

**Off-diagonal expansion quantum Monte Carlo**Tameem Albash,<sup>1,2</sup> Gene Wagenbreth,<sup>3</sup> and Itay Hen<sup>1,2,\*</sup><sup>1</sup>*Information Sciences Institute, University of Southern California, Marina del Rey, California 90292, USA*<sup>2</sup>*Department of Physics and Astronomy and Center for Quantum Information Science & Technology, University of Southern California, Los Angeles, California 90089, USA*<sup>3</sup>*Cray, Seattle, Washington 98164, USA*

(Received 17 January 2017; revised manuscript received 6 November 2017; published 20 December 2017)

We propose a Monte Carlo algorithm designed to simulate quantum as well as classical systems at equilibrium, bridging the algorithmic gap between quantum and classical thermal simulation algorithms. The method is based on a decomposition of the quantum partition function that can be viewed as a series expansion about its classical part. We argue that the algorithm not only provides a theoretical advancement in the field of quantum Monte Carlo simulations, but is optimally suited to tackle quantum many-body systems that exhibit a range of behaviors from “fully quantum” to “fully classical,” in contrast to many existing methods. We demonstrate the advantages, sometimes by orders of magnitude, of the technique by comparing it against existing state-of-the-art schemes such as path integral quantum Monte Carlo and stochastic series expansion. We also illustrate how our method allows for the unification of quantum and classical thermal parallel tempering techniques into a single algorithm and discuss its practical significance.

DOI: [10.1103/PhysRevE.96.063309](https://doi.org/10.1103/PhysRevE.96.063309)**I. INTRODUCTION**

Quantum Monte Carlo (QMC) simulations [1,2] are essentially the only method for studying equilibrium properties of large quantum many-body systems without approximations in more than one dimension.<sup>1</sup> These methods have become key to understanding pertinent macroscopic quantum physical phenomena in numerous fields from superconductivity and novel quantum materials to the physics of neutron stars and quantum chromodynamics, making efficient QMC schemes a necessary tool more than ever.

Over the years, novel QMC techniques have been devised with the goal of speeding up convergence rates, thereby allowing the study of ever larger system sizes. One notable example is the shift from discrete-time techniques to continuous time for which Trotter discretization errors are absent [4–7]. For many quantum many-body systems, however, convergence can be so slow that the simulation is considered virtually intractable regardless of which method is used. First and foremost in that regard is the negative sign problem [8,9], which arises whenever the partition function of the system cannot be decomposed into positive-valued Monte Carlo weights.

QMC algorithms are also known to be notoriously inefficient in describing certain models that do not suffer from the sign problem. A notable example is quantum spin glasses in “almost classical” parameter regimes [10–12] where updates resulting from thermal fluctuations are expected to be far more dominant than those resulting from quantum fluctuations. These models can be parametrically tuned from a quantum ordered phase on one end to a classical disordered phase at the other end. Since QMC methods normally evolve via configuration updates that are based on quantum fluctuations, the acceptance rates of quantum updates, e.g., single spin (or

cluster) flips in the dual Ising system, decrease dramatically in classical regimes, often causing QMC algorithms to dramatically slow down or “freeze” (see, e.g., Ref. [13]).

Efficient classical thermal updates are typically hard to implement within the framework of QMC algorithms because these algorithms do not normally converge to classical Monte Carlo algorithms in the limit where the model becomes classical. For this reason, there are almost no algorithms that efficiently simulate systems that exhibit the full range of behavior from being “fully quantum” to “fully classical.” For the successful simulation of systems exhibiting the above characteristics, it is therefore important to devise Monte Carlo schemes that can function both as quantum as well as classical algorithms when necessary. Efficient methods of this type will have wide-ranging applicability in diverse areas ranging from statistical physics through quantum chemistry to quantum computing and beyond.

Here, we propose an algorithm that has the algorithmic flexibility to simulate interacting many-body systems ranging from the fully quantum to the fully classical. We present a Monte Carlo scheme that is based on a decomposition of the canonical quantum partition function into a sum of Boltzmann-type weights and that converges to the usual decomposition of the classical partition function in the limit where the Hamiltonian of the system becomes classical. Based on this unique decomposition, our algorithm aims to improve the convergence rates of simulated systems for which existing techniques are often inefficient.

Within our approach, the quantum imaginary-time dimension of the algorithm is “elastic,” i.e., it can stretch or shrink dynamically depending on the strength of the quantum part of the system: the off-diagonal portion of the Hamiltonian. In addition, the proposed method does not introduce Trotter-type errors as in discrete-time path-integral QMC, a source of errors that normally occurs from an insufficient discretization of the imaginary-time dimension (overdiscretization tends to sharply reduce the acceptance rates of the QMC updates). Moreover, in the classical limit where off-diagonal terms

\*itayhen@isi.edu

<sup>1</sup>In one dimension, the density matrix renormalization group has proven to be an extremely powerful tool [3].

vanish, our algorithm naturally reduces to a classical thermal simulation. As we illustrate, the above properties allow our method to naturally overcome certain inefficiencies typically encountered by other QMC algorithms.

The structure of the paper is as follows. In Sec. II, we describe the decomposition of the canonical quantum partition function into what we refer to as generalized Boltzmann weights (GBWs). We then proceed in Sec. III to present the basic updates and measurements of our off-diagonal expansion (ODE) quantum Monte Carlo algorithm that builds on the above decomposition. To illustrate the practicality of our algorithm, we examine in Sec. IV simulations of the transverse field Ising model, especially inside the spin-glass phase, where the model is known to be hard to simulate. We demonstrate the advantages, sometimes by orders of magnitude, of our scheme over existing state-of-the-art QMC methods such as discrete-time path integral quantum Monte Carlo with cluster updates along the imaginary-time direction (PIQMC) [14] and stochastic series expansion (SSE) [7,15,16]. We also discuss in this context the unification of quantum and classical parallel tempering. We present some conclusions in Sec. V.

## II. GENERALIZED BOLTZMANN WEIGHTS

### A. Decomposition of the partition function

The main insight at the heart of our approach is a decomposition of the canonical quantum partition function which, as we argue, allows for the development of a QMC algorithm that has certain advantages over existing methods. Our work builds in part on the stochastic series expansion (SSE) algorithm, a well-known and successful QMC algorithm pioneered by Sandvik [7,15,16], which, unlike the traditional slicing of the partition function into Trotter segments, involves a Taylor series expansion in the inverse temperature  $\beta = 1/T$  (in our units  $k_B = 1$ ) of the partition function as was originally suggested by Handscomb [17,18].

The canonical quantum partition function of a system described by a Hamiltonian  $H$  is given by  $Z = \text{Tr}[e^{-\beta H}]$ . Our decomposition begins by first writing the Hamiltonian in the form

$$H = H_c - \sum_j \Gamma_j V_j. \quad (1)$$

Here,  $H_c$  is the classical portion of the Hamiltonian, i.e., a diagonal operator in some known basis, which we refer to as the computational basis, and whose basis states will be denoted by  $\{|z\rangle\}$ . The  $\{\Gamma_j\}$  are generally complex-valued parameters, and  $\{V_j\}$  are off-diagonal operators satisfying  $[V_j, H_c] \neq 0$  that give the system its “quantum dimension.” In an analogous way to standard SSE, in order for the decomposition of the partition function to be feasible, we require the off-diagonal operators to be chosen such that they obey  $V_j|z\rangle = |z'\rangle$  for every basis state  $|z\rangle$ , where  $|z'\rangle \neq |z\rangle$  is also a basis state. For simplicity, we henceforth assume that all the  $\Gamma_j$  parameters are identical, namely, that  $\Gamma_j = \Gamma, \forall j$ , however, as will become evident shortly this restriction is by no means necessary.

We now present the main steps for the decomposition of the quantum partition function. We first replace the trace operation  $\text{Tr}[\cdot]$  with the explicit sum  $\sum_z \langle z| \cdot |z\rangle$  and then expand the

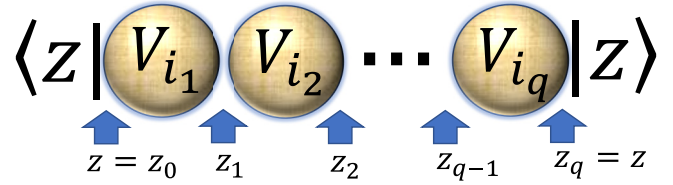


FIG. 1. A diagrammatic representation of the term  $\langle z|S_q|z\rangle$ . The sequence of operators  $S_q = V_{i_1}V_{i_2}\dots V_{i_q}$  is sandwiched between classical bra  $\langle z|$  and ket  $|z\rangle$ , inducing a sequence of classical states  $(|z_0\rangle, \dots, |z_q\rangle)$ . The multiset of classical energies of the states  $|z_i\rangle$ , namely,  $E_i = E(z_i) = \langle z_i|H_c|z_i\rangle$ , generate the generalized Boltzmann weight.

exponent in the partition function in a Taylor series:

$$\begin{aligned} Z &= \sum_z \sum_{n=0}^{\infty} \frac{\beta^n}{n!} \langle z|(-H)^n|z\rangle \\ &= \sum_z \sum_{n=0}^{\infty} \frac{\beta^n}{n!} \langle z| \left( -H_c + \Gamma \sum_j V_j \right)^n |z\rangle \\ &= \sum_z \sum_{n=0}^{\infty} \sum_{\{S_n\}} \frac{\beta^n}{n!} \langle z|S_n|z\rangle, \end{aligned} \quad (2)$$

where in the last step we have also expanded  $(-H)^n$ , and  $\{S_n\}$  denotes the set of all sequences of length  $n$  composed of products of basic operators  $H_c$  and  $V_j$ .

We proceed by stripping all the diagonal Hamiltonian terms off the sequences  $\langle z|S_n|z\rangle$ . We do so by evaluating their action on the relevant basis states, leaving only the off-diagonal operators unevaluated inside the sequence. At this point, the partition function can be written as

$$\begin{aligned} Z &= \sum_z \sum_{q=0}^{\infty} \sum_{\{S_q\}} \Gamma^q \langle z|S_q|z\rangle \left( \sum_{n=q}^{\infty} \frac{\beta^n (-1)^{n-q}}{n!} \right. \\ &\quad \left. \times \sum_{\sum k_i = n-q} E^{k_0}(z_0) \dots E^{k_q}(z_q) \right), \end{aligned} \quad (3)$$

where  $E(z_i) = \langle z_i|H_c|z_i\rangle$  and  $\{S_q\}$  denotes the set of all sequences of length  $q$  of “bare” off-diagonal operators  $V_j$ . The term in large parentheses sums over the diagonal contribution of all  $\langle z|S_n|z\rangle$  terms that correspond to a single  $\langle z|S_q|z\rangle$  term. The various  $\{|z_i\rangle\}$  states are the states obtained from the action of the ordered  $V_j$  operators in the sequence  $S_q$  on  $|z_0\rangle$ , then on  $|z_1\rangle$ , and so forth.<sup>2</sup> Figure 1 gives a schematic representation of  $\langle z|S_q|z\rangle$ .

Remarkably, the infinite sum over energies in Eq. (3) may be simplified to an easily calculated Boltzmann-type weight using the so-called divided differences representation [19,20] of the sum (the reader is referred to Appendix A for more details and a full derivation). This allows us to write the partition function

<sup>2</sup>For example, for  $S_q = V_{i_q} \dots V_{i_2} V_{i_1}$ , we obtain  $|z_0\rangle = |z\rangle, V_{i_1}|z_0\rangle = |z_1\rangle, V_{i_2}|z_1\rangle = |z_2\rangle$ , etc.

in terms of classical “effective energies” as

$$Z = \sum_{\{z\}} \sum_q \sum_{\{S_q\}} \frac{(\beta\Gamma)^q}{q!} \langle z|S_q|z \rangle e^{-\beta E_{(0,\dots,q)}}, \quad (4)$$

where  $E_{(0,\dots,q)}$  is the “effective classical energy” of the configuration calculated from the multiset of energies  $[E_0, \dots, E_q]$  generated by the term  $\langle z|S_q|z \rangle$  and from  $\beta$ . The effective classical energy is a single real-valued number  $E_{(0,\dots,q)} \in (\min[E_0, \dots, E_q], \max[E_0, \dots, E_q])$ . Specifically,  $E_{(0,\dots,q)}$  lies within the spectrum of the classical Hamiltonian. To calculate  $E_{(0,\dots,q)}$ , one may use the divided differences recursion relation (see Appendix B) which in terms of effective classical energies becomes

$$\frac{(-\beta)^q}{q!} e^{-\beta E_{(0,\dots,q)}} = \frac{(-\beta)^{q-1}}{(q-1)!} \frac{(e^{-\beta E_{(0,\dots,q-1)}} - e^{-\beta E_{(1,\dots,q)}})}{E_0 - E_q}. \quad (5)$$

Isolating  $E_{(0,\dots,q)}$ , we arrive at

$$E_{(0,\dots,q)} = \bar{E} - \frac{1}{\beta} \log \frac{2q \sinh \beta \Delta E}{\beta(E_q - E_0)}, \quad (6)$$

where  $2\bar{E} = E_{(1,\dots,q)} + E_{(0,\dots,q-1)}$  and  $2\Delta E = E_{(1,\dots,q)} - E_{(0,\dots,q-1)}$ . In the limiting case where all energies in the sequence are equal, the above relation neatly becomes  $E_{(0,\dots,q)} = E_{(0)} = E_0$ . The initial condition for the above recursion is simply  $e^{-\beta E_{(i)}} = e^{-\beta E_i}$ . An illustration of how the recursion relation is used to calculate  $E_{(0,\dots,q)}$  is depicted in Fig. 2 and is discussed in more detail in Appendix C.

