated.

Figure 7 shows the computed mean energy as a function of β for a slightly frustrated model where only a small fraction of the edges are antiferromagnetic. The inset shows the error estimates from binning analysis and from independent

---

[3]These classical cluster moves can be generalized to quantum cluster moves for which the configuration weights are different from Boltzmann weights. However, such a generalization is beyond the scope of the present paper.
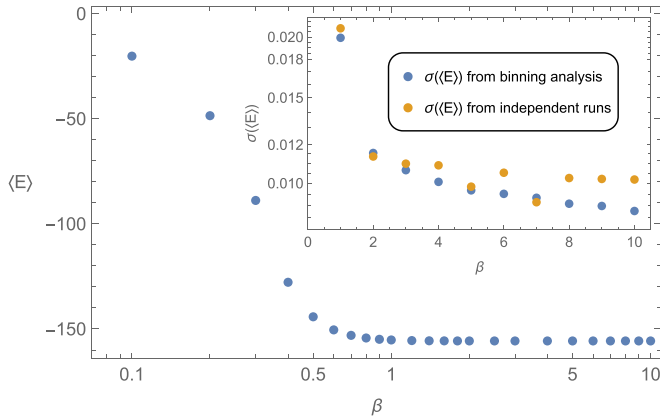
FIG. 7. Dependence of mean energy on β for the slightly frustrated XY model on a triangular lattice. Inset: Error estimation. Parameters: $L = 8$, $\Gamma = -0.25$, $J = 1$ for a few randomly selected edges, and $J = -1$ for the remaining edges.

runs. This example suggests that it is possible to compute the observables with high accuracy at low temperatures if the fraction of frustrated plaquettes is sufficiently low.

F. Code parallelization

Markov chain Monte Carlo algorithms are naturally well suited for massively parallel simulations, where independently run Markov chains contribute equally to the collection of statistics. As shown in Fig. 8, the algorithm, indeed, demonstrates a near-perfect strong scaling speedup, with the only restriction being that each of the parallel processes is expected to employ a sufficient number of initial thermalization (warmup) steps prior to measurement collection.

In addition to sequential execution of the C + + program code, our software package includes capabilities for executing the code in a parallel fashion on high-performance
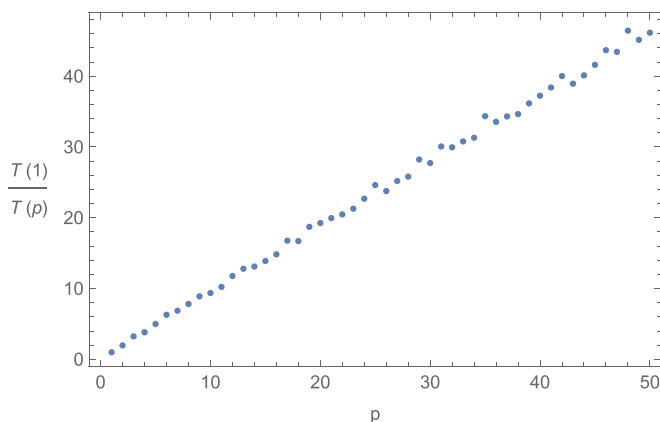

FIG. 8. Strong scaling speedup. Here, $T(p)$ is the time required to complete the same amount of work using the parallel setup with p processing threads. The total numbers of initial thermalization QMC updates and main QMC updates are equal to 2.5×10^7 and 2.5×10^8, respectively, for each of the parallel setups. Calculations were performed for the XY model on a square lattice with periodic boundary conditions with $L = 8$, $J = 1$, and $\Gamma = -0.25$.

compute clusters using message passing interface (MPI) protocols [47], allowing for extensive parallelization of the algorithm [15].

VII. SUMMARY AND OUTLOOK

We presented a universal, parameter-free, Trotter-error-free quantum Monte Carlo scheme capable of simulating, for the first time, arbitrary spin-1/2 Hamiltonians. We have demonstrated that the permutation matrix representation of Hamiltonians allows one to automatically produce QMC updates that are provably ergodic and satisfy detailed balance, thereby ensuring the convergence of the Markov chain to the proper thermal equilibrium. We have in addition illustrated how a wide range of observables may be calculated throughout the simulation.

Our algorithm therefore allows one to study the equilibrium properties of essentially any conceivable spin-1/2 system with a single piece of code that accepts as input a description of a Hamiltonian. This is in stark contrast to existing techniques, which generally require specially tailored model-specific QMC updates for each to-be-simulated system, and are thus limited to simulating models of very specified structures and geometries.

