Quantum Monte Carlo with directed loops

Olav F. Syljuåsen[1,*] and Anders W. Sandvik[2,†]
[1]*NORDITA, Blegdamsvej 17, DK-2100 Copenhagen Ø, Denmark*
[2]*Department of Physics, Åbo Akademi University, Porthansgatan 3, FIN-20500 Turku, Finland*
(Received 19 February 2002; revised manuscript received 20 May 2002; published 2 October 2002)

We introduce the concept of directed loops in stochastic series expansion and path-integral quantum Monte Carlo methods. Using the detailed balance rules for directed loops, we show that it is possible to smoothly connect generally applicable simulation schemes (in which it is necessary to include backtracking processes in the loop construction) to more restricted loop algorithms that can be constructed only for a limited range of Hamiltonians (where backtracking can be avoided). The ''algorithmic discontinuities'' between general and special points (or regions) in parameter space can hence be eliminated. As a specific example, we consider the anisotropic $S=1/2$ Heisenberg antiferromagnet in an external magnetic field. We show that directed-loop simulations are very efficient for the full range of magnetic fields (zero to the saturation point) and anisotropies. In particular, for weak fields and anisotropies, the autocorrelations are significantly reduced relative to those of previous approaches. The back-tracking probability vanishes continuously as the isotropic Heisenberg point is approached. For the XY model, we show that back tracking can be avoided for all fields extending up to the saturation field. The method is hence particularly efficient in this case. We use directed-loop simulations to study the magnetization process in the two-dimensional Heisenberg model at very low temperatures. For $L\times L$ lattices with L up to 64, we utilize the step structure in the magnetization curve to extract gaps between different spin sectors. Finite-size scaling of the gaps gives an accurate estimate of the transverse susceptibility in the thermodynamic limit: $\chi_\perp = 0.0659 \pm 0.0002$.

DOI: 10.1103/PhysRevE.66.046701 PACS number(s): 02.70.Ss, 05.30.−d, 75.10.Jm, 75.40.Mg

I. INTRODUCTION

In recent years, significant advances in quantum Monte Carlo (QMC) algorithms have opened up several classes of quantum many-body models to the kind of large-scale numerical studies that were previously possible only for classical systems. The progress has been along two main lines: (i) the elimination [1–5] of the systematic error of the Trotter decomposition [6] on which most of the early finite-temperature QMC algorithms [7–11] were based (with the exception of Handscomb's method [12–15], the utility of which was limited), and (ii) the development of loop-cluster algorithms [16] for efficient sampling in the quantum mechanical configuration space [3,4,17–19]. Algorithms incorporating both (i) and (ii) have been devised starting from either the Euclidean path integral (world-line QMC methods operating in continuous imaginary time [3,4]) or the power series expansion of the partition function (stochastic series expansion, hereafter SSE [18], which is an extension of Handscomb's method). While the Trotter error is a controllable one and can be eliminated in standard approaches by extrapolating results for different imaginary time discretizations to the continuum [6,20], its absence directly at the level of the simulation can imply considerable time savings when unbiased results are needed, e.g., in finite-size scaling studies. The loop-cluster algorithms (world-line loops [16,17,19], SSE operator loops [18], and world-line worms [3]) have offered even more dramatic speed-ups, in many cases reducing autocorrelation times by several orders of magnitude and thus enabling studies of system sizes much larger than what was possible with local sampling algorithms. In addition, in some special cases, fermionic and other sign problems can be eliminated with the loop-cluster algorithms [21–23].

The new QMC methods have become important tools for quantum many-body research in condensed matter physics (with applications to quantum spins [24–35], bosons [36–38], and one-dimensional fermion systems [39,40]) as well as in lattice gauge theory [21,22]. An important property of some of the loop-cluster algorithms is that they are efficient also in the presence of external fields [18,19,22,41]. In particular, the SSE algorithm with the operator-loop update [18] has proven very powerful in several recent studies of quantum spin systems [33–35] and boson systems [36–38] including, respectively, a magnetic field and a chemical potential. It is interesting to note that in this respect QMC algorithms now perform better than classical Monte Carlo, since in the latter case external fields still pose challenging problems.