Finally, since by construction the term  $\langle z|S_q|z \rangle$  evaluates to either 0 or to 1 (the operation  $S_q|z \rangle$  returns a basis state  $|z' \rangle$  and therefore  $\langle z|S_q|z \rangle = \langle z|z' \rangle = \delta_{z,z'}$ ), the partition function can be neatly presented in its final form as a sum over only nonvanishing terms:

$$Z = \sum_{\{S_q: \langle z|S_q|z \rangle = 1\}} \frac{(\beta\Gamma)^q}{q!} e^{-\beta E_{(0,\dots,q)}}. \quad (7)$$

We interpret the terms in the sum in Eq. (7) as weights, i.e.,  $Z = \sum_{\{C\}} W_C$ , where the set of configurations  $\{C\}$  is all the distinct pairs  $\{|z\rangle, S_q\}$ . Because of the form of  $W_C$ ,

$$W_C = \frac{(\beta\Gamma)^q}{q!} e^{-\beta E_{(0,\dots,q)}}, \quad (8)$$

we refer to it as a generalized Boltzmann weight (or GBW). We shall refer to  $E_{(0,\dots,q)}$  as the “effective classical energy” of the configuration  $C$  and denote it at times for brevity simply by  $E_C$ .

In order to interpret the  $W_C$  terms as actual weights, they must be non-negative for any simulated system that is not plagued by the sign problem [2]. It is therefore interesting to note that the above weights are automatically positive if  $\Gamma$  is positive, i.e., if the off-diagonal elements are nonpositive, which is the case for stoquastic Hamiltonians [21,22]. As is also evident from the above expression, even values of  $q$  also yield positive weights regardless of the sign of  $\Gamma$ . This corresponds to a scenario where off-diagonal operators must be injected along the imaginary-time dimension in even numbers in order to ensure nonzero weights. One such example is the

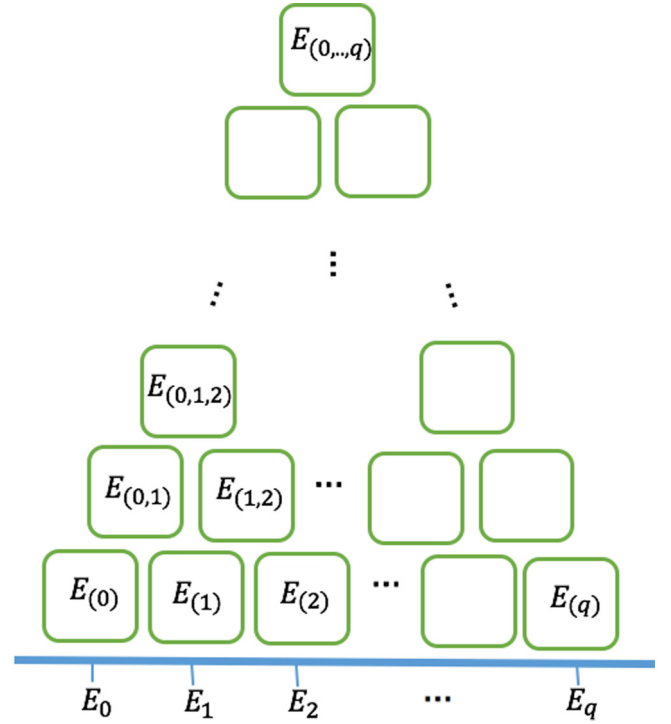


FIG. 2. Calculating the effective classical energy of the generalized Boltzmann weight using a “pyramid” structure. The evaluation of the divided differences of the exponential function of  $q + 1$  input energies consists of calculating each level of the pyramid starting at its base. The values at the base of the pyramid  $E_{(j)}$  are simply the energy inputs  $E_j$  (shown as the blue line at the bottom of the pyramid), with all identical energies placed together as a group (the exact ordering of the energies is not important). To calculate the elements at the next level of the pyramid, we use the relation in Eq. (6). This procedure is continued until the final level of the pyramid is evaluated, which corresponds to the effective classical energy in the GBW, namely,  $E_{(0,\dots,q)}$ .

transverse-field Ising Hamiltonian

$$H = \sum_{(i,j)} J_{ij} \sigma_i^z \sigma_j^z + \sum_j h_j \sigma_j^z - \Gamma \sum_j \sigma_j^x, \quad (9)$$

where  $H_c = \sum_{(i,j)} J_{ij} \sigma_i^z \sigma_j^z + \sum_j h_j \sigma_j^z$  and the off-diagonal operators are the spin-flip terms  $V_j = \sigma_j^x$ . In order for the  $\langle z|S_q|z \rangle$  terms to evaluate to one rather than to zero, off-diagonal operators must always be produced and annihilated in pairs, implying that the total sign of the weight [Eq. (8)] is positive. We have thus established a decomposition of the canonical quantum partition function into a sum of positive-valued weights.

## B. Properties of the GBWs

One property of the above decomposition of the canonical quantum partition function is that it may be viewed as unifying the classical and quantum partition functions. Specifically, it contains as a subsum the classical partition function decomposition of its diagonal part  $H_c$ . Writing the quantum partition function as a series in the “quantum strength” parameter  $\Gamma$ , one obtains the classical partition function as the zeroth

term, namely,

$$Z = \sum_{\{S_0: \langle z|z \rangle=1\}} e^{-\beta E(z)} + \frac{(\beta\Gamma)^2}{2} \sum_{\{S_2: \langle z|S_2|z \rangle=1\}} e^{-\beta E_{(0,1,2)}} + \dots \quad (10)$$

Furthermore, in classical regimes where  $\Gamma$  is zero or very small, the dominant configurations, i.e., those with highest weights, have no off-diagonal terms, and only the  $q = 0$  terms survive. In this case, the typical weights are

$$\frac{(\beta\Gamma)^q}{q!} e^{-\beta E_{(0,\dots,q)}} \Big|_{q=0} = e^{-\beta E_{(0)}} = e^{-\beta E(z)}, \quad (11)$$

where  $E(z)$  is the classical energy of the spin configuration  $z$ . Our decomposition thus automatically reduces to the usual sum over Boltzmann weight of classical Hamiltonians.<sup>3</sup>

The GBW [Eq. (8)] also has several attractive properties that make it useful for Monte Carlo simulations. First, as was already mentioned, it is strictly positive for stoquastic systems. This feature automatically resolves the ‘‘diagonal sign problem’’ that sometimes appears in other schemes [23], where constants must be added to the diagonal bonds to rectify the problem. Moreover, since the addition of such constants considerably affects the convergence rate of the algorithm [5,7,15,16,23], these constants usually have to be optimized for faster convergence. A QMC algorithm based on the GBW decomposition is in this respect parameter free, a property that is expected to facilitate computations.

Second, any arbitrary energy shift  $\Delta E$  of the diagonal part of the Hamiltonian has a trivial effect on the effective classical energy  $E_{(0,\dots,q)} \rightarrow E_{(0,\dots,q)} + \Delta E$ , reflecting that the addition of constants to the simulated Hamiltonian leaves ratios of weights, which in turn determine the acceptance rates of the QMC updates, invariant as they should be.

On a more academic note, it is interesting to observe that the proposed algorithm also has close relations to continuous-time QMC methods (e.g., Ref. [6]), via the Hermite-Genocchi formula [20]

$$e^{-\beta[E_0,\dots,E_q]} = \int_{\Omega} dt_0 \dots dt_q e^{-\beta(E_0 t_0 + E_1 t_1 + \dots + E_q t_q)}, \quad (12)$$

where  $t_i \geq 0$  and the area of integration  $\Omega$  is bounded by  $t_0 + t_1 + \dots + t_q$  from above.

### C. A simple analytical example

As a first illustration of the above decomposition, let us calculate the partition function of a transverse-field Ising Hamiltonian in the case where the classical Ising part vanishes, namely, where  $H_c = 0$ . In this case, the model becomes the trivial system  $H = -\Gamma \sum_i \sigma_i^x$ . The partition function in this special case is decomposed as

$$Z = \sum_{\{S_q: \langle z|S_q|z \rangle=1\}} \frac{(\beta\Gamma)^q}{q!}, \quad (13)$$

where the classical energies are  $E_0 = \dots = E_q = 0$ , corresponding to  $E_{(0,\dots,q)} = 0$ . In this case, we have  $\langle z|S_q|z \rangle = 1$  if and only if all  $\sigma_i^x$  off-diagonal operators in  $S_q$  appear an even number of times. Denoting by  $N_p(q)$  the number of nonzero weights for each value of (even)  $q$  and every  $|z\rangle$ , the partition function can be simplified to

$$Z = 2^N \sum_{q \text{ even}} \frac{(\beta\Gamma)^q}{q!} N_p(q). \quad (14)$$

A simple calculation (see Appendix D) reveals

$$N_p(q) = \frac{1}{2^N} \sum_{k=0}^N \binom{N}{k} (N-2k)^q, \quad (15)$$

which yields

$$\begin{aligned} Z &= \sum_q \frac{(\beta\Gamma)^q}{q!} \sum_{k=0}^N \binom{N}{k} (N-2k)^q \\ &= \sum_{k=0}^N \binom{N}{k} \sum_{q \geq 0, \text{ even}} [\beta\Gamma(N-2k)]^q. \end{aligned} \quad (16)$$

Carrying out the sum over  $q$ , we end up with

$$Z = \sum_{k=0}^N \binom{N}{k} \cosh[\beta\Gamma(N-2k)] = (2 \cosh \beta\Gamma)^N, \quad (17)$$

which is the correct result for the partition function for the noninteracting system  $H = -\Gamma \sum_i \sigma_i^x$ .

## III. OFF-DIAGONAL EXPANSION QMC ALGORITHM

We now describe the basic ingredients of our off-diagonal expansion (ODE) algorithm, based on the above partition function decomposition. For concreteness we discuss the algorithm as it applies to the transverse-field Ising model [Eq. (9)], however, we note that generalization to other systems should be straightforward. We first establish the computational complexity associated with implementing this new algorithm, discussing in detail generic updates as well as measurements. We then present some results that allow us to fully characterize and to quantify the advantages of the algorithm over state-of-the-art QMC methods, specifically path-integral QMC and SSE.

### A. General description of the algorithm

An ODE configuration is a pair  $\mathcal{C} = \{|z\rangle, S_q\}$  where  $|z\rangle$  corresponds to a classical bit configuration and  $S_q = V_{i_1} V_{i_2} \dots V_{i_q}$  is a sequence of (possibly repeated) off-diagonal operators. As was discussed above, each configuration  $\mathcal{C}$  induces a list of states  $Z = \{|z_0\rangle = |z\rangle, |z_1\rangle, \dots, |z_q\rangle = |z\rangle\}$  (see Fig. 1), which in turn also generates a corresponding multiset of diagonal energies  $M_{\mathcal{C}} = \{E_0, E_1, \dots, E_q\}$  of not-necessarily-distinct values (recall that  $E_i = \langle z_i | H_c | z_i \rangle$ ). For systems with discretized energy values, the multiset can be stored efficiently in a ‘‘multiplicity table’’  $\{m_0, m_1, \dots, m_j, \dots\}$ , where  $m_j$  is the multiplicity of the energy  $E_j$  in the multiset. Given  $M_{\mathcal{C}}$ , the evaluation of the effective classical energy  $E_{\mathcal{C}}$  and the GBW  $W_{\mathcal{C}}$  follow from the definition of the GBW [Eq. (8)]. The actual evaluation of the effective classical energy is schematically

<sup>3</sup>This is to be contrasted with other decompositions of the partition function where the classical limit is either unnatural or ill defined.



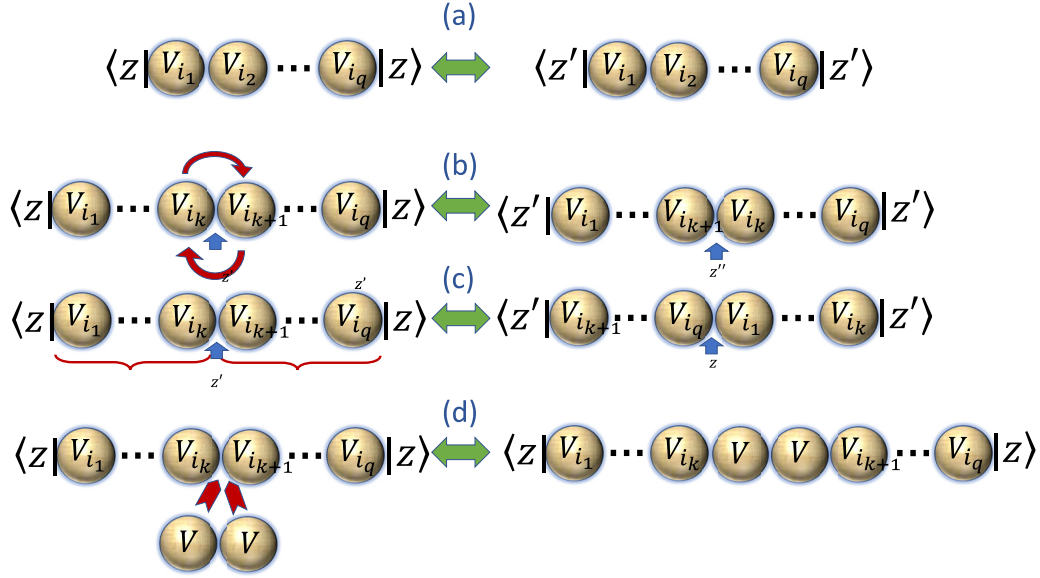


FIG. 3. Basic update moves of the ODE algorithm. (a) Classical moves (e.g., a single bit flip), whereby only the initial state  $z$  is changed to  $z'$  leaving  $S_q$  unchanged. (b) Local swap, whereby two adjacent operators  $V_{i_k} V_{i_{k+1}}$  are interchanged changing the state between them from  $z'$  to  $z''$ . (c) Block swap, whereby two partitions of the sequence are interchanged. This changes the initial state from  $z$  to  $z'$  as well as the ordering of the sequence. (d) Pair creation and annihilation, whereby a new pair of operators is inserted or deleted.

given in Fig. 2 and is discussed in more technical detail in Appendix C.