We believe that the generality and versatility of our approach make our proposed technique a very useful tool for condensed matter physicists studying spin systems, allowing the community to explore, with ease, an extremely wide range of physical models, many of which have so far been inaccessible, cumbersome to code, or too large to implement with existing techniques. To that aim, we have made our program code freely accessible on GitHub [15].

We note though that while our approach guarantees a correct equilibrium distribution of the Markov chain, the proposed algorithm does not guarantee a universal rapid mixing of the Markov chain, nor does it resolve or aim to resolve the sign problem.

The generality of the technique covered in this study makes it easily extendable to other types of systems, e.g., fermionic, bosonic, or higher spin systems. We intend to explore these in future work.

ACKNOWLEDGMENTS

APPENDIX A: CYCLE COMPLETION WITH GAPS

As was mentioned in the main text, a simple cycle completion protocol, in which consecutive elements of $S_{\mathbf{i}_q}$ form a subsequence S, may lead to a violation of the ergodicity condition and therefore also an incorrect calculation. The reason for this is the possibility of the resultant configuration having zero weight [as per Eq. (7)]. As a consequence, the fundamental

cycle completion routine may never be applied for some of the fundamental cycles during the Markov process, which can in turn lead to the ergodicity violation.

To resolve this issue, we have developed a protocol that we call "cycle completion with gaps." This protocol does not require the elements of the sequence S to form a consecutive unit within $S_{\mathbf{i}_q}$. A detailed description of this subroutine follows.

The parameters are $r_{\min}$, $r_{\max}$, $l_{\min}(r)$, and $l_{\max}(r)$. We usually choose $r_{\min} = (f_{\min} - 1)/2$, $r_{\max} = (f_{\max} + 1)/2$, $l_{\min}(r) = 2r - 1$, and $l_{\max}(r) = 2r + 1$, where $f_{\min}$ and $f_{\max}$ are minimal and maximal fundamental cycle lengths, respectively. The other option ("exhaustive search") is to choose $r_{\min} = 0$, $r_{\max} = f_{\max}$, $l_{\min}(r) = r$, and $l_{\max}(r) = f_{\max}$.

The sequence of operations is as follows:

(1) Pick a random integer u according to a geometric distribution p_u. As we will see, u is the total number of operators in the "gaps."

(2) If $q < u + r_{\min}$, then the update is rejected.

(3) Pick a random integer r such that $r_{\min} \leqslant r \leqslant \min(r_{\max}, q - u)$. We note that the probability $p_r(q) = (\min(r_{\max}, q - u) - r_{\min} + 1)^{-1}$ depends on q.

(4) Randomly pick a subsequence $\widetilde{S}$ of length $r + u$ containing consecutive operators from the sequence $S_{\mathbf{i}_q}$.

(5) Randomly choose a subsequence S of length r from $\widetilde{S}$. The remaining u operators in $\widetilde{S}$ we will call gaps.

(6) If S contains repeated operators, the update is rejected.

(7) Find all fundamental cycles of length l such that $l_{\min}(r) \leqslant l \leqslant l_{\max}(r)$, each containing all operators of the subsequence S. Denote by n_c the number of found cycles.

(8) If $n_c = 0$, the update is rejected. Otherwise, we randomly choose one of the found fundamental cycles.

(9) Attempt to replace the subsequence $\widetilde{S}$ of length $r + u$ by the sequence $\widetilde{S}'$ of length $r' + u$ which contains all the remaining r' operators from the selected cycle, as well as all the gaps. We shuffle the sequence $\widetilde{S}'$ so that its operators are contained in random order. We accept the update with the probability P_{accept}, which is considered below.

Let us now find the acceptance probability P_{accept} such that the detailed balance holds for the above protocol. Suppose that the u gaps contain u_i of operators P_i, where $i = 1, 2, \ldots, M$, so that $\sum_i u_i = u$. Let us denote the old and new configurations as A and B, probability to select B from A as $P_{\text{select}}(A \rightarrow B)$, and probability to select A from B as $P_{\text{select}}(B \rightarrow A)$. Then, we have

$$
P_{\text{select}}(A \rightarrow B)
$$
$$
= p_u\, p_r(q)(q - (r + u) + 1)^{-1}
$$
$$
\times \frac{1}{n_c}\binom{r+u}{u}^{-1}\binom{r'+u}{u}^{-1}\frac{1}{r'!}\frac{u_1! \cdots u_M!}{u!}, \quad \text{(A1)}
$$

$$
P_{\text{select}}(B \rightarrow A)
$$
$$
= p_u\, p_r(q')(q' - (r' + u) + 1)^{-1}
$$
$$
\times \frac{1}{n'_c}\binom{r'+u}{u}^{-1}\binom{r+u}{u}^{-1}\frac{1}{r!}\frac{u_1! \cdots u_M!}{u!}. \quad \text{(A2)}
$$