In this paper we present a general framework for constructing loop-type algorithms both in SSE and path-integral methods. We focus primarily on the SSE approach, which owing to the manifestly discrete nature of its configuration space is easier to implement and, for the same reason, also is more efficient in most cases. In the SSE operator-loop update introduced in Ref. [18], a distinction was made between a general algorithm (where it is necessary to allow the propagating end of the operator path to backtrack) and special ones applicable only for certain Hamiltonians (where the paths do not backtrack). For example, in the case of the $S=1/2$ Heisenberg model with uniaxial anisotropy Δ and external magnetic field h (also known as the XXZ model),

*Electronic address: sylju@nordita.dk
†Electronic address: asandvik@abo.fi

$$H = J \sum_{\langle i,j \rangle} [S_i^x S_j^x + S_i^y S_j^y + \Delta S_i^z S_j^z] - h \sum_i S_i^z , \qquad (1)$$

particularly efficient algorithms were devised at the isotropic Heisenberg point ($\Delta = 1, h = 0$) and for the XY model ($\Delta = 0, h = 0$). While the general algorithm can be used for any Δ, h, it does not perform as well in the limits $\Delta \to 1, h \to 0$ and $\Delta \to 0, h \to 0$ as the special algorithms exactly at these points (which are the only points at which the more efficient algorithms can be used). Hence, one should switch algorithms when crossing the isotropic Heisenberg and XY points. The presence of such ''algorithmic discontinuities'' is clearly bothersome, both from a mathematical and practical point of view. Here we show how the algorithmic discontinuities can be eliminated within a more general framework of satisfying detailed balance when constructing the operator loop. For reasons that will become clear below, we call the entities involved in this type of update ''directed loops.'' With these, we are able to carry out simulations as efficiently in the limits approaching the Heisenberg and XY points as exactly at those points. We also show that this scheme can be easily adapted to continuous-time path integrals.

The outline of the rest of the paper is the following: In Sec. II we review the SSE method and the operator-loop update on which the directed-loop algorithm is based. We outline a proof of detailed balance and also discuss a few special cases in which back tracking can be easily avoided in the loop construction. In Sec. III we first discuss a more general condition for satisfying detailed balance in the SSE method, which leads us to the directed-loop equations. We then show in detail how this scheme is applied to the $S = 1/2$ XXZ model. We present two solutions of the directed-loop equations. One is identical to the previous generic operator-loop update and the other smoothly connects to the special ''deterministic'' operator-loop algorithm at the isotropic Heisenberg point. We also briefly discuss the structure of the directed-loop equations for a more general class of Hamiltonians. Implementation of directed loops in the path-integral formalism is discussed in Sec. IV. In Sec. V we present simulation results in various parameter regions of the XXZ model. We compare autocorrelation times for simulations using the two different directed-loop solutions. We also extract the dynamic exponent in simulations of isotropic Heisenberg models at critical points in one, two, and three dimensions. In Sec. VI, as a demonstration of what can be accomplished with the improved solution, we present results for the magnetization as a function of the external field in the two-dimensional (2D) Heisenberg model at very low temperatures. We calculate the magnetic susceptibility using gaps between different spin sectors extracted from the step structure in the magnetization curve. We conclude with a summary and discussion in Sec. VII. In an Appendix we outline the basic elements of a simple and efficient computer implementation of the SSE method.

II. STOCHASTIC SERIES EXPANSION

The SSE method is a generalization [1,2,18] of Handscomb's power series expansion method for the isotropic $S = 1/2$ Heisenberg ferromagnet [12] and antiferromagnet [13,14] to a much wider range of systems. The performance is significantly improved also for the Heisenberg model [27,28,42]. Early attempts of such generalizations [15] were limited by the difficulties in analytically evaluating the traces of the terms of the expansion. This problem was solved [1,2] by the development of a scheme for importance sampling also of the individual terms of the traces expressed in a conveniently chosen basis. The starting point of the SSE method is hence the power series expansion of the partition function:

$$Z = \mathrm{Tr}\{e^{-\beta H}\} = \sum_\alpha \sum_{n=0}^{\infty} \frac{(-\beta)^n}{n!} \langle \alpha | H^n | \alpha \rangle , \qquad (2)$$

where the trace has been written as a sum over diagonal matrix elements in a basis $\{|\alpha\rangle\}$. Simulation algorithms based on this expansion can be formulated without sign problems for the same models as those for which world-line methods [9] are applicable. There are no approximations causing systematic errors, and very efficient loop-type updating algorithms have also recently been devised [18,23,40,43]. A distinct advantage of SSE over continuous-time world-line methods [3,4] is the discrete nature of the configuration space, which can be sampled without floating point operations.