The initial configuration of the ODE algorithm is a random classical configuration  $|z\rangle$  and the empty sequence  $S_{q=0} = 1$ . The weight of this initial configuration is

$$W_{C_{\text{init}}} = e^{-\beta E(z)}, \quad (18)$$

i.e., the classical Boltzmann weight of the initial random state  $|z\rangle$ . Here, the effective classical energy  $E_{C_{\text{init}}}$  is the classical energy of  $|z\rangle$ .

## B. Updates

We next describe the basic update moves for the algorithm. We consider here only generic local updates but note that updates of the global type can be specifically tailored to the system in question. An update is considered local if it changes the multiset  $M_C$  by a finite (i.e., by a system-size-independent) number of terms, e.g.,  $M_C \rightarrow M_C + \{E(z_i)\} - \{E(z_j)\}$ . The basic updates are summarized in Fig. 3 and are discussed in detail below.

### 1. Classical moves

Classical moves are any moves that involve a manipulation of the classical state  $|z\rangle$  while leaving  $S_q$  unchanged [see Fig. 3(a)]. In a single bit-flip classical move, a spin from the classical bit-string state  $|z\rangle$  of  $\mathcal{C}$  is picked randomly and is flipped, generating a state  $|z'\rangle$  and hence a new configuration  $C'$ . Performing this change requires recalculating the energies associated with the sequence  $S_q$  leading to a new multiset  $M_{C'}$  and can become computationally intensive if  $q$  is large. Classical moves should therefore be attempted with low probabilities if  $q$  is large. Simply enough, the acceptance

probability for a classical move is

$$p = \min\left(1, \frac{W_{C'}}{W_C}\right) = \min(1, e^{-\beta \Delta E}), \quad (19)$$

where  $\Delta E = E_{C'} - E_C$  is the difference between the effective classical energies of the proposed configuration  $C'$  and current configuration  $C$ .

In the absence of a quantum part to the Hamiltonian ( $\Gamma = 0$ ), not only are classical moves the only moves necessary, but they are also the only moves with a nonzero acceptance probability. In this case, the ODE 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.

### 2. Local swap

A local swap is the swapping of neighboring off-diagonal operators in the sequence  $S_q$ . A random pair of adjacent off-diagonal operators in the sequence is picked and swapped [as shown in Fig. 3(b)]. If the state between  $V_{i_k}$  and  $V_{i_{k+1}}$  is  $|z'\rangle$  and is  $|z''\rangle$  after the swap, then the swap involves adding an energy  $E(z'')$  and removing an energy  $E(z')$  from the energy multiset [note that  $E(z')$  and  $E(z'')$  may be the same]. The acceptance probability for the move is as in Eq. (19) with  $M_{C'} = M_C + \{E(z'')\} - \{E(z')\}$ .

### 3. Block swap

A block swap [Fig. 3(c)] is a local update that involves a change of the classical state  $z$ . Here, a random position  $k$  in the sequence  $S_q$  is picked such that the sequence is split into two (nonempty) parts,  $S_q = S_1 S_2$ , with  $S_1 = V_{i_1} \dots V_{i_k}$  and  $S_2 = V_{i_{k+1}} \dots V_{i_q}$ . The classical state  $|z'\rangle$  at position  $k$  in the sequence is given by

$$\langle z' | = \langle z | S_1 = \langle z | V_{i_1} \dots V_{i_k}, \quad (20)$$

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  $E(z)$ . We consider a new configuration defined by  $\langle z'|S_2S_1|z'\rangle$ . The multiplicity table of this configuration differs from that of the current configuration by having one fewer  $E(z)$  state and one additional  $E(z')$  state. The weight of the new configuration is then proportional to  $e^{-\beta M_{C'}}$  where the multiset  $M_{C'} = M_C + \{E(z')\} - \{E(z)\}$ . The acceptance probability is as in Eq. (19) with the aforementioned  $M_{C'}$ .

#### 4. Creation and annihilation of off-diagonal operators

The moves presented so far have left the size of  $S_q$  unchanged. The creation-or-annihilation move shown in Fig. 3(d) has the effect of changing the value of  $q$  by 2, i.e.,  $q \rightarrow q \pm 2$ , which in the transverse-field Ising model corresponds to creating or destroying off-diagonal operators  $\sigma_j^x$  in pairs. We implement this via the insertion or deletion of two adjacent, identical operators. With probability  $p_{\text{del}}$  (e.g.,  $p_{\text{del}} = \frac{1}{2}$ ) we try to annihilate an adjacent pair, and with probability  $1 - p_{\text{del}}$  we try to insert a pair.

For pair insertion, we randomly pick an internal insertion point in the sequence (we denote this internal state by  $|z'\rangle$ ) and a random  $V$  to insert. This adds two new energies  $E(z')$  and  $E(z'')$  to the multiset, where  $|z''\rangle = V|z'\rangle$ . The acceptance probability for pair creation is given by

$$p = \min\left(1, \frac{p_{\text{del}}}{1 - p_{\text{del}}} \frac{N\beta^2\Gamma^2}{(q+2)(q+3)} e^{-\beta\Delta E}\right), \quad (21)$$

where as before  $\Delta E = E_{C'} - E_C$  is the difference between the effective classical energies of the proposed configuration  $C'$  and current configuration  $C$  and  $M_{C'} = M_C + \{E(z'), E(z'')\}$ . For deletion, we randomly pick an internal point in the sequence. If the two operators to the side of the insertion point are not identical, no deletion is performed, and the move is rejected. If the two operators are identical, they are deleted and the relevant energies  $E(z')$  and  $E(z'')$  are removed from the multiplicity table. The probability of acceptance for the deletion move is

$$p = \min\left(1, \frac{1 - p_{\text{del}}}{p_{\text{del}}} \frac{q(q+1)}{\beta^2\Gamma^2 N} e^{-\beta\Delta E}\right), \quad (22)$$

where as before  $\Delta E = E_{C'} - E_C$  and  $M_{C'} = M_C - \{E(z'), E(z'')\}$ .

The size of the imaginary-time dimension  $q$  comes strictly from off-diagonal terms and shrinks or grows depending on the strength of the ‘‘quantum component’’ of the model. This property is expected to be heavily utilized in order to overcome the freezing of QMC algorithms in almost classical regimes. In these regimes,  $q$  is small, and the algorithm reduces to being a classical thermal algorithm.<sup>4</sup>

<sup>4</sup>This is to be contrasted with the standard SSE formalism where one normally introduces an additional parameter  $L$  in order to fix the size of imaginary-time dimension for more efficient weight calculations. The fixing of the size of imaginary time may adversely affect the convergence of the algorithm if it is chosen to be too large. Here, this parameter too is spurious.

### C. Measurements

An integral part of any QMC algorithm is the acquisition of various properties of the model such as average energy, magnetization, specific heat, and correlation functions. In the ODE algorithm (as in SSE), diagonal (classical) measurements are measured differently than off-diagonal ones.

#### 1. Diagonal measurements

Diagonal operators  $D$  obey  $D|z\rangle = d(z)|z\rangle$  where  $d(z)$  is a number that depends both on the operator and the state it acts on. Since  $\langle z|DS_q|z\rangle = d(z)\langle z|S_q|z\rangle$ , for any given configuration  $C = (|z\rangle, S_q)$ , there is a contribution  $d = d(z)$  to the diagonal operator thermal average  $\langle D \rangle$ . To improve statistics, we can also consider rotations in (the periodic) imaginary time. To do that, we may consider ‘‘virtual’’ block-swap moves (see Sec. III B 3) that rotate  $S_q$  and as a result also change the classical configuration from  $|z\rangle$  to  $|z_i\rangle$ . The contribution to the expectation value of a diagonal operator  $D$  thus becomes

$$d = \frac{1}{\mathcal{Z}} \sum_{i=0}^{q-1} d(z_i) e^{-\beta E_{C_i}}, \quad (23)$$

where  $E_{C_i}$  is the effective classical energy associated with configuration  $C_i$  whose multiset is  $M_{C_i} = M_C + \{E(z_i)\} - \{E(z)\}$  (recall that  $z_0 \equiv z$ , so  $M_{C_0} = M_C$ ). The normalization factor  $\mathcal{Z}$  above is the sum

$$\mathcal{Z} = \sum_{j=0}^{q-1} e^{-\beta E_{C_j}} = \sum_j m_j e^{-\beta E_{C_j}} \quad (24)$$

over all nonzero multiplicities  $m_j$ . In the case where  $D = H_c$ , the above expression simplifies to

$$d = \frac{1}{\mathcal{Z}} \sum_{i=0}^{q-1} E(z_i) e^{-\beta E_{C_i}} = \frac{1}{\mathcal{Z}} \sum_j m_j E(z_j) e^{-\beta E_{C_j}}. \quad (25)$$

#### 2. Off-diagonal measurements

We next consider the case of measuring the expectation value of an off-diagonal operator  $V_k$ , namely,  $\langle V_k \rangle$ . To do this, we interpret the instantaneous configuration as follows:

$$\begin{aligned} W_C &= \frac{(\beta\Gamma)^q e^{-\beta E_C}}{q!} \langle z|S_q|z\rangle \\ &= \left(\frac{\beta\Gamma}{q e^{-\beta\Delta E}}\right) \left[ \frac{(\beta\Gamma)^{q-1} e^{-\beta E_{C'}}}{(q-1)!} \langle z|S_{q-1}V_{i_q}|z\rangle \right], \end{aligned} \quad (26)$$

where  $\Delta E = E_{C'} - E_C$  and  $C'$  is the configuration associated with the multiset  $M_{C'} = M_C - \{E(z)\}$ . In the above form, we can reinterpret the weight  $W_C$  as contributing

$$v_k = \delta_{k,i_q} \frac{q e^{-\beta\Delta E}}{\beta\Gamma} \quad (27)$$

to  $\langle V_k \rangle$ .

As in the case of the diagonal measurements, one can take advantage of the periodicity in the imaginary-time direction to improve statistics by rotating the sequence such that any of the elements of  $S_q$  become the last element of the sequence

(see Sec. III B 3), weighted accordingly by the block-swap probability. By doing so,  $v_k$  becomes

$$v_k = \sum_i \frac{q}{\beta\Gamma} \frac{e^{-\beta E_{C_i}}}{\sum_{j=0}^{q-1} e^{-\beta E_{C_j}}} \frac{e^{-\beta E_{C'_i}}}{e^{-\beta E_{C_i}}} = \frac{q N_k}{\mathcal{Z}\beta\Gamma} e^{-\beta E_{C'_i}}, \quad (28)$$

where  $M_{C'_i} = M_C + \{E(z_i)\} - \{E(z)\}$ , the sum  $\sum_i$  is over all rotated configurations  $C'$  whose  $S_q$  ends with  $V_k$ , and  $N_k$  is the number of times  $V_k$  appears in the sequence  $S_q$ .

### 3. Products of off-diagonal measurements

The sampling of the expectation values of the form  $\langle V_{k_1} V_{k_2} \rangle$  proceeds very similarly to the single operator case except that now both operators must appear at the end of the sequence. The argument proceeds similarly to the single off-diagonal measurement, and we have that the contribution to the expectation value of  $\langle V_{k_1} V_{k_2} \rangle$  is

$$v_{k_1, k_2} = \delta_{i_{q-1}, k_1} \delta_{i_q, k_2} \frac{q(q-1)}{\beta^2 \Gamma^2} \frac{e^{-\beta E_{C'_i}}}{e^{-\beta E_C}}, \quad (29)$$

with  $M_{C'_i} = M_C - \{E(z), E(z_{q-1})\}$ . As in the single off-diagonal operator case, we can use the block-swap move to alter the elements at the end of the sequence, and for each pair of adjacent operators in the sequence obtain an improved contribution

$$\begin{aligned} v_{k_1, k_2} &= \frac{q(q-1)}{\beta^2 \Gamma^2} \sum_i \frac{e^{-\beta E_{C_i}}}{\sum_{j=0}^{q-1} e^{-\beta E_{C_j}}} \frac{e^{-\beta E_{C'_i}}}{e^{-\beta E_{C_i}}} \\ &= \frac{q(q-1)}{\mathcal{Z}\beta^2 \Gamma^2} \sum_i e^{-\beta E_{C'_i}}, \end{aligned} \quad (30)$$

where  $M_{C_i} = M_C + \{E(z_k)\} - \{E(z)\}$ ,  $M_{C'_i} = M_C - \{E(z), E(z'')\}$  with  $|z''\rangle = V_{k_2}|z'\rangle$ , and  $|z'\rangle$  is the classical state after the block swap. Similar to the single off-diagonal operator case, the sum  $\sum_i$  is over all rotated configurations  $C'$  whose  $S_q$  ends with  $V_{k_1} V_{k_2}$ .

Measurements of thermal averages of products of more than two off-diagonal operators can also be derived in a straightforward manner.

### D. ODE weights vs SSE weights

It is worthwhile to contrast the properties of standard SSE weights against ODE weights. Foremost, while in standard SSE the classical part of the Hamiltonian must be broken up into local bonds, in ODE this is not the case: within ODE the diagonal portion remains “intact.” Moreover, a single divided-difference term is a sum of an infinite number of terms of standard SSE weights, meaning that ODE weights will generally represent very many standard SSE configurations. This can be immediately seen in Eq. (3), which relates the standard SSE weight involving sequences of diagonal as well as off-diagonal bonds (denoted by  $S_n$ ) to the weights of the current algorithm that only involve off-diagonal bonds. The differences above suggest that ODE weights are orders-of-magnitude larger than SSE weights, and as such fewer of them are required for the algorithm to converge.