Here, n'_c is the number of fundamental cycles of length l such that $l_{\min}(r') \leqslant l \leqslant l_{\max}(r')$, each containing all r' operators of

the subsequence S'. Since $q' = q + r' - r$, we have $q - (r + u) + 1 = q' - (r' + u) + 1$. Therefore,

$$
P_{\text{accept}}(A \rightarrow B) = \min\left(1, \frac{W_B}{W_A}\frac{P_{\text{select}}(B \rightarrow A)}{P_{\text{select}}(A \rightarrow B)}\right)
$$
$$
= \min\left(1, \frac{W_B}{W_A}\frac{p_r(q')}{p_r(q)}\frac{n_c}{n'_c}\frac{r'!}{r!}\right). \quad \text{(A3)}
$$

Here, W_A and W_B are the weights of the old and the new operator sequences. Because $P(A \rightarrow B) = P_{\text{select}}(A \rightarrow B)P_{\text{accept}}(A \rightarrow B)$ and $P(B \rightarrow A) = P_{\text{select}}(B \rightarrow A)P_{\text{accept}}(B \rightarrow A)$, Eq. (A3) satisfies the detailed balance condition.

APPENDIX B: ESTIMATION OF THE STATISTICAL ERRORS

It is known that statistical errors in a Monte Carlo calculation can be estimated by employing binning analysis [48,49]. By grouping N measurements of an observable $\mathcal{O}$ into n_B nonoverlapping blocks of length $B = N/n_B$, one first obtains a single data point for each of the n_B bins as follows:

$$
\mathcal{O}_i^{(B)} = \frac{1}{B}\sum_{j=(i-1)B+1}^{iB}\mathcal{O}_j, \quad i = 1, \ldots, n_B. \quad \text{(B1)}
$$

If the bins are large enough, the averages $\mathcal{O}_i^{(B)}$ will be effectively uncorrelated, and one could then use the simple (uncorrelated) variance estimator to find the variance of the mean,

$$
\sigma^2(\langle\mathcal{O}\rangle) = \frac{1}{n_B(n_B - 1)}\sum_{i=1}^{n_B}\left(\mathcal{O}_i^{(B)} - \langle\mathcal{O}\rangle\right)^2, \quad \text{(B2)}
$$

where $\langle\mathcal{O}\rangle = \langle\mathcal{O}^{(B)}\rangle = \sum_{j=1}^{n_B}\mathcal{O}_j^{(B)}/n_B$. Similarly, the covariance estimator of the mean values of two observables $\mathcal{O}$ and $\mathcal{Q}$ is

$$
\text{cov}(\langle\mathcal{O}\rangle, \langle\mathcal{Q}\rangle)
$$
$$
= \frac{1}{n_B(n_B - 1)}\sum_{i=1}^{n_B}\left(\mathcal{O}_i^{(B)} - \langle\mathcal{O}\rangle\right)\left(\mathcal{Q}_i^{(B)} - \langle\mathcal{Q}\rangle\right). \quad \text{(B3)}
$$

Since $\langle A\rangle$ is the ratio [Eq. (41)] of two random variables, the improved approximation for $\langle A\rangle$ and the approximation for $\sigma^2(A)$ are as follows [50]:

$$
\langle A\rangle = \frac{\langle A \cdot \text{sgn}\rangle}{\langle\text{sgn}\rangle}\left(1 + \frac{\sigma^2(\langle\text{sgn}\rangle)}{\langle\text{sgn}\rangle^2}\right) - \frac{\text{cov}(\langle\text{sgn}\rangle, \langle A \cdot \text{sgn}\rangle)}{\langle\text{sgn}\rangle^2},
$$
$$
\text{(B4)}
$$

and

$$
\sigma^2(A) = \frac{\langle A \cdot \text{sgn}\rangle^2}{\langle\text{sgn}\rangle^2}\left(\frac{\sigma^2(\langle A \cdot \text{sgn}\rangle)}{\langle A \cdot \text{sgn}\rangle^2} + \frac{\sigma^2(\langle\text{sgn}\rangle)}{\langle\text{sgn}\rangle^2}\right.
$$
$$
\left. - 2\frac{\text{cov}(\langle\text{sgn}\rangle, \langle A \cdot \text{sgn}\rangle)}{\langle\text{sgn}\rangle\langle A \cdot \text{sgn}\rangle}\right). \quad \text{(B5)}
$$

Here, the values of $\sigma^2(\langle\text{sgn}\rangle)$, $\sigma^2(\langle A \cdot \text{sgn}\rangle)$, and $\text{cov}(\langle\text{sgn}\rangle, \langle A \cdot \text{sgn}\rangle)$ should be obtained via Eqs. (B2) and (B3).