Here we first review an implementation of the SSE method for the anisotropic $S = 1/2$ Heisenberg model. A proof of detailed balance in the operator-loop updating scheme is then outlined. Several practical issues related to the operator loops are also discussed. Estimators for physical observables will not be discussed here. Several classes of expectation values have been derived in Ref. [2]. Observables of interest in the context of the Heisenberg model have been discussed in Ref. [42]. Off-diagonal correlation functions (single-particle Green's functions) have been studied in Ref. [44].

A. SSE configuration space

For the anisotropic Heisenberg antiferromagnet, Eq. (1), with N spins it is convenient to use the standard basis

$$|\alpha\rangle = |S_1^z, S_2^z, \ldots, S_N^z\rangle , \qquad (3)$$

and to write the Hamiltonian in terms of bond operators H_b, where b refers to a pair of sites $i(b), j(b)$,

$$H = -J \sum_{b=1}^{N_b} H_b \quad (J > 0). \qquad (4)$$

For a d-dimensional cubic lattice, the number of bonds $N_b = dN$. The bond operators are further decomposed into two operators:

$$H_b = H_{1,b} - H_{2,b} , \qquad (5)$$

where $H_{1,b}$ is diagonal and $H_{2,b}$ off-diagonal,

$$H_{1,b} = C - \Delta S_{i(b)}^z S_{j(b)}^z + h_b [S_{i(b)}^z + S_{j(b)}^z], \qquad (6)$$

$$H_{2,b}=\frac{1}{2}[S^{+}_{i(b)}S^{-}_{j(b)}+S^{-}_{i(b)}S^{+}_{j(b)}], \tag{7}$$

and we have defined the magnetic field on a bond, $h_b \equiv h/(2dJ)$. The constant C should be chosen such that all matrix elements of $H_{1,b}$ are positive, i.e., $C \geqslant \Delta/4+h_b$. We will henceforth use the notation

$$C=C_0+\epsilon, \quad C_0=\Delta/4+h_b, \tag{8}$$

where $\epsilon \geqslant 0$. In the Hamiltonian (4) we have neglected a constant $N_b C$, which should be kept in mind when calculating the energy.

The powers of H in Eq. (2) can be expressed as sums of products of the bond operators (6) and (7). Such a product is conveniently referred to by an operator-index sequence

$$S_n=[a_1,b_1],[a_2,b_2], \ldots ,[a_n,b_n], \tag{9}$$

where $a_i \in \{1,2\}$ corresponds to the type of operator (1, diagonal; 2, off-diagonal) and $b_i \in \{1, \ldots ,N_b\}$ is the bond index. Hence,

$$Z=\sum_{\alpha}\sum_{n=0}^{\infty}\sum_{S_n}(-1)^{n_2}\frac{\beta^n}{n!}\left\langle \alpha \left| \prod_{i=1}^{n} H_{a_i,b_i} \right| \alpha \right\rangle, \tag{10}$$

where $\beta \equiv J/T$ and n_2 is the total number of spin-flipping operators $[2,b]$ in S_n. It is useful to define the normalized states resulting when $|\alpha\rangle$ is propagated by a fraction of the SSE operator string:

$$|\alpha(p)\rangle \sim \prod_{i=1}^{p} H_{a_i,b_i}|\alpha\rangle. \tag{11}$$

Note that there is no branching, i.e., all $|\alpha(p)\rangle$ are basis states, and $|\alpha(p)\rangle$ and $|\alpha(p+1)\rangle$ are either same states or differ only by a flipped pair of spins. In order for a term (α,S_n) to contribute to the partition function, the boundary condition $|\alpha(n)\rangle=|\alpha(0)\rangle$ has to be satisfied. On a bipartite lattice n_2 must therefore be even, and the expansion is then positive definite. The terms (configurations) can thus be sampled using Monte Carlo techniques without sign problems.