On the other hand, the computational cost of updating ODE weights within a Monte Carlo scheme is generally higher than standard SSE weights since the calculation of

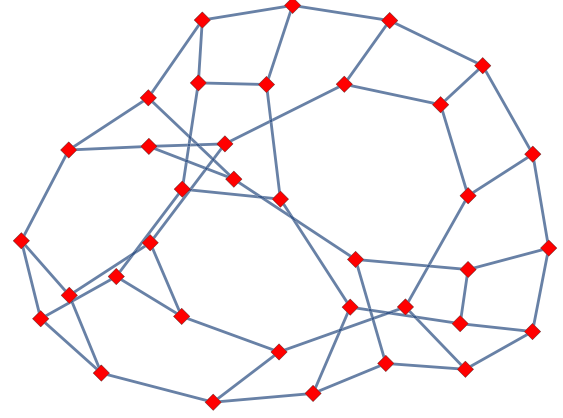


FIG. 4. Connectivity of a randomly generated  $N = 36$ -spin 3-regular MAX2SAT instance. Here, the diamonds denote spins and the edges denote antiferromagnetic couplings with strength  $J_{ij} = 1$ . Each spin is connected to three other randomly chosen spins.

the effective classical energy is, as we show below explicitly, proportional to the number of terms in the sequence  $S_q$  (by direct divided-differences calculation, see Appendix B), which scales linearly with the inverse temperature  $\beta$  and with the number of particles in the system  $N$  (this is discussed in detail later on). Within SSE, weight updates normally require a fixed number of operations.

As we demonstrate below, the balance between the cost increase and weight size can result in considerable speedups in general over standard methods, including SSE.

## IV. RESULTS

Having laid the groundwork for the ODE QMC algorithm, we present in this section some results that highlight some of its properties and advantages over existing QMC techniques, specifically a discrete-time PIQMC algorithm with cluster updates<sup>5</sup> and with the continuous-time SSE algorithm with cluster updates [7, 13]. For benchmarking purposes, we study random 3-regular MAX2SAT instances augmented with a transverse field. This class of instances corresponds to a particular choice of the Ising Hamiltonian given in Eq. (9), whereby each spin is coupled antiferromagnetically (with strength  $J_{ij} = 1$ ) with exactly three other spins picked at random (see Fig. 4 for an illustration). This class of instances is known to exhibit a quantum spin-glass phase transition and is notoriously difficult to simulate by standard QMC techniques (see, e.g., Refs. [13, 26]), making it suitable to illustrate the strengths of the ODE algorithm.

### A. Correctness of algorithm and elastic imaginary time

As a preliminary test, we verify that we are able to reproduce the correct thermal expectation values for sufficiently small systems where comparison to exact diagonalization is feasible. An example is given in Fig. 5 illustrating the excellent

<sup>5</sup>We use Wolff cluster updates [24] along the imaginary-time direction [25].

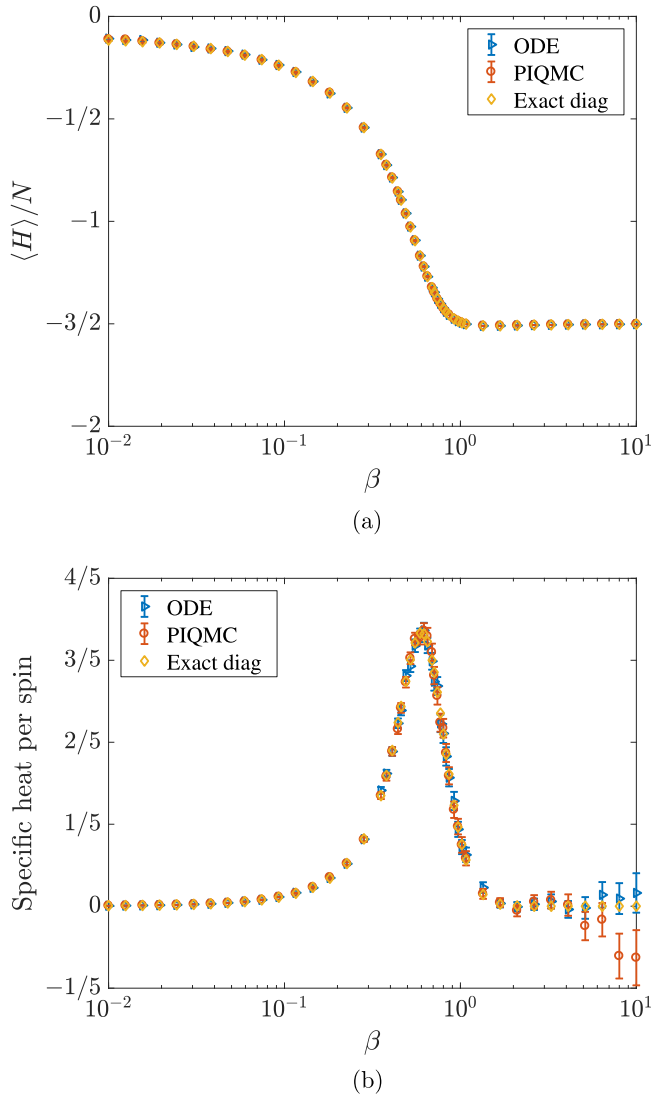


FIG. 5. Agreement between ODE, PIQMC, and exact diagonalization for small systems. The thermal expectation value of the internal energy per spin  $\langle H \rangle / N$  and specific heat per spin  $C = \beta^2(\langle H^2 \rangle - \langle H \rangle^2) / N$  for a 3-regular MAX2SAT instance of size  $N = 12$  for a range of  $\beta$  with  $\Gamma = (10\beta)^{-1/2}$  as calculated using ODE, PIQMC (with 5120 Trotter slices), and exact diagonalization. Error bars correspond to  $2\sigma$  generated by performing 1000 bootstraps over the measurements.

agreement of ODE with the exact-numerical values, even in the high- $\beta$  but low- $\Gamma$  regime where PIQMC begins to show deviations from the exact results. Increasing the number of measurements for PIQMC rectifies this discrepancy, but the deviation already suggests that the ODE algorithm may require fewer measurements over PIQMC in the low- $\Gamma$  but large- $\beta$  regime.

We next study in Fig. 6 the dependence of the average size of the imaginary-time dimension, namely,  $q$  on system size  $N$ , inverse temperature  $\beta$ , and transverse-field strength  $\Gamma$ .<sup>6</sup>

<sup>6</sup>The warm-up of the simulations involved a linear anneal in  $\beta$  from an initial value that is a factor  $10^3$  smaller than the target  $\beta$  to the

As was discussed earlier, the ODE QMC does not presume *a priori* a size for the imaginary-time dimension but rather allows it to be set dynamically during the simulation. As is shown in Fig. 6(a), as the simulation advances, the instantaneous  $q$  which starts at  $q = 0$  gradually grows and eventually fluctuates around an average value indicating the size of the imaginary-time dimension. As we expect, the average value of  $q$ , which we denote  $\langle q \rangle$ , scales linearly with  $N$  and  $\beta$  with fluctuations on the order of  $\sqrt{N}$  and  $\sqrt{\beta}$  [Figs. 6(b) and 6(c), respectively]. Moreover, we find that  $\langle q \rangle$  does indeed grow with the quantum strength of the model. Specifically, we find it to scale quadratically with  $\Gamma$  as indicated in Fig. 6(d).

## B. Quantum-classical parallel tempering

As we demonstrated in Sec. II, the ODE partition function decomposition naturally reduces to the classical one when the strength of the off-diagonal terms in the Hamiltonian are sent to zero. As we show next, this allows us to naturally unify the classical parallel tempering (CPT) algorithm (also known as “exchange Monte Carlo”) [27,28] and its quantum counterpart (QPT see, e.g., Refs. [29,30]). CPT is a refinement of the simulated annealing algorithm [31], whereby  $N_T$  replicas of an  $N$ -spin system at inverse temperatures  $\beta_1 < \beta_2 < \dots < \beta_{N_T}$  undergo Metropolis spin-flip updates independently of one another and, in addition, replicas with neighboring temperatures regularly attempt to swap their temperatures with probabilities that satisfy detailed balance [32]. In this way, each replica performs a random walk on the temperature axis, which generally allows for quicker equilibration of the system in comparison to other techniques. Analogously in QPT, temperature is replaced by a parameter  $\Gamma$  of the (quantum) Hamiltonian, e.g., the strength of the transverse magnetic field in the transverse Ising model, and each replica performs a random walk on the  $\Gamma$  axis.

Both CPT and QPT are two widely used variations on Monte Carlo schemes but have so far been considered as separate algorithms. The current formulation allows to unify the two tempering algorithms in a straightforward and natural manner. A simple generalization is to consider a tempering algorithm that traces an arbitrary curve in the classical-quantum  $\beta$ - $\Gamma$  plane. This opens up the opportunity to study, e.g., certain properties of experimental quantum annealers (see, for example, Ref. [33]) which trace such quantum-classical curves as well as to study classical-quantum optimization techniques and equilibration methods, by, e.g., looking for curves that would allow one to bypass first order phase transitions.

If we consider replicas along a curve in the  $\beta$ - $\Gamma$  plane at points  $\{(\beta_1, \Gamma_1), \dots, (\beta_{N_T}, \Gamma_{N_T})\}$ , then a parallel tempering swap probability between the  $i$ th and  $(i+1)$ th replica is given by

$$P = \min \left( 1, \frac{W_{C_i}(\beta_{i+1}, \Gamma_{i+1}) W_{C_{i+1}}(\beta_i, \Gamma_i)}{W_{C_i}(\beta_i, \Gamma_i) W_{C_{i+1}}(\beta_{i+1}, \Gamma_{i+1})} \right), \quad (31)$$

target  $\beta$ .  $10^6$  sweeps are performed in total during the warm-up. After the warm-up,  $10^4$  measurements are performed, with  $10^2$  sweeps between measurements to ensure the subsequent measurements are uncorrelated.



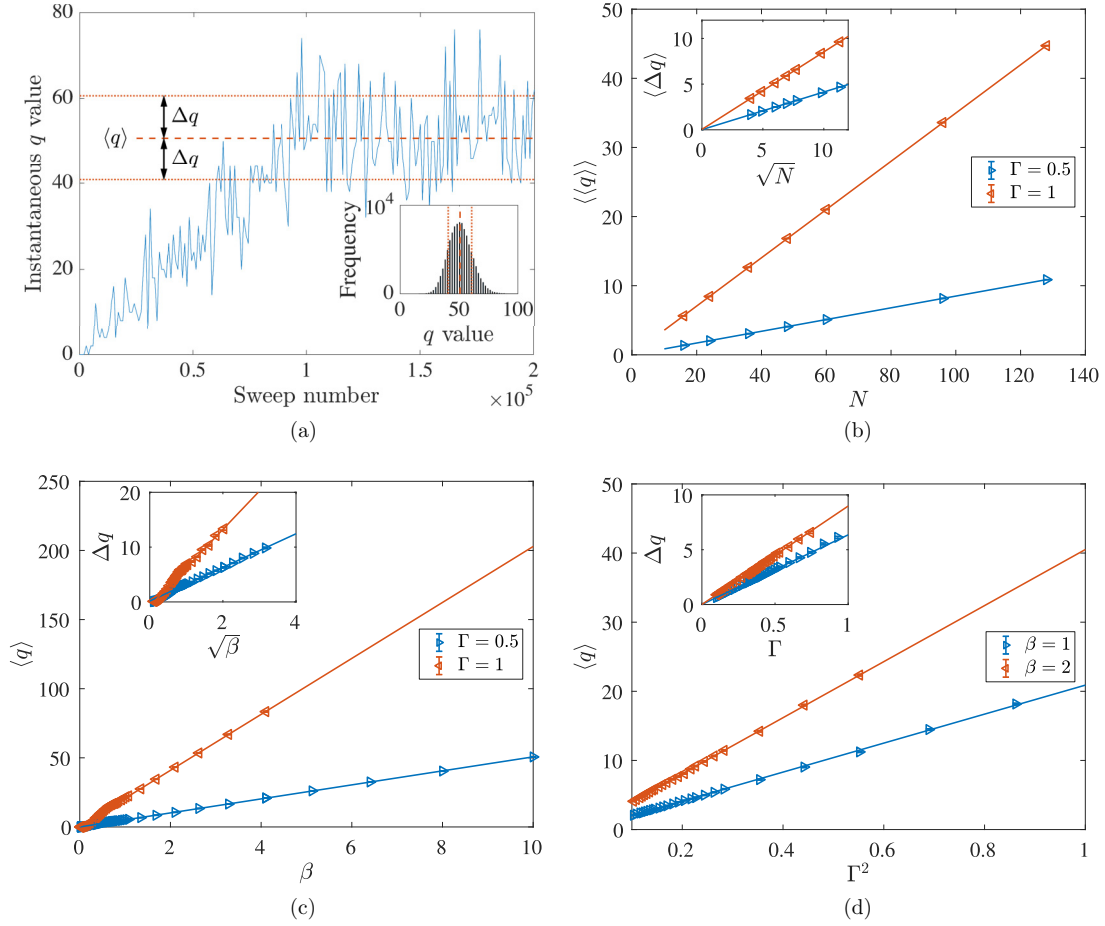


FIG. 6. Size of imaginary-time dimension as a function of inverse temperature, problem size, and quantum strength. (a) Instantaneous value of  $q$  as the algorithm advances, showing  $q$  growing gradually from zero and then stabilizing around a mean value  $\langle q \rangle$  (dashed red line) with fluctuations  $\Delta q = \sqrt{\langle q^2 \rangle - \langle q \rangle^2}$  (dotted red line). (b)  $\langle q \rangle$  averaged over 48 instances (denoted  $\langle \langle q \rangle \rangle$ ) as a function of problem size (here,  $\beta = 1$  and  $\Gamma = 0.5, 1$ ). The inset shows  $\Delta q$  averaged over 48 instances (denoted  $\langle \Delta q \rangle$ ) as a function of  $\sqrt{N}$ . (c)  $\langle q \rangle$  as a function of  $\beta$  for a single instance of size  $N = 60$  and  $\Gamma = 0.5, 1$ . The inset shows  $\Delta q$  as a function of  $\sqrt{\beta}$ . Inset shows  $\Delta q$  as a function of  $\sqrt{N}$ . (d)  $\langle q \rangle$  as a function of  $\Gamma$  for the same instance as in (b) of size  $N = 60$  and  $\beta = 1, 2$ . (b)–(d) The solid curves correspond to linear fits of the data points. Error bars correspond to  $2\sigma$  generated by performing 1000 bootstraps over the measurements for (a) and (d) and over the mean from the 48 instances for (b) and (c).

where the above weight ratio is conveniently simplified to

$$\frac{W_{C_i}(\beta_{i+1}, \Gamma_{i+1}) W_{C_{i+1}}(\beta_i, \Gamma_i)}{W_{C_i}(\beta_i, \Gamma_i) W_{C_{i+1}}(\beta_{i+1}, \Gamma_{i+1})} = \left( \frac{\beta_i \Gamma_i}{\beta_{i+1} \Gamma_{i+1}} \right)^{q_{i+1} - q_i} \frac{e^{-\beta_i (E'_{C_{i+1}} - E_{C_i})}}{e^{-\beta_{i+1} (E_{C_{i+1}} - E'_{C_i})}}, \quad (32)$$

where  $E_{C_i}$  and  $E_{C_{i+1}}$  are the effective classical energies of configurations  $C_i$  and  $C_{i+1}$ , respectively, and  $E'_{C_i}$  and  $E'_{C_{i+1}}$  are the effective classical energies of these configurations when calculated with switched  $\beta$  and  $\Gamma$ .