[1] J. Gubernatis, N. Kawashima, and P. Werner, *Quantum Monte Carlo Methods: Algorithms for Lattice Models* (Cambridge University Press, Cambridge, UK, 2016).

[2] D. Landau and K. Binder, *A Guide to Monte Carlo Simulations in Statistical Physics*, 5th ed. (Cambridge University Press, Cambridge, UK, 2021).

[3] A. Lüchow, Quantum Monte Carlo methods, WIREs Comput. Mol. Sci. **1**, 388 (2011).

[4] A. Aspuru-Guzik, A. D. Dutoi, P. J. Love, and M. Head-Gordon, Simulated quantum computation of molecular energies, Science **309**, 1704 (2005).

[5] I. Kassal, S. P. Jordan, P. J. Love, M. Mohseni, and A. Aspuru-Guzik, Polynomial-time quantum algorithm for the simulation of chemical dynamics, Proc. Natl. Acad. Sci. USA **105**, 18681 (2008).

[6] B. P. Lanyon, J. D. Whitfield, G. G. Gillett, M. E. Goggin, M. P. Almeida, I. Kassal, J. D. Biamonte, M. Mohseni, B. J. Powell, M. Barbieri, A. Aspuru-Guzik, and A. G. White, Towards quantum chemistry on a quantum computer, Nat. Chem. **2**, 106 (2010).

[7] D. Lonardoni, F. Pederiva, and S. Gandolfi, From hypernuclei to the inner core of neutron stars: A quantum Monte Carlo study, J. Phys.: Conf. Ser. **529**, 012012 (2014).

[8] S. Chandrasekharan and U.-J. Wiese, Meron-cluster solution of fermion sign problems, Phys. Rev. Lett. **83**, 3116 (1999).

[9] T. D. Kieu and C. J. Griffin, Monte Carlo simulations with indefinite and complex-valued measures, Phys. Rev. E **49**, 3855 (1994).

[10] A. W. Sandvik, A generalization of Handscomb's quantum Monte Carlo scheme—application to the 1-D Hubbard model, J. Phys. A: Math. Gen. **25**, 3667 (1992).

[11] A. W. Sandvik, Stochastic series expansion method with operator-loop update, Phys. Rev. B **59**, R14157 (1999).

[12] 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. Exp. Theor. Phys. **87**, 310 (1998).

[13] M. Iazzi and M. Troyer, Efficient continuous-time quantum Monte Carlo algorithm for fermionic lattice models, Phys. Rev. B **91**, 241118(R) (2015).

[14] Z.-X. Li, Y.-F. Jiang, and H. Yao, Solving the fermion sign problem in quantum Monte Carlo simulations by Majorana representation, Phys. Rev. B **91**, 241117(R) (2015).

[15] Permutation matrix representation quantum Monte Carlo for arbitrary spin-1/2 Hamiltonians: program code in C++, https://github.com/LevBarash/PMRQMC

[16] L. Gupta, T. Albash, and I. Hen, Permutation matrix representation quantum Monte Carlo, J. Stat. Mech. (2020) 073105.

[17] T. Albash, G. Wagenbreth, and I. Hen, Off-diagonal expansion quantum Monte Carlo, Phys. Rev. E **96**, 063309 (2017).

[18] I. Hen, Off-diagonal series expansion for quantum partition functions, J. Stat. Mech. (2018) 053102.

[19] D. Joyner, *Adventures in Group Theory: Rubik's Cube, Merlin's Machine, and Other Mathematical Toys* (Johns Hopkins University Press, Baltimore, 2008).

[20] J. A. Bondy and U. S. R. Murty, *Graph Theory with Applications* (North Holland, Amsterdam, 1976).

[21] E. T. Whittaker and G. Robinson, in *The Calculus of Observations: A Treatise on Numerical Mathematics* (Blackie and Son, London, 1924).

[22] C. de Boor, Divided differences, Surv. Approximation Theory **1**, 46 (2005).

[23] M. Newman and G. Barkema, *Monte Carlo Methods in Statistical Physics* (Clarendon Press, Oxford, UK, 1999).

[24] G. Grimmett and D. Stirzaker, *Probability and Random Processes*, 4th ed. (Oxford University Press, Oxford, UK, 2020).