To simplify the Monte Carlo sampling, it is useful [1] (although not necessary [2]) to truncate the expansion at a maximum power $n=M$ and to insert $M-n$ ''fill-in'' unit operators $H_{0,0} \equiv 1$ in the operator products in all possible ways. This gives

$$Z=\sum_{\alpha}\sum_{S_M}\frac{\beta^n(M-n)!}{M!}\left\langle \alpha \left| \prod_{i=1}^{M} H_{a_i,b_i} \right| \alpha \right\rangle, \tag{12}$$

where n is the number of bond operators, i.e., the number of elements $[a_i,b_i] \neq [0,0]$. One can show that [1,12] the average expansion order

$$\langle n \rangle = \beta N_b |E_b|, \tag{13}$$

where E_b is the internal energy per bond, $E_b=-\langle H_b \rangle$ [including the constant C in Eq. (6)], and that the width of the distribution is approximately $\langle n \rangle^{1/2}$. M can therefore be chosen so that n never reaches the cutoff during the simulation ($M \sim \beta N$). The truncation error is then completely negligible. In practice, M is gradually adjusted during the equilibration of the simulation, so that $M=an_{\max}$, where $n_{\max}$ is the highest n reached. A practical range for the factor a is 1.2–1.5. The simulation can be started with some random state $|\alpha\rangle$ and an ''empty'' operator string $[0,0]_1, \ldots ,[0,0]_M$ (we sometimes use the notation $[a,b]_p$ instead of $[a_p,b_p]$). Ergodic sampling of the configurations (α,S_n) is accomplished using two different types of updates.

B. Updating scheme

The first update (diagonal update) is of the type $[0,0]_p \leftrightarrow [1,b]_p$, involving a single diagonal operator which changes the expansion order n by ± 1 [42]. The corresponding Metropolis acceptance probabilities are

$$P([0,0]_p \rightarrow [1,b]_p)=\frac{N_b \beta \langle \alpha(p)|H_{1,b}|\alpha(p)\rangle}{M-n}, \tag{14}$$

$$P([1,b]_p \rightarrow [0,0]_p)=\frac{M-n+1}{N_b \beta \langle \alpha(p)|H_{1,b}|\alpha(p)\rangle}, \tag{15}$$

where $P>1$ should be interpreted as probability 1. The presence of N_b in these probabilities reflects the fact that there are N_b random choices for the bond b in a substitution $[0,0] \rightarrow [1,b]$ but only one way to replace $[1,b] \rightarrow [0,0]$ when b is given. These diagonal updates are attempted consecutively for all $p=1, \ldots ,M$, and at the same time the state $|\alpha\rangle$ is propagated when spin-flipping operators $[2,b]$ are encountered (these cannot be changed in a single-operator update), so that the states $|\alpha(p)\rangle$ are available when needed to calculate the probabilities (14) and (15).

The purpose of the second type of update—the operator loop [18]—is to accomplish substitutions $[1,b]_p \leftrightarrow [2,b]_p$ for a varying number of operators, thereby flipping spins also in several of the propagated states (11). The expansion order n does not change. It is then convenient to disregard the $[0,0]$ unit operator elements in S_M and instead work with the original sequences S_n of Eq. (10), which contain only elements $[1,b]$ and $[2,b]$. For the discussion of the operator loops, the propagation index p will refer to this reduced sequence. It is also convenient to introduce two-spin states

$$|\alpha_{b_p}(p)\rangle = |S^z_{i(b_p)}(p),S^z_{j(b_p)}(p)\rangle, \tag{16}$$

i.e., the spins at bond b_p in the propagated state $|\alpha(p)\rangle$ as defined in Eq. (11). The weight factor corresponding to Eq. (10) can then be written as

$$W(\alpha,S_n)=\frac{\beta^n}{n!}\prod_{p=1}^{n}\langle \alpha_{b_p}(p)|H_{b_p}|\alpha_{b_p}(p-1)\rangle, \tag{17}$$

where the nonzero two-spin matrix elements are

$$\langle \downarrow\downarrow|H_b|\downarrow\downarrow\rangle=\epsilon,$$

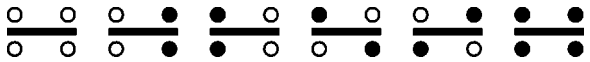

FIG. 1. The six different vertices corresponding to the matrix elements in Eqs. (18). The horizontal bar represents the full bond operator H_b and the circles beneath (above) represent the spin state (solid and open circles for spin-$\uparrow$ and spin-$\downarrow$, respectively) before (after) operation with either the diagonal or off-diagonal part of H_b.