In the classical limit  $\Gamma \rightarrow 0$  the ratio readily reduces to the standard CPT acceptance ratio

$$\frac{W_{C_i}(\beta_{i+1}) W_{C_{i+1}}(\beta_i)}{W_{C_{i+1}}(\beta_{i+1}) W_{C_i}(\beta_i)} = e^{\Delta\beta \Delta E}, \quad (33)$$

where  $\Delta\beta = \beta_{i+1} - \beta_i$  and  $\Delta E$  is the change in classical energy between the two configurations. Furthermore, in the case of pure quantum parallel tempering, i.e., if  $\beta$  is fixed

between neighboring replicas, the acceptance ratio neatly reduces to

$$\frac{W_{C_i}(\beta, \Gamma_{i+1}) W_{C_{i+1}}(\beta, \Gamma_i)}{W_{C_i}(\beta, \Gamma_i) W_{C_{i+1}}(\beta, \Gamma_{i+1})} = \left( \frac{\Gamma_i}{\Gamma_{i+1}} \right)^{q_{i+1} - q_i}. \quad (34)$$

We show in Fig. 7 results for our quantum-classical parallel tempering (QCPT) algorithm along different curves in the  $\beta$ - $\Gamma$  plane. The parallel tempering algorithm gives excellent agreement with numerical calculations using PIQMC with a temperature annealing protocol for every individual  $(\beta, \Gamma)$  point (as in our comparisons in the previous section).<sup>7</sup> We nicely see the precursor of the quantum phase transition in our results.

<sup>7</sup>The QPT algorithms used  $10^5$  swap sweeps with 10 MC sweeps per swap, whereas the PIQMC algorithm used  $10^6$  sweeps and 5120 Trotter slices. Both algorithms took  $10^4$  measurements with 100 sweeps between measurements.

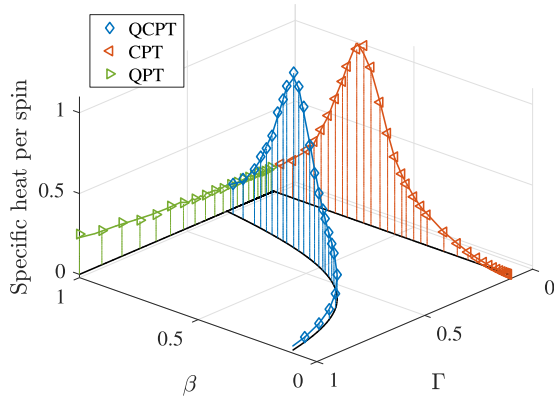


FIG. 7. Classical parallel tempering (CPT), quantum parallel tempering (QPT), and quantum-classical parallel tempering (QCPT). Results for an instance with  $N = 60$  using our generalized parallel tempering algorithm along different curves in the  $\beta$ - $\Gamma$  plane. Shown are the cases for CPT (red), QPT (green), and a case where  $\Gamma = (10\beta)^{-1/2}$  (blue). To verify the accuracy of our algorithm, we also show the PIQMC prediction (solid line).

It is important to note that while parallel tempering along one-dimensional curves in the  $\beta$ - $\Gamma$  plane is certainly possible in other QMC techniques, they are not natural in the following sense. Both discrete-time PIQMC and the continuous-time SSE (in their standard applications) require setting up the size of imaginary-time dimension prior to execution. In PIQMC, the number of slices is a user-chosen parameter and in SSE the size is determined in a dynamical way prior to the full run. In both cases, all PT replicas must use the same imaginary-time size for a swap to take place, which then requires the PT simulation to use the largest imaginary-time size needed over all  $\beta$ - $\Gamma$  points along the chosen curve. This can be a source of considerable inefficiency since many points in the  $\beta$ - $\Gamma$  plane may not normally require such a large number of imaginary-time slices (e.g., if  $\beta$  and/or  $\Gamma$  are small).

### C. Benchmarks against PIQMC and SSE

To demonstrate the advantages of ODE over PIQMC and SSE, we carry out multiple runs of parallel tempering QMC simulations on random MAX2SAT instances augmented with a transverse field [see Eq. (9)] with various problem sizes  $N$ , a range of  $\beta$  (inverse temperature), and  $\Gamma$  (quantum strength) values.

We begin by comparing ODE against the discrete-time PIQMC algorithm. Since the value of  $q$  determines the cost of calculating the GBWs, our results in Fig. 6 indicate that the ODE algorithm can have significant advantages in the low- $\Gamma$  but large- $\beta$  regime. For the 3-regular MAX2SAT class, this would be in the spin-glass phase, where we can expect QMC algorithms to become less efficient. We quantify this possible advantage by comparing the performance of our algorithm against PIQMC and SSE in this regime. In Fig. 8, we compare the warm-up time required to reach close to the thermal state for ODE and PIQMC. We observe that in order for the (discrete-time) PIQMC algorithm to achieve this, we need a sufficiently large Trotter slicing ( $>1024$ ), which in turn increases the time cost of performing a sweep in the simulations. In this regard, the ODE algorithm reaches the thermal state in less computational time, with even a factor of 10 advantage when compared to PIQMC with 2048 Trotter slices.

Runtime comparisons of ODE against both PIQMC and SSE are summarized in Figs. 9 and 10. Both figures depict the performance of ODE, PIQMC, and SSE as it is reflected by the thermal average of  $x$  magnetization  $H_x = -\sum_{i=1}^N \sigma_i^x$  (other observables exhibit similar behavior) for increasing problem size and different values of  $\beta$  and  $\Gamma$ . Each data point in each of the panels is an average over eight random instances.

The first of the figures, Fig. 9, depicts the results of PT simulations done along a fixed  $\Gamma$  line in the  $\beta$ - $\Gamma$  plane, that is, different  $\beta$  values. As is clearly evident from the figure, while for large temperatures ( $\beta = 2$ ) the performance of all three algorithms is comparable, with decreasing temperature

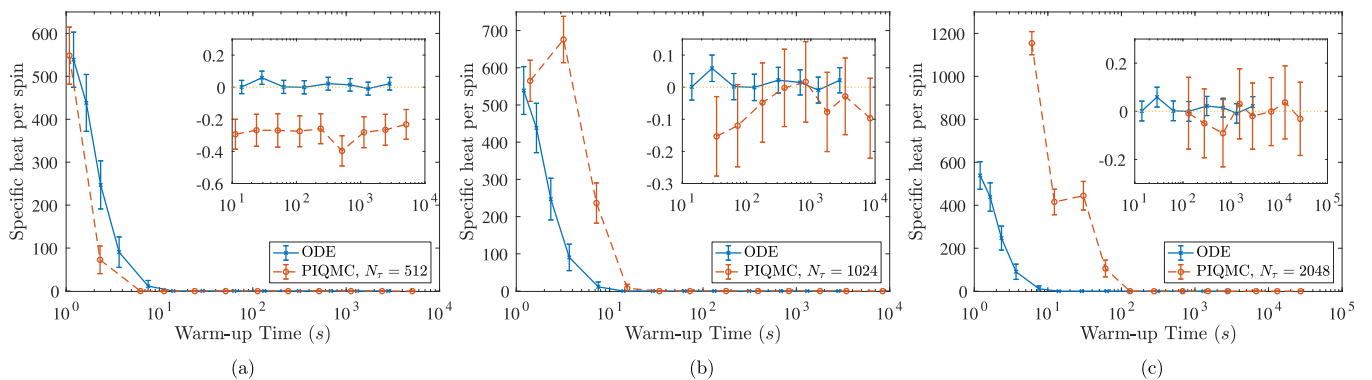


FIG. 8. Performance of ODE vs PIQMC. Required simulation time to reach a thermal state for  $\Gamma = 0.1$  and  $\beta = 30$ . Here, we calculate the specific heat per spin  $C = \beta^2(\langle H^2 \rangle - \langle H \rangle^2)/N$ . For these values, we expect the thermal state to have  $\langle H^2 \rangle \approx \langle H \rangle^2$  since the thermal state should have almost all its weight on the ground state. (a) PIQMC with 512 Trotter slices ( $N_T$ ). (b) PIQMC with 1024 Trotter slices. (c) PIQMC with 2048 Trotter slices. The warm-up of both ODE and PIQMC simulations involved a linear anneal in  $\beta$  from an initial value that of 0.1 to the target  $\beta$  of 30. The number of warm-up sweeps was varied from  $10^2$  to  $10^6$ . After the warm-up, 100 measurements are performed, with  $10^2$  sweeps between measurements. We ran  $10^3$  independent simulations. Error bars correspond to  $2\sigma$  generated by performing  $10^3$  bootstrap over the mean values from the  $10^3$  independent simulations [34].

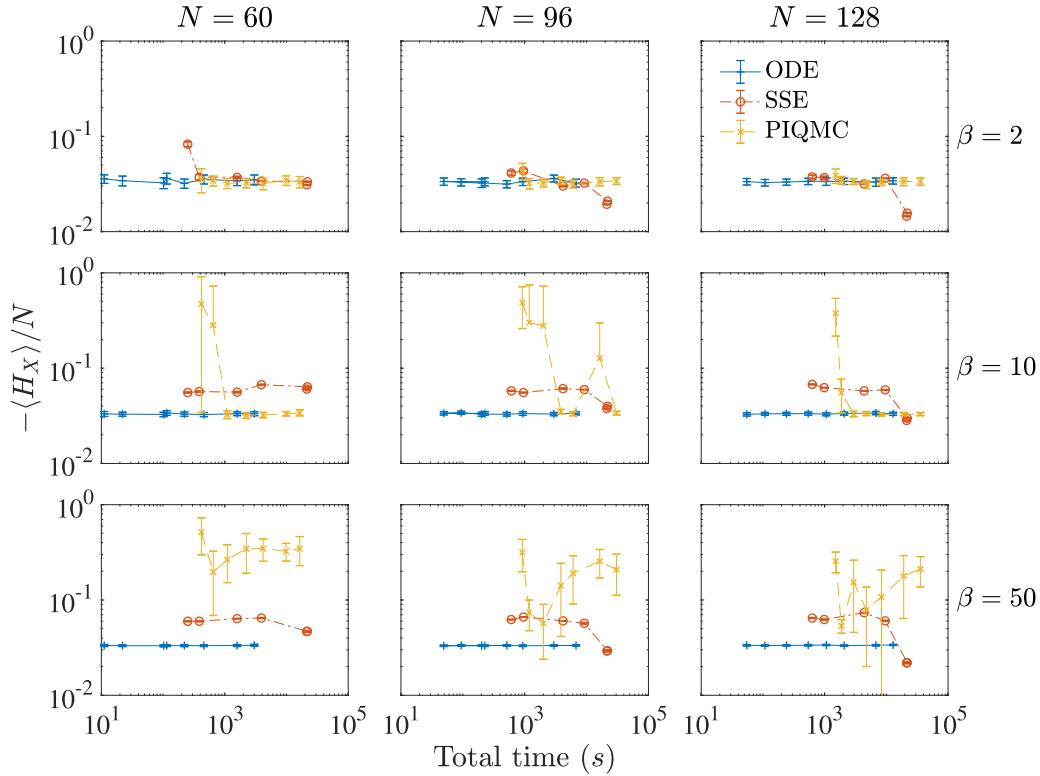


FIG. 9. Thermal average of the off-diagonal Hamiltonian as obtained by ODE, PIQMC, and SSE for a fixed  $\Gamma = 0.1$ . Results are obtained using PT simulations of different system sizes  $N = 60, 96,$  and  $128,$  each with 11 replicas at inverse temperatures  $\beta \in \{0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 30, 40, 50\}.$  A subset of the results are shown for  $\beta = 2, 10,$  and  $50.$  Each data point is the mean value of  $10^3$  bootstraps performed on the 500 measurements of the  $x$ -magnetization-per-spin taken in the course of simulations with different runtimes (horizontal axis), while the error bars correspond to the  $2\sigma$  confidence interval generated by the bootstrap. For PIQMC we use 1280 Trotter slices. As is evident, ODE converges to the true value at times orders of magnitude faster than both SSE and PIQMC [34].

and as problem sizes grow, ODE performs up to four and more orders of magnitude faster than the other two algorithms. Specifically, a large fraction of the instances did not finish

running over the 24 hour window allocated for each of the simulations, whereas ODE had converged after a few seconds.