[25] Çetin K. Koç and S. N. Arachchige, A fast algorithm for gaussian elimination over GF(2) and its implementation on the GAPP, J. Parallel Distrib. Comput. **13**, 118 (1991).

[26] L. Gupta, L. Barash, and I. Hen, Calculating the divided differences of the exponential function by addition and removal of inputs, Comput. Phys. Commun. **254**, 107385 (2020).

[27] E. Akaturk, L. Gupta, and I. Hen, Permutation matrix representation quantum Monte Carlo: Advanced measurement techniques (unpublished).

[28] I. Hen, Determining quantum Monte Carlo simulability with geometric phases, Phys. Rev. Res. **3**, 023080 (2021).

[29] M. Troyer and U.-J. Wiese, Computational complexity and fundamental limitations to fermionic quantum Monte Carlo simulations, Phys. Rev. Lett. **94**, 170201 (2005).

[30] L. Gupta and I. Hen, Elucidating the interplay between nonstoquasticity and the sign problem, Adv. Quantum Technol. **3**, 1900108 (2020).

[31] Y.Q. Ma and W. Figueiredo, Phase diagram of the anisotropic XY model, Phys. Rev. B **55**, 5604 (1997).

[32] A. Pires, Phase diagram for the two-dimensional quantum anisotropic XY model, J. Magn. Magn. Mater. **260**, 397 (2003).

[33] P. C. Hohenberg and B. I. Halperin, Theory of dynamic critical phenomena, Rev. Mod. Phys. **49**, 435 (1977).

[34] A. Kitaev, Fault-tolerant quantum computation by anyons, Ann. Phys. **303**, 2 (2003).

[35] B. Field and T. Simula, Introduction to topological quantum computation with non-Abelian anyons, Quantum Sci. Technol. **3**, 045004 (2018).

[36] C. D. Freeman, C. M. Herdman, D. J. Gorman, and K. B. Whaley, Relaxation dynamics of the toric code in contact with a thermal reservoir: Finite-size scaling in a low-temperature regime, Phys. Rev. B **90**, 134302 (2014).

[37] D. Kandel, R. Ben-Av, and E. Domany, Cluster dynamics for fully frustrated systems, Phys. Rev. Lett. **65**, 941 (1990).

[38] G. M. Zhang and C. Z. Yang, Cluster Monte Carlo dynamics for the antiferromagnetic Ising model on a triangular lattice, Phys. Rev. B **50**, 12546 (1994).

[39] P. D. Coddington and L. Han, Generalized cluster algorithms for frustrated spin models, Phys. Rev. B **50**, 3058 (1994).

[40] J. Houdayer, A cluster Monte Carlo algorithm for 2-dimensional spin glasses, Eur. Phys. J. B **22**, 479 (2001).

[41] G. Rakala and K. Damle, Cluster algorithms for frustrated two-dimensional Ising antiferromagnets via dual worm constructions, Phys. Rev. E **96**, 023304 (2017).

[42] R. H. Swendsen and J.-S. Wang, Replica Monte Carlo simulation of spin-glasses, Phys. Rev. Lett. **57**, 2607 (1986).

[43] K. Hukushima and K. Nemoto, Exchange Monte Carlo method and application to spin glass simulations, J. Phys. Soc. Jpn. **65**, 1604 (1996).

[44] J. Machta, Population annealing with weighted averages: A Monte Carlo method for rough free-energy landscapes, Phys. Rev. E **82**, 026704 (2010).

[45] L. Y. Barash, M. Weigel, M. Borovský, W. Janke, and L. N. Shchur, GPU accelerated population annealing algorithm, Comput. Phys. Commun. **220**, 341 (2017).

[46] L. Barash, J. Marshall, M. Weigel, and I. Hen, Estimating the density of states of frustrated spin systems, New J. Phys. **21**, 073065 (2019).

[47] F. Nielsen, Introduction to MPI: The message passing interface, in *Introduction to HPC with MPI for Data Science* (Springer International Publishing, Cham, 2016), pp. 21–62.

[48] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap* (Chapman and Hall, Boca Raton, FL, 1994).

[49] W. Janke, Monte Carlo methods in classical statistical physics, in *Computational Many-Particle Physics*, edited by H. Fehske, R. Schneider, and A. Weiße (Springer, Berlin, 2008), pp. 79–140.

[50] A. Stuart and K. Ord, *Kendall's Advanced Theory of Statistics, Volume 1, Distribution Theory*, 6th ed. (John Wiley & Sons, New York, 1998).