$$\langle\downarrow\uparrow|H_b|\downarrow\uparrow\rangle=\langle\uparrow\downarrow|H_b|\uparrow\downarrow\rangle=\Delta/2+h_b+\epsilon,$$
$$\langle\uparrow\downarrow|H_b|\downarrow\uparrow\rangle=\langle\downarrow\uparrow|H_b|\uparrow\downarrow\rangle=1/2, \tag{18}$$
$$\langle\uparrow\uparrow|H_b|\uparrow\uparrow\rangle=\epsilon+2h_b.$$

In principle, the value of $\epsilon\geqslant 0$ is arbitrary, but in practice a large constant is inconvenient since the average expansion order (13) has a contribution $\epsilon\beta N_b$. In many cases the simulation performs better with a small $\epsilon>0$ than with $\epsilon=0$, however, as will be demonstrated in Sec. V. For $\epsilon=0$, the number of allowed matrix elements is reduced from six to four (if $h=0$) or five (if $h>0$).

The matrix element product in the weight (17) can be represented as a network of n vertices, with two spins $S_i^z(p-1),S_j^z(p-1)$ "entering" the pth vertex and $S_i^z(p),S_j^z(p)$ "exiting." The six allowed vertices, corresponding to the nonzero matrix elements (18), are illustrated in Fig. 1. The direction of propagation (here and in other illustrations) is such that moving upward corresponds to increasing the propagation index p.

In order to carry out the operator-loop update, a linked list of the vertices is first constructed. For each of the four legs on each vertex there is a spin state and a link to the following (in the direction of increasing p) or previous (direction of decreasing p) vertex leg at the same site. The periodic boundary condition of the propagated states must be taken into account, i.e., the links can span across $p=0$ and every leg then has an outgoing and incoming link (i.e., a bidirectional link). In case a spin (site) is acted upon only by a single operator in S_n, the corresponding two legs of that vertex are linked together. Otherwise, for a site acted upon by two or more operators, all links are between different vertices. An example of an SSE configuration and its corresponding linked vertex list is shown in Fig. 2. Clearly, in an allowed configuration, links can exist only between legs in the same spin state. Note that in the representation with the

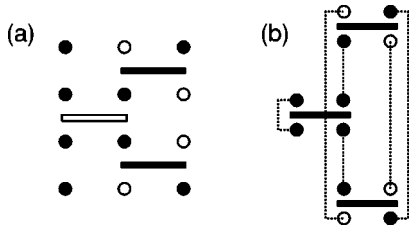

FIG. 2. (a) An SSE configuration for a three-site system with three operators, shown along with all the propagated states. Here open and solid bars indicate diagonal and off-diagonal operators, respectively. (b) The linked vertex list corresponding to (a). The dashed lines represent bidirectional links.

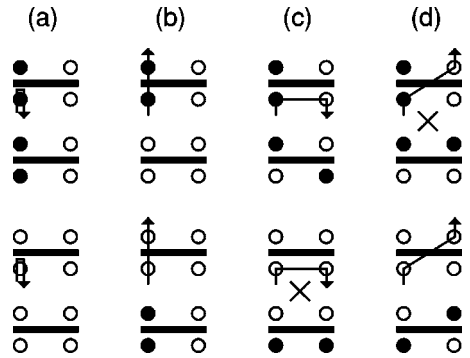

FIG. 3. All four paths through two vertices where the entrance is at the lower left leg. The arrow indicates the exit leg. The resulting updated vertices, with the spin at the entrance and exit legs flipped, are also shown. The two cases marked with an X are forbidden, since the updated vertices do not correspond to operators in the Hamiltonian considered here. We refer to the four different processes as (a) "bounce," (b) "continue-straight," (c) "switch-and-reverse," and (d) "switch-and-continue."

full states in (a), which is not stored in the actual simulation but is included here for illustrative purposes, we distinguish between diagonal and off-diagonal operators (as is also done in the stored operator sequence S_M used in the diagonal update). In the vertex representation (b) the two-spin states are taken from the full propagated states (16), and the type of the operator (diagonal or off-diagonal) is implicitly given by the four spin states. The bar is hence strictly redundant, but we include it in the figures as a reminder that the vertices represent matrix elements of the bond operators.