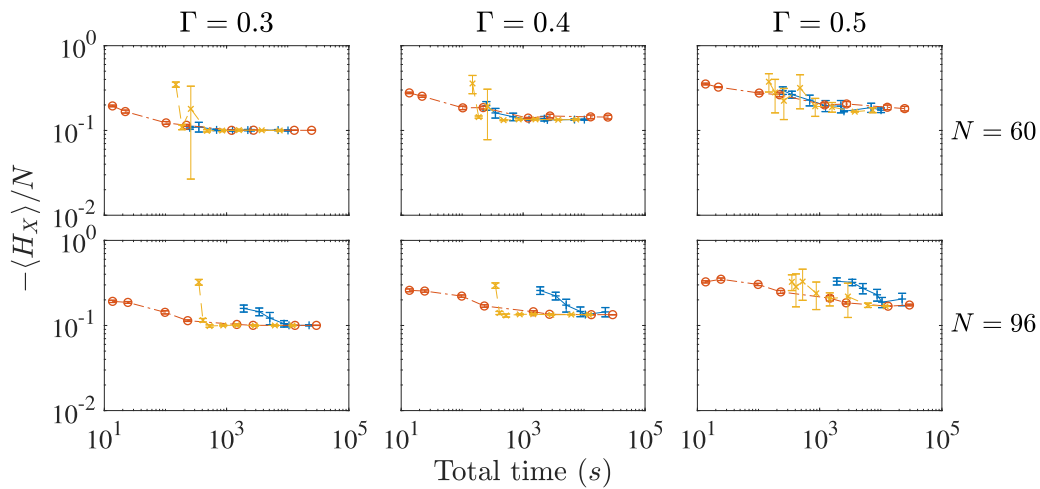


FIG. 10. Thermal average of the off-diagonal Hamiltonian as obtained by ODE, PIQMC, and SSE for a fixed  $\beta = 10.$  Results are obtained using PT simulations of different system sizes  $N = 60$  and  $96,$  each with 5 replicas at transverse-field strengths  $\Gamma \in \{0.1, 0.2, 0.3, 0.4, 0.5\}.$  A subset of the results are shown for  $\Gamma = 0.3, 0.4,$  and  $0.5.$  Each data point is the mean value of  $10^3$  bootstraps performed on the 500 measurements of the  $x$ -magnetization-per-spin taken in the course of simulations with different runtimes (horizontal axis), while the error bars correspond to the  $2\sigma$  confidence interval generated by the bootstrap. For PIQMC we use 1280 Trotter slices. Here, all three algorithms perform similarly [34].

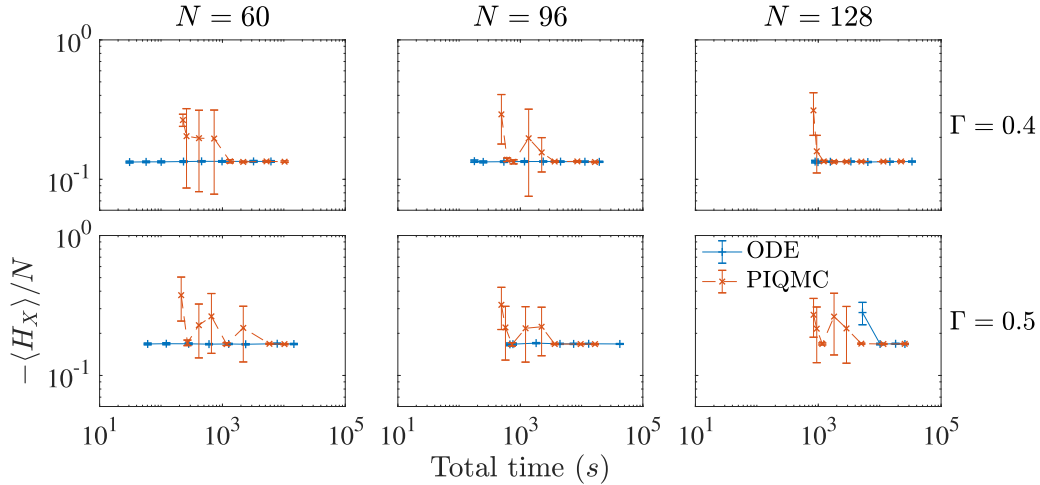


FIG. 11. Thermal average of the off-diagonal Hamiltonian as obtained by ODE and PIQMC at large  $\Gamma$  values. Results are obtained using PT simulations of different system sizes  $N = 60, 96,$  and  $128$ , each with 7 replicas at inverse temperatures  $\beta \in \{0.1, 0.2, 0.5, 1, 2, 5, 10\}$ . Results are shown for  $\beta = 10$ . Each data point is the mean value of  $10^3$  bootstraps performed on the 500 measurements of the  $x$ -magnetization-per-spin taken in the course of simulations with different runtimes (horizontal axis), while the error bars correspond to the  $2\sigma$  confidence interval generated by the bootstrap. For PIQMC we use 1280 Trotter slices. As is evident, even at large  $\Gamma$  values, ODE can converge faster than PIQMC by several orders of magnitude [34].

Figure 10 depicts the results of PT simulations done along a fixed  $\beta$  line in the  $\beta$ - $\Gamma$  plane, that is, different  $\Gamma$  values. Interestingly, here all three algorithms seem to perform comparably, indicating that ODE is, as expected, at its optimum in near classical regimes.

To demonstrate the performance of the ODE algorithm away from the “almost classical” regime, i.e., in regimes where the quantum strength parameter  $\Gamma$  is not small, we show in Fig. 11 some results obtained for  $\Gamma = 0.4$  and  $0.5$  for different problem sizes. Here too, we find a regime where ODE converges faster to the true values by several orders of magnitude than PIQMC (not shown in this figure are SSE results, which we expect perform similarly to PIQMC in these regimes as is evident from the data presented in the preceding figures).

## V. CONCLUSIONS

We have developed a parameter-free Monte Carlo scheme designed to simulate quantum and classical many-body systems under a single unifying framework. The method is based on a decomposition of the quantum partition function that can be viewed as an expansion in the “quantumness” of the system. We have argued that the classical limit of the expansion together with the elastic quantum dimension make the method suitable to simulate models that exhibit the full range of quantum and classical behavior, specifically, systems with a non-negligible glassy classical component, which are often difficult to simulate using existing QMC techniques. We have shown that a single weight in the proposed decomposition corresponds to infinitely many weights of the standard SSE algorithm and have demonstrated the effectiveness of our algorithm using instances from 3-regular MAX2SAT where clear advantages can be observed over PIQMC in the near-classical regime.

This near-classical regime is particularly relevant in the case of the quantum spin glasses when additional structure

occurs after the paramagnetic spin-glass phase transition, as numerically shown for 3-regular MAX2SAT in Ref. [13]. An analytic study of the two-pattern Gaussian Hopfield model in Ref. [35] exhibits similar rich behavior in terms of  $O(\log N)$  ground state minima in the spin-glass phase. Such examples are particularly relevant for adiabatic approaches to quantum computation like quantum annealing [36–45], where these minima in the spin-glass phase may be the relevant bottlenecks for the quantum algorithm as opposed to the critical point associated with the paramagnetic spin-glass phase transition.

Additionally, the feature of naturally transitioning from the quantum to the classical regime also lends itself to simulating quantum annealing. Since quantum annealing processes are typically simulated by applying equilibrium QMC algorithms to a slowly changing Hamiltonian interpolating between a transverse-field initial Hamiltonian and a typically classical final Hamiltonian (see, for example, Refs. [7, 13, 14, 30, 46–50]; a notable exception to this approach is Ref. [26]), it has become crucially important to devise quantum Monte Carlo approaches capable of effectively simulating the full range of the quantum annealing process. We believe our algorithm will be particularly suited for this purpose.

We also showed how the algorithm naturally unifies classical and quantum parallel tempering into a single parallel tempering process along curves in the classical-quantum  $\beta$ - $\Gamma$  plane. This highlights a key feature of our method, which is that it naturally bridges the algorithmic gap between quantum Monte Carlo and classical (thermal) Monte Carlo. This property opens up the possibility of exploring optimal curves that speed up equilibration in the classical-quantum plane.

We have demonstrated how the algorithm applies to the transverse-field Ising model. It would be interesting to see how it performs with respect to existing techniques on other models considered difficult to simulate. Another aspect worth studying is the existence of additional updates that are more



global in nature in order to further speed up convergence. These will more likely have to be specifically tailored to the system in question. Last, methods to facilitate the evaluation of the generalized Boltzmann weights are of significance as these scale in the worst case as the square of the imaginary-time dimension. More efficient methods will serve to further increase the usefulness of the ODE algorithm. We leave the resolution of these questions for future work.

### ACKNOWLEDGMENTS

T.A. was supported under ARO MURI Grant No. W911NF-11-1-0268, ARO MURI Grant No. W911NF-15-1-0582, and NSF Grant No. INSPIRE-1551064. Computation for the work described here was supported by the University of Southern California's Center for High-Performance Computing (<http://hpcc.usc.edu>). The research is based upon work (partially) supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via the US Army Research Office Contract No. W911NF-17-C-0050. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the US Government. The US Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

### APPENDIX A: PARTITION FUNCTION DERIVATION

Here, we discuss in more detail the derivation of the partition function decomposition to as sum of Boltzmann-type weights. We begin by recalling that the partition function may be written as

$$Z = \sum_z \sum_{q=0}^{\infty} \sum_{\{S_q\}} \Gamma^q \langle z | S_q | z \rangle \left( \sum_{n=q}^{\infty} \frac{\beta^n (-1)^{n-q}}{n!} \times \sum_{\sum k_i = n-q} E^{k_0}(z_0) \dots E^{k_q}(z_q) \right), \quad (\text{A1})$$

where  $E(z_i) = \langle z_i | H_c | z_i \rangle$  and  $\{S_q\}$  denotes the set of all sequences of length  $q$  of “bare” *off-diagonal* operators  $V_j$ . The term in large parentheses sums over the diagonal contribution of all  $\langle z | S_n | z \rangle$  terms that correspond to a single  $\langle z | S_q | z \rangle$  term. The various  $\{z_i\}$  states are the states obtained from the action of the ordered  $V_j$  operators in the sequence  $S_q$  on  $|z_0\rangle$ , then on  $|z_1\rangle$ , and so forth.<sup>8</sup> After a change of variables,  $n \rightarrow n + q$ , we arrive at

$$Z = \sum_z \sum_{q=0}^{\infty} \sum_{\{S_q\}} \langle z | S_q | z \rangle \left( (\beta\Gamma)^q \sum_{n=0}^{\infty} \frac{(-\beta)^n}{(n+q)!} \times \sum_{\sum k_i = n} E^{k_0}(z_0) \dots E^{k_q}(z_q) \right). \quad (\text{A2})$$

<sup>8</sup>For example, for  $S_q = V_{i_q} \dots V_{i_2} V_{i_1}$ , we obtain  $|z_0\rangle = |z\rangle, V_{i_1}|z_0\rangle = |z_1\rangle, V_{i_2}|z_1\rangle = |z_2\rangle$ , etc.

Abbreviating  $E_i \equiv E(z_i)$  (note that the various  $\{E_i\}$  are functions of the  $|z_i\rangle$  states created by the operator sequence  $S_q$ ), the partition function is now given by

$$Z = \sum_{q=0}^{\infty} (-\Gamma)^q \sum_{z, \{S_q\}} \langle z | S_q | z \rangle \times \left( \sum_{\{k_i\}=(0,\dots,0)}^{(\infty,\dots,\infty)} \frac{(-\beta)^q}{(q + \sum k_i)!} \prod_{j=0}^q (-\beta E_j)^{k_j} \right). \quad (\text{A3})$$

A feature of the above infinite sum is that the term in large parentheses can be further simplified to give the *exponent of divided differences* of the  $E_i$ 's (we give a short description of divided differences and an accompanying proof of the above assertion in Appendix B), namely, it can be succinctly rewritten as

$$\sum_{\{k_i\}} \frac{(-\beta)^q}{(q + \sum k_i)!} \prod_{j=0}^q (-\beta E_j)^{k_j} = e^{-\beta[E_0, \dots, E_q]},$$

where  $[E_0, \dots, E_q]$  is a *multiset* of energies and where a function  $F[\cdot]$  of a multiset of input values is defined by

$$F[E_0, \dots, E_q] \equiv \sum_{j=0}^q \frac{F(E_j)}{\prod_{k \neq j} (E_j - E_k)} \quad (\text{A4})$$

and is called the *divided difference* [19,20] of the function  $F[\cdot]$  with respect to the list of real-valued input variables  $[E_0, \dots, E_q]$ . In our case,  $F[\cdot]$  is the function

$$F[E_0, \dots, E_q] = e^{-\beta[E_0, \dots, E_q]}. \quad (\text{A5})$$

A feature of divided differences is that they are invariant under rearrangement of the input values, so the input sequence forms a multiset, i.e., a generalization of the mathematical set which allows repetitions but where order does not play a role. The above infinite sum over energies may be simplified to

$$Z = \sum_z \sum_{q=0}^{\infty} \sum_{\{S_q\}} \langle z | S_q | z \rangle (-\Gamma)^q e^{-\beta[E_0, \dots, E_q]}. \quad (\text{A6})$$

Furthermore, the mean value theorem for divided differences [19,20] together with the monotonicity of the exponential function allows us to write

$$e^{-\beta[E_0, \dots, E_q]} = \frac{d^q (e^{-\beta E})}{dE^q} \Big|_{E=E_{(0, \dots, q)}} = \frac{(-\beta)^q e^{-\beta E_{(0, \dots, q)}}}{q!}. \quad (\text{A7})$$

We thus arrive at an expression for the partition function in terms of classical “effective energies” as

$$Z = \sum_{\{z\}} \sum_q \sum_{\{S_q\}} \frac{(\beta\Gamma)^q}{q!} \langle z | S_q | z \rangle e^{-\beta E_{(0, \dots, q)}}, \quad (\text{A8})$$

where  $E_{(0, \dots, q)}$  is the “effective classical energy” of the configuration, calculated from the multiset of energies  $[E_0, \dots, E_q]$  generated from  $\langle z | S_q | z \rangle$  and from  $\beta$ . The effective classical energy is a single real-valued energy  $E_{(0, \dots, q)} \in (\min[E_0, \dots, E_q], \max[E_0, \dots, E_q])$ . Specifically,  $E_{(0, \dots, q)}$  lies within the spectrum of the classical Hamiltonian.