To construct an operator loop, one of the $4n$ vertex legs is first selected at random as an initial entrance leg. One of the four legs belonging to the same vertex as the entrance leg is then chosen as the exit from the vertex, and both the entrance and exit spins are flipped. Examples of how vertices change in the four types of processes are shown in Fig. 3. The probability of exiting at a given leg, given the entrance leg and the four spin states defining the vertex, is taken proportional to that matrix element in Eq. (18) which corresponds to the vertex generated when the entrance and exit spins have been flipped. As an example, defining matrix elements obtained by flipping spins in a vertex as

$$W\left(\begin{smallmatrix}f_3,f_4\\f_1,f_2\end{smallmatrix}\right)(p)$$
$$=\langle f_3S_i^z(p),f_4S_j^z(p)|H_b|f_1S_i^z(p-1),f_2S_j^z(p-1)\rangle, \tag{19}$$

where $f_i=-1$ if the spin on leg i ($i=1,2,3,4$) is flipped and $f_i=+1$ if it is not flipped, the probability of exiting at leg 2 if the entrance is at leg 1 is given by

$$P_{2,1}=\frac{W\left(\begin{smallmatrix}+&+\\-&-\end{smallmatrix}\right)}{W\left(\begin{smallmatrix}+&+\\+&+\end{smallmatrix}\right)+W\left(\begin{smallmatrix}+&+\\-&-\end{smallmatrix}\right)+W\left(\begin{smallmatrix}-&+\\-&+\end{smallmatrix}\right)+W\left(\begin{smallmatrix}+&-\\-&+\end{smallmatrix}\right)}, \tag{20}$$

where we have used $\pm$ for ± 1. The reasons for this choice for the probability will be discussed in Sec. II C. If the entrance and exit correspond to different sites (the switch-and-

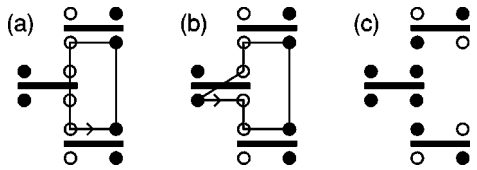

FIG. 4. Two different ways in which an operator loop can close. The starting points of the loops in (a) and (b) are the legs from which the arrows point out. In (a) the last segment of the loop connects the initial and final vertices, resulting in the starting spin being flipped in the final configuration. In (b) the last loop segment is within the initial vertex and the starting spin is flipped twice, with the net effect of no change. Both loops (a) and (b) here result in the updated vertices shown in (c).

reverse and switch-and-continue processes in Fig. 3), the change in the vertex corresponds to a change of the type of the operator (diagonal $\leftrightarrow$ off-diagonal). The leg to which the exit is linked is taken as the entrance to the next vertex, from which an exit is again chosen. This procedure is repeated until the original starting point is reached (the loop closes). The mismatches (links connecting different spin states) existing at the original entrance and at the propagating end of the path are then ''healed'' and a new configuration contributing to the partition function has been created. Note that, depending on the way the loop closes, the spin at the leg from which the loop construction was started may or may not be flipped after the full loop has been completed. Examples illustrating this are given in Fig. 4.

One of the two site-switching paths—the switch-and-reverse in Fig. 3(c) or switch-and-continue in 3(d)—is always forbidden since the corresponding off-diagonal matrix element of the Heisenberg bond operator is zero. The bounce path in Fig. 3(a) is always allowed since the vertex is unaffected (the same spin is flipped twice, resulting in no net change). The continue-straight path of Fig. 3(b) is always allowed if the constant $\epsilon > 0$ so that all the diagonal matrix elements in Eq. (18) are larger than zero. If $\epsilon = 0$, at least one of the diagonal matrix elements vanishes, and the continue-straight process is then forbidden in some cases.

If a spin in the state $|\alpha\rangle$ is not acted upon by any of the operators in S_M, it cannot be flipped by the operator-loop update. Such ''free'' spins can, however, be flipped with probability 1/2 since they do not appear in the weight function. Since the average of n, the number of operators in S_M, grows linearly with β, free spins appear frequently only at relatively high temperatures.

It is convenient to define a Monte Carlo step (MCS) as a sweep of diagonal updates at all positions in S_M where possible, followed by the construction of the linked list in which a number N_l of operator loops are constructed before mapping back to a new S_M and $|\alpha\rangle$ and flipping free spins. Observables can be measured after every, or every few, MCSs (in some cases, it may even be worthwhile to record measurements after every loop).

The remaining question now is how many operator loops one should construct in each MCS. The operator loops are typically of highly varying lengths. Each MCS should involve several loop updates, so that a significant fraction of the vertices are visited. In order not to bias the measurements, it is important that N_l is fixed. One cannot, e.g., keep on constructing loops until the number of vertices visited exceeds a predetermined number. The average size of the operator loops depends strongly on the model parameters. It is therefore useful to record the loop sizes and periodically adjust N_l during the equilibration of the simulation. Typically, we determine N_l such that the average cumulative loop length (the number of vertices visited) during one MCS is approximately $2\langle n\rangle$ or $2M$. In recording the loop length, we do not count bounces (since no change results in the vertex off which the path bounces). Among the counted steps there is still some fraction of backtracking ones, i.e., segments of the operator loop where completed vertex updates are reversed. If a bounce occurs already at the first step the loop closes immediately. With our definition, this is a completed loop of length 0. In order not to bias the measurements, such length-0 loops also have to be counted among the N_l completed loops.

One could also fix N_l based on a criterion involving the average number of leg spins which are actually flipped during an MCS, but recording this number is slightly more complicated than just keeping track of the loop lengths. Since this has to be done only during equilibration, the cost is not prohibitive, however. The exact definition of N_l and precisely what constitutes one MCS are not critical issues (as in the classical Wolff cluster algorithm [45], where the MCS can also be defined in a way analogous to what we have discussed here).

The operator-loop construction (the operator path) is a type of random walk in a $(d+1)$-dimensional space (although the network of connected vertices does not necessarily have this dimensionality—it could effectively have a fractal dimension $< d+1$). One may therefore wonder whether the closing of the loop could become problematic, especially for large systems in three dimensions. In some cases, an operator loop can indeed become very long before it closes. In rare cases a loop may even not close during a simulation of practical length. The loop size distribution is always very broad, however, and the nonclosing problem can simply be circumvented by imposing a maximum length beyond which the loop construction is terminated. The way we typically implement this termination is by immediately initiating a new MCS (beginning with a diagonal update), hence disregarding all the loops that were constructed during the MCS of the terminated loop. This way, we do not have to save actual operator paths (needed in order to undo the changes done during the construction of the terminated loop), which would become impractical for long paths. The termination does not violate detailed balance and hence the correct distribution of configurations contributing to Z is maintained. Termination of incomplete loops does introduce a bias in quantities that are related to the extended configuration space of unclosed paths, however, such as single-particle Green's functions [44]. Typically, we use a maximum loop length $\approx 100\langle n\rangle$. For the XXZ model (in any number of dimension) incomplete loop termination is then extremely rare (excessively long loops can occur more frequently in other models [40] but never seem to be a very serious problem). The av-

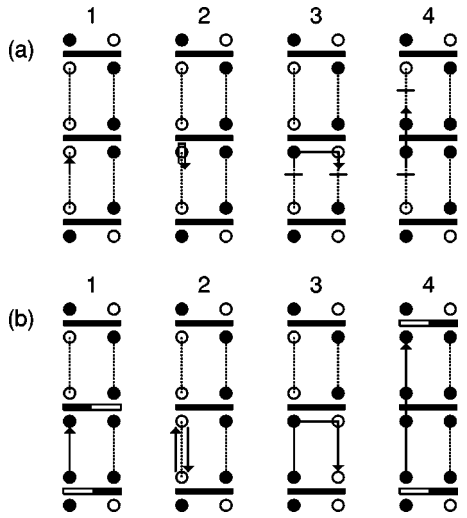

FIG. 5. Two ways to look at the extended configuration space generated during operator-loop construction. Examples of how the configuration shown in (a1) is modified at the beginning of an update in the link-discontinuity picture (a) and ladder operator picture (b) are shown. In (a) the arrow in 1 indicates the proposed starting point of the loop. In (b) a first step of flipping the two spins at this link has already been carried out (generating ladder operators which are indicated by vertices with semifilled bars), and the arrow indicates the entrance point for the following step. In both (a) and (b) configurations that can be generated out of 1 are shown in 2–4. Link discontinuities are indicated by small horizontal lines in (a). In both cases, configuration 2 corresponds to the bounce process, which results in immediate return to the original configuration.

erage loop length is typically much smaller than $\langle n \rangle$, but can in some cases be a significant fraction of $\langle n \rangle$ (up to tens of percent).