To calculate  $E_{(0,\dots,q)}$ , one may use the divided differences recursion relation (see Appendix B) which in terms of effective classical energies becomes

$$\frac{(-\beta)^q}{q!} e^{-\beta E_{(0,\dots,q)}} = \frac{(-\beta)^{q-1}}{(q-1)!} \frac{(e^{-\beta E_{(0,\dots,q-1)}} - e^{-\beta E_{(1,\dots,q)}})}{E_0 - E_q}. \quad (\text{A9})$$

Isolating  $E_{(0,\dots,q)}$ , we arrive at

$$E_{(0,\dots,q)} = \bar{E} - \frac{1}{\beta} \log \frac{2q \sinh \beta \Delta E}{\beta(E_q - E_0)}, \quad (\text{A10})$$

where  $2\bar{E} = E_{(1,\dots,q)} + E_{(0,\dots,q-1)}$  and  $2\Delta E = E_{(1,\dots,q)} - E_{(0,\dots,q-1)}$ . In the limiting case where all energies in the sequence are equal, the above relation neatly becomes  $E_{(0,\dots,q)} = E_{(0)} = E_0$ . The initial condition for the above recursion is simply  $e^{-\beta E_{(i)}} = e^{-\beta E_i}$ . An illustration of how the recursion relation is used to calculate  $E_{(0,\dots,q)}$  is depicted in Fig. 2 and is discussed in more detail in Appendix C.

## APPENDIX B: NOTES ON DIVIDED DIFFERENCES

We provide below a brief summary of the concept of divided differences which is a recursive division process. This method is typically encountered when calculating the coefficients in the interpolation polynomial in the Newton form.

The divided difference [19,20] of a function  $F(\cdot)$  is defined as

$$F[x_0, \dots, x_q] \equiv \sum_{j=0}^q \frac{F(x_j)}{\prod_{k \neq j} (x_j - x_k)} \quad (\text{B1})$$

with respect to the list of real-valued input variables  $[x_0, \dots, x_q]$ . The above expression is ill defined if some of the inputs have repeated values, in which case one must resort to a limiting process. For instance, in the case where  $x_0 = x_1 = \dots = x_q = x$ , the definition of divided differences reduces to

$$F[x_0, \dots, x_q] = \frac{F^{(q)}(x)}{q!}, \quad (\text{B2})$$

where  $F^{(n)}(\cdot)$  stands for the  $n$ th derivative of  $F(\cdot)$ . Divided differences can alternatively be defined via the recursion relations

$$F[x_i, \dots, x_{i+j}] = \frac{F[x_{i+1}, \dots, x_{i+j}] - F[x_i, \dots, x_{i+j-1}]}{x_{i+j} - x_i}, \quad (\text{B3})$$

with  $i \in \{0, \dots, q-j\}$ ,  $j \in \{1, \dots, q\}$  with the initial conditions

$$F[x_i] = F(x_i), \quad i \in \{0, \dots, q\} \quad \forall i. \quad (\text{B4})$$

A function of divided differences can be defined in terms of its Taylor expansion. In the case where  $F(x) = e^{-\beta x}$ , we have

$$e^{-\beta[x_0, \dots, x_q]} = \sum_{n=0}^{\infty} \frac{(-\beta)^n [x_0, \dots, x_q]^n}{n!}. \quad (\text{B5})$$

Moreover, it is easy to verify that

$$[x_0, \dots, x_q]^{q+m} = \begin{cases} m < 0, & 0 \\ m = 0, & 1 \\ m > 0, & \sum_{\sum k_j = m} \prod_{j=0}^q x_j^{k_j}. \end{cases}$$

One may therefore write

$$\begin{aligned} e^{-\beta[x_0, \dots, x_q]} &= \sum_{n=0}^{\infty} \frac{(-\beta)^n [x_0, \dots, x_q]^n}{n!} \\ &= \sum_{n=q}^{\infty} \frac{(-\beta)^n [x_0, \dots, x_q]^n}{n!} \\ &= \sum_{m=0}^{\infty} \frac{(-\beta)^{q+m} [x_0, \dots, x_q]^{q+m}}{(q+m)!} \\ &= \sum_{m=0}^{\infty} \frac{(-\beta)^q}{(q+m)!} \sum_{\sum k_j = m} \prod_{j=0}^q (-\beta x_j)^{k_j} \quad (\text{B6}) \end{aligned}$$

as was asserted in Appendix A.

## APPENDIX C: EVALUATION OF THE GBWS: TECHNICAL DETAILS

The basic data structures we use to store the ODE configuration  $\mathcal{C} = (|z\rangle, S_q)$  are the classical configuration  $|z\rangle$ , which is an array of  $N$  bits, and the indices for the sequence of off-diagonal operators appearing in  $S_q$ . It is also useful to store the following:

- (i) The  $(q+1)$  labels and indices of the classical energies along the imaginary-time dimension.
- (ii) The multiplicity table of classical energies  $\{(m_j, E_j)\}$  counting the number of times each energy level appears.
- (iii) The pyramid: an ordered set of  $(q+1)(q+2)/2$  real-valued numbers. See Fig. 2 in the main text:

$$\begin{aligned} F[E_i, \dots, E_{i+j}] &= \frac{F[E_{i+1}, \dots, E_{i+j}] - F[E_i, \dots, E_{i+j-1}]}{E_{i+j} - E_i}. \quad (\text{C1}) \end{aligned}$$

### 1. The pyramid

As illustrated in Fig. 2 of the main text, the pyramid provides a convenient way to calculate the divided difference of  $e^{-\beta[E_0, \dots, E_q]}$  or, equivalently, the effective classical energy of the instantaneous configuration, namely,  $E_{(0,\dots,q)}$  or  $E_C$ . It relies on the recursive relation given in Eq. (6), namely,

$$E_{(0,\dots,q)} = \bar{E} - \frac{1}{\beta} \log \frac{2q \sinh \beta \Delta E}{\beta(E_q - E_0)}, \quad (\text{C2})$$

where

$$\begin{aligned} 2\bar{E} &= E_{(1,\dots,q)} + E_{(0,\dots,q-1)} \quad \text{and} \\ 2\Delta E &= E_{(1,\dots,q)} - E_{(0,\dots,q-1)}, \end{aligned}$$

with the initial conditions  $E_{(i)} = E_i$ . In the main text, we describe how the pyramid can be used to calculate the effective classical energy  $E_C$  associated with the instantaneous

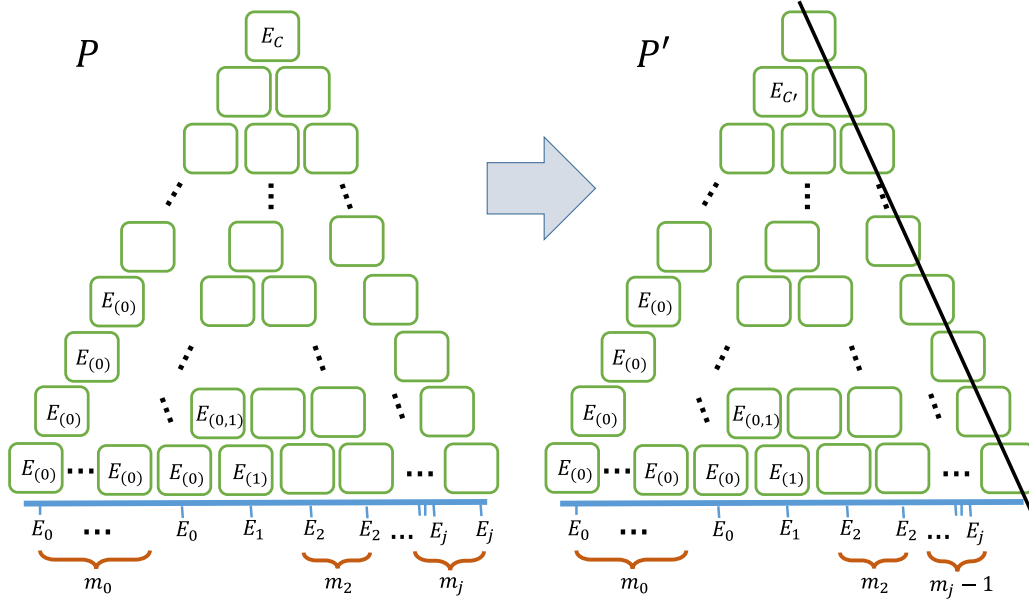


FIG. 12. Removal of a single energy level from the edge of the pyramid. The calculation of the effective classical energy upon a change  $M_C \rightarrow M_C - \{E(z')\}$  requires no calculations if the removed energy level is from the edge of the pyramid. In order to illustrate this, let us assume that the energy  $E(z') = E_j$  (with multiplicity  $m_j$ ) occurs at the rightmost edge of the pyramid  $P$ , as depicted in the left panel. The removal of this energy from the base of the pyramid  $P$  eliminates the rightmost element for all higher levels of the pyramid, including the topmost element, as depicted in the right panel. The new top element of the pyramid  $P'$  is the element labeled  $E_{C'}$ , which requires no further calculation since it is inherited from the pyramid  $P$ .

configuration  $C$ . The base of the pyramid has  $q + 1$  elements, corresponding to the “initial” energies  $E_i = E_i$  with  $i = 0 \dots q$ . These would be the classical energies  $E(z_i)$  of the intermediary classical states induced by the off-diagonal operators in  $S_q$  acting on  $|z\rangle$  sequentially. Let us denote this as level zero. Level one of the pyramid, which has  $q$  elements only, is now evaluated as follows. For each element at level one, we invoke the recursion relation above using the two elements below it (see Fig. 2 in the main text) at level zero, i.e.,

$$E_{(i,i+1)} = \frac{E_{(i)} + E_{(i+1)}}{2} - \frac{1}{\beta} \log \frac{2 \sinh \frac{\beta}{2} (E_{(i+1)} - E_{(i)})}{\beta (E_{(i+1)} - E_{(i)})}. \quad (\text{C3})$$

To avoid ill-defined ratios, we order the energies at level zero such that repeated values are grouped together. In this case, the evaluation of  $E_{(i,i+1)}$  for  $E_{(i)} = E_{(i+1)}$  gives  $E_{(i,i+1)} = E_{(i)}$ . Similarly, level two elements are calculated via

$$E_{(i-1,i,i+1)} = \frac{E_{(i-1,i)} + E_{(i,i+1)}}{2} - \frac{1}{\beta} \log \frac{4 \sinh \frac{\beta}{2} (E_{(i-1,i)} - E_{(i,i+1)})}{\beta (E_{(i+1)} - E_{(i-1)})}. \quad (\text{C4})$$

This procedure can be continued until the top level (level  $q$ ) of the pyramid is reached, which gives the value of  $E_C = E_{(0,\dots,q)}$  the effective classical energy of the configuration, from which the GBW is calculated via Eq. (8).

## 2. Virtual vs actual moves

Naively, calculating the value of a GBW, or equivalently the effective classical energy of a configuration, requires  $(q + 1)(q + 2)/2$  operations as the number of blocks in the pyramid. However, small changes to an already evaluated pyramid generate a new pyramid whose GBW is easier to evaluate. For instance, the GBW associated with the removal of a single energy value  $M_C \rightarrow M_C - \{E(z')\}$  requires no calculations if the to-be-removed energy  $E(z')$  appears at an outer edge of the base level. This is illustrated in Fig. 12. (It can be shown that similar tricks may be applied even if the energy level to be removed is from the “bulk” of the pyramid.) Similarly, the addition of a single energy level often requires only  $O(q)$  operations.

A key property of the divided difference of a function is that it is invariant under reordering of the input values. In the context of our “pyramid scheme” of calculating the divided difference, this means that while the ordering of groups of identical energies in the multiplicity table changes the intermediate values of the pyramid, it does not change the value of the topmost level of the pyramid. Therefore, by manipulating the ordering of the energies such that local changes to the multiset of energies (as occurs for the local swap, block swap and annihilation and creation moves described in the main text) occur at the edges of the pyramid, it is possible to minimize the number of computations needed to determine the topmost level of the pyramid from  $O(q^2)$  to  $O(q)$  or  $O(1)$ . This allows us to calculate the weights of proposed changes more efficiently than recalculating the entire pyramid, although this procedure may leave some elements of the pyramid undefined. These virtual moves are highly useful both for updates as well as in measurement steps where virtual

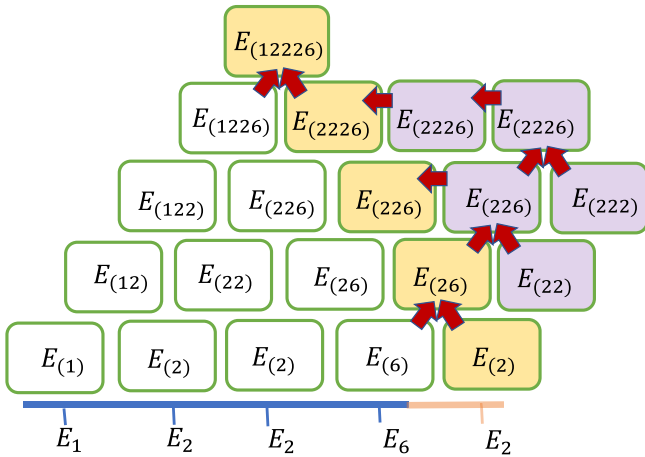


FIG. 13. Addition of an energy level to the pyramid. The multiplicity table is such that the energy  $E_i$  appears  $m_i$  times (the multiplicity of the energy  $E_i$ ), with  $\sum_i m_i = q + 1$ . Let us consider that a proposed move changes the multiplicity of the energy  $E_2$ , i.e.,  $m_2 \rightarrow m_2 + 1$ . In order to determine the pyramid  $P'$  associated with this change, we perform the following manipulations to the original pyramid  $P$ . The subpyramid associated with  $E_2, \dots, E_j$  can be “flipped” such that the energy  $E_2$  appears at the edge of  $P$  (dark-colored blocks on left panel). Because this manipulation does not change the base of the subpyramid nor that of the entire  $P$ , the values at the top level of subpyramid and  $P$  remain unaffected (the unaffected blocks are the empty blocks on the right panel). However, this move does change the values of other elements in  $P$  (purple colored blocks), but we will not need to calculate them. Because  $E_2$  now appears at the edge of  $P$ , introducing an additional  $E_2$  to the base of  $P$  (to generate  $P'$ ) requires us to recalculate the new elements that appear at the edge of  $P'$ , namely, only  $O(q)$  operations.

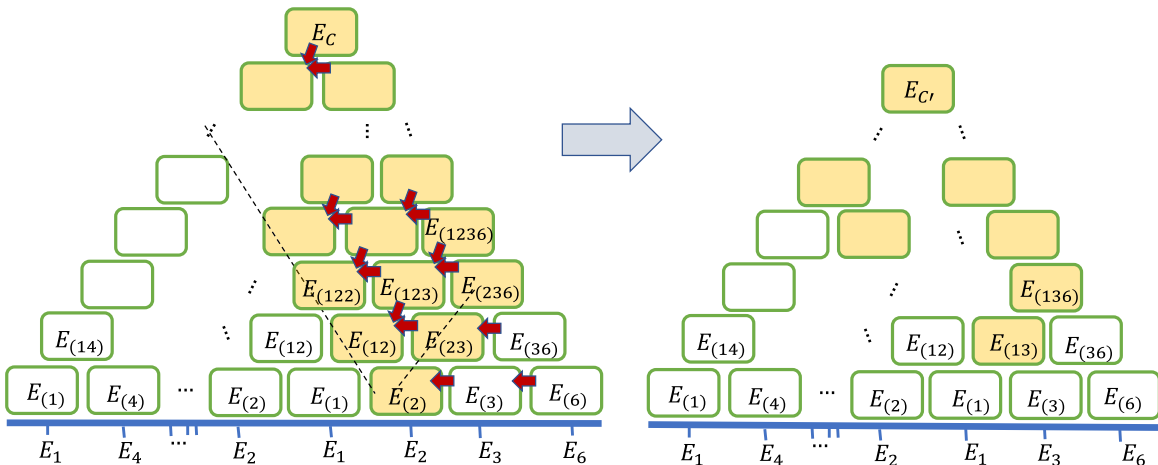


FIG. 14. Removal of an energy level from the bulk of the pyramid. The multiplicity table is such that the energy  $E_i$  appears  $m_i$  times (the multiplicity of the energy  $E_i$ ), with  $\sum_i m_i = q + 1$ . Let us consider that a proposed move changes the multiplicity of the energy  $E_2$ , i.e.,  $m_2 \rightarrow m_2 + 1$ . In order to determine the pyramid  $P'$  associated with this change, we perform the following manipulations to the original pyramid  $P$ . The subpyramid associated with  $E_2, \dots, E_j$  can be “flipped” such that the energy  $E_2$  appears at the edge of  $P$  (dark-colored blocks on left panel). Because this manipulation does not change the base of the subpyramid nor that of the entire  $P$ , the values at the top level of subpyramid and  $P$  remain unaffected (the unaffected blocks are the empty blocks on the right panel). However, this move does change the values of other elements in  $P$  (purple colored blocks), but we will not need to calculate them. Because  $E_2$  now appears at the edge of  $P$ , introducing an additional  $E_2$  to the base of  $P$  (to generate  $P'$ ) requires us to recalculate the new elements that appear at the edge of  $P'$ , namely, only  $O(q)$  operations.

TABLE I. Computational complexity of virtual updates for local changes to the multiplicity table. While naively the calculation of a GBW requires  $O(q^2)$  operations, changes to the GBW can be carried out much more efficiently if the multiplicity table is only locally perturbed.

Update	Change to the energy multiset $M_C$	Computational complexity (worst case)
Local swap	$M_C \rightarrow M_C + \{E(z_i)\} - \{E(z_j)\}$	$O(1)$
Block swap	$M_C \rightarrow M_C + \{E(z_i)\} - \{E(z_j)\}$	$O(1)$
Pair creation	$M_C \rightarrow M_C + \{E(z_i), E(z_j)\}$	$O(q)$
Pair annihilation	$M_C \rightarrow M_C - \{E(z_i), E(z_j)\}$	$O(q)$
$\langle V_k \rangle$ measurement	$M_C \rightarrow M_C - \{E(z)\}$	$O(q)$

rotations of  $S_q$  are useful. Only if the move is accepted do we need to calculate these “missing,” or unevaluated, elements of the pyramid. We call this process a “virtual move.” We illustrate two such procedures in Figs. 13 and 14.

The computational complexity associated with calculating the changes to the effective classical energy (equivalently, the change to the GBW) due to the local updates and measurements discussed in the main text is summarized in Table I.

### 3. Precision issues

The calculation of the effective classical energy  $E_C$  and GBWs using the recursion scheme described above may require recursive operations on pairs of numbers of approximately equal magnitude whose difference is an order of magnitude closer to zero. If we restrict ourselves to a fixed bit-precision representation of the numbers, the calculated difference may be erroneous because of truncation errors. In order to avoid this problem, it is necessary to check



periodically whether an increase in the bit precision used changes the result. In our simulations, we initially use 53 bits for the significand (also known as mantissa), which is the number used for “double” precision in the ANSI/IEEE-754 standard, but increase the number of bits by factors of 1.2 when necessary using the GNU Multiple-precision Binary Floating-point Library with Correct Rounding library [51].

#### APPENDIX D: CALCULATION OF $N_p(q)$ [EQ. (15)]

The calculation of  $N_p(q)$ , the number of  $S_q$  sequences comprised of  $q$  off-diagonal  $\sigma_i^x$  operators ( $i = 1 \dots N$ ) such that each operator appears an even number of times is carried out as follows [52]. By definition,

$$N_p(q) = \sum_{\sum_{i=1}^N k_i = q, k_i \text{ even}} \binom{n}{k_1 k_2 \dots k_N}, \quad (\text{D1})$$

where  $k_i$  is the number of times that operator  $\sigma_i^x$  appears in the sequence. We note that  $\binom{n}{k_1 k_2 \dots k_N}$  is the coefficient of  $x_1^{k_1} x_2^{k_2} \dots x_N^{k_N}$  in the expansion of  $(x_1 + x_2 + \dots + x_N)^q$ . The sum of all these coefficients is obtained by substituting  $x_1 = x_2 = \dots = x_N$ . To eliminate odd powers  $k_1$ , we can consider the expansion of

$$\frac{1}{2}[(x_1 + x_2 + \dots + x_N)^q + (-x_1 + x_2 + \dots + x_N)^q]. \quad (\text{D2})$$

Continuing this way, we eventually arrive at

$$N_p(q) = \frac{1}{2^N} \sum_{t_i=0,1} [(-1)^{t_1} + (-1)^{t_2} + \dots + (-1)^{t_N}]^q, \quad (\text{D3})$$

which can be further simplified to

$$N_p(q) = \frac{1}{2^N} \sum_{k=0}^N \binom{N}{k} (N - 2k)^q. \quad (\text{D4})$$

- 
- [1] D. Landau and K. Binder, *A Guide to Monte Carlo Simulations in Statistical Physics* (Cambridge University Press, New York, 2005).
- [2] M. N. G. Barkema, *Monte Carlo Methods in Statistical Physics* (Oxford University Press, Oxford, 1999).
- [3] S. R. White, *Phys. Rev. Lett.* **69**, 2863 (1992).
- [4] A. W. Sandvik, *J. Phys. A: Math Gen.* **25**, 3667 (1992).
- [5] A. W. Sandvik, *Phys. Rev. B* **59**, R14157 (1999).
- [6] N. V. Prokof'ev, B. V. Svistunov, and I. S. Tupitsyn, *J. Exp. Theor. Phys.* **87**, 310 (1998).
- [7] A. W. Sandvik, *Phys. Rev. E* **68**, 056701 (2003).
- [8] E. Y. Loh, J. E. Gubernatis, R. T. Scalettar, S. R. White, D. J. Scalapino, and R. L. Sugar, *Phys. Rev. B* **41**, 9301 (1990).
- [9] M. Troyer and U.-J. Wiese, *Phys. Rev. Lett.* **94**, 170201 (2005).
- [10] P. Pfeuty and R. J. Elliott, *J. Phys. C: Solid State Phys.* **4**, 2370 (1971).
- [11] R. B. Stinchcombe, *J. Phys. C: Solid State Phys.* **6**, 2459 (1973).
- [12] B. K. Chakrabarti, *Phys. Rev. B* **24**, 4062 (1981).
- [13] E. Farhi, D. Gosset, I. Hen, A. W. Sandvik, P. Shor, A. P. Young, and F. Zamponi, *Phys. Rev. A* **86**, 052334 (2012).
- [14] S. Boixo, T. F. Ronnow, S. V. Isakov, Z. Wang, D. Wecker, D. A. Lidar, J. M. Martinis, and M. Troyer, *Nat. Phys.* **10**, 218 (2014).
- [15] A. W. Sandvik, *Phys. Rev. Lett.* **95**, 207203 (2005).
- [16] A. W. Sandvik and H. G. Evertz, *Phys. Rev. B* **82**, 024407 (2010).
- [17] D. C. Handscomb, *Proc. Cambridge Philos. Soc.* **58**, 594 (1962).
- [18] D. C. Handscomb, *Proc. Cambridge Philos. Soc.* **60**, 115122 (1964).
- [19] E. T. Whittaker and G. Robinson, in *The Calculus of Observations: A Treatise on Numerical Mathematics* (Dover, New York, 1967).
- [20] C. de Boer, *Surveys Approx. Theory* **1**, 46 (2005).
- [21] S. Bravyi, D. P. DiVincenzo, R. I. Oliveira, and B. M. Terhal, *Quantum Inf. Comput.* **8**, 0361 (2008).
- [22] S. Bravyi and M. Hastings, *Commun. Math. Phys.* **349**, 1 (2107).
- [23] A. Dorneich and M. Troyer, *Phys. Rev. E* **64**, 066701 (2001).
- [24] U. Wolff, *Phys. Rev. Lett.* **62**, 361 (1989).
- [25] R. Martoňák, G. E. Santoro, and E. Tosatti, *Phys. Rev. B* **66**, 094203 (2002).
- [26] C.-W. Liu, A. Polkovnikov, and A. W. Sandvik, *Phys. Rev. Lett.* **114**, 147203 (2015).
- [27] K. Hukushima and K. Nemoto, *J. Phys. Soc. Jpn.* **65**, 1604 (1996).
- [28] E. Marinari, in *Advances in Computer Simulation*, edited by J. Kertész and I. Kondor (Springer, Berlin, 1998), p. 50.
- [29] P. Sengupta, A. W. Sandvik, and D. K. Campbell, *Phys. Rev. B* **65**, 155113 (2002).
- [30] I. Hen and A. P. Young, *Phys. Rev. E* **84**, 061152 (2011).
- [31] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, *Science* **220**, 671 (1983).
- [32] A. Sokal, in *Functional Integration: Basics and Applications*, edited by C. DeWitt-Morette, P. Cartier, and A. Folacci (Plenum, New York, 1997).
- [33] M. W. Johnson, M. H. S. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, R. Harris, A. J. Berkley, J. Johansson, P. Bunyk, E. M. Chapple, C. Enderud, J. P. Hilton, K. Karimi, E. Ladizinsky, N. Ladizinsky, T. Oh, I. Perminov, C. Rich, M. C. Thom *et al.*, *Nature (London)* **473**, 194 (2011).
- [34] The simulations were performed on individual cores of Intel Xeon E5-2640 CPUs running at 2.60 GHz.
- [35] S. Knysh, *Nat. Commun.* **7**, 12370 (2016).
- [36] B. Apolloni, C. Carvalho, and D. de Falco, *Stochastic Proc. Applicat.* **33**, 233 (1989).
- [37] B. Apolloni, N. Cesa-Bianchi, and D. de Falco, in *Proceedings of the Ascona/Locarno Conference* (World Scientific, Singapore, 1988), p. 97.
- [38] R. L. Somorjai, *J. Phys. Chem.* **95**, 4141 (1991).
- [39] P. Amara, D. Hsu, and J. E. Straub, *J. Phys. Chem.* **97**, 6715 (1993).
- [40] A. B. Finnila, M. A. Gomez, C. Sebenik, C. Stenson, and J. D. Doll, *Chem. Phys. Lett.* **219**, 343 (1994).
- [41] T. Kadowaki and H. Nishimori, *Phys. Rev. E* **58**, 5355 (1998).
- [42] J. Brooke, D. Bitko, T. F. Rosenbaum, and G. Aeppli, *Science* **284**, 779 (1999).
- [43] J. Brooke, T. F. Rosenbaum, and G. Aeppli, *Nature (London)* **413**, 610 (2001).
- [44] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, *arXiv:quant-ph/0001106*.

- [45] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, and D. Preda, *Science* **292**, 472 (2001).
- [46] B. Heim, T. F. Rønnow, S. V. Isakov, and M. Troyer, *Science* **348**, 215 (2015).
- [47] G. E. Santoro, R. Martoňák, E. Tosatti, and R. Car, *Science* **295**, 2427 (2002).
- [48] A. P. Young, S. Knysh, and V. N. Smelyanskiy, *Phys. Rev. Lett.* **101**, 170503 (2008).
- [49] I. Hen, *Phys. Rev. E* **85**, 036705 (2012).
- [50] T. Albash, S. Boixo, D. A. Lidar, and P. Zanardi, *New J. Phys.* **14**, 123016 (2012).
- [51] L. Fousse, G. Hanrot, V. Lefèvre, P. Pélicier, and P. Zimmermann, *ACM Trans. Math. Software* **33**, 1 (2007).
- [52] M. Alekseyev (<http://mathoverflow.net/users/7076/maxalekseyev>), What is this restricted sum of multinomial coefficients? MathOverflow, URL:<http://mathoverflow.net/q/73616> (version: 2011-08-24), <http://mathoverflow.net/q/73616>.