C. Detailed balance

In the originally proposed operator-loop scheme [18], the probability of selecting an exit leg is proportional to the corresponding matrix element (18) when the entrance and exit spins have been flipped (with the factor of proportionality chosen to give probability 1 for the sum of the four probabilities), a specific example of which is written in Eq. (20). One can prove that detailed balance is satisfied in this process by considering an extended configuration space which includes also the intermediate configurations generated during the loop construction (which do not contribute to the partition function).

The detailed balance proof is illustrated by an example for a configuration with three vertices in Fig. 5(a). In (a1), the leg with the arrow has been selected as the initial entrance point of the operator loop. An exit leg is chosen according to the probabilities discussed above. Flipping both the entrance spin and the exit spin leads to a new configuration in the extended space. In Fig. 5(a), the three resulting configurations which have nonzero probability are shown in (a2)–(a4). The entrance $\rightarrow$ exit paths are also indicated and the corresponding spins have been flipped. The probability of process (a3) corresponds to the example given in Eq. (20), which when the actual spin states are inserted becomes

$$P_{2,1}=\frac{\langle\uparrow\downarrow|H_b|\downarrow\uparrow\rangle}{\langle\downarrow\uparrow|H_b|\downarrow\uparrow\rangle+\langle\uparrow\downarrow|H_b|\downarrow\uparrow\rangle+\langle\uparrow\uparrow|H_b|\uparrow\uparrow\rangle}. \quad (21)$$

In (a2), the entrance and exit are at the same leg. This is a bounce process which closes the loop immediately with no change in the configuration. In (a3) and (a4) the vertex has changed and two links have appeared which connect legs with different spin states. We call these ''link discontinuities.'' Only configurations with no link discontinuities contribute to Z. All configurations created during the loop construction contain two link discontinuities, until the loop closes (which can be seen as the discontinuities annihilating each other). There are no weight changes associated with the link discontinuities—the configuration weight is still considered to be given by Eq. (17). Hence, the only weight change arises from the change in the affected vertex when the entrance and exit spins are flipped.

The way the exit leg is chosen at the start of the operator loop corresponds to a heat-bath algorithm. The probabilities of no change (staying in the original subspace) or transfer to a configuration with two link discontinuities are proportional to the respective weights in the extended space. Once a configuration with two discontinuities has been created (i.e., the first step was not a bounce), we do not want to create more discontinuities (which would take us out of the extended space considered here) and therefore the following updates can only take place at the discontinuities (the end points of the path), i.e., the discontinuities can be moved. Here the same heat-bath algorithm as in the first step is used. The only difference is that the entrance leg is not chosen at random but is given by a link from the previous vertex. Hence, the whole process consists of a series of heat-bath steps, which satisfy detailed balance and therefore generate configurations according to probabilities proportional to the weight in the extended space. The subset of configurations with zero link discontinuities, which contribute to Z, are therefore also generated with the correct distribution. The process is ergodic because all types of vertices can be generated and the operator path can wind around the periodic boundaries and change both the spatial winding number and the total magnetization. Within a sector of fixed winding number and magnetization, local updates which constitute a small subset of the operator loops suffice to ensure ergodicity [42].

Instead of thinking about the extended configuration space in terms of link discontinuities, one can consider the vertices created when one of the spins in the original vertices of Fig. 1 is flipped. These new vertices correspond to the single-spin flipping (ladder) operators S_i^+ and S_i^-. The loop construction can be formulated in terms of introducing pairs of these, which are then randomly propagated until they reach the same vertex and annihilate each other. The start of such a process is illustrated in Fig. 5(b), using the same configuration and starting point as in Fig. 5(a). The difference with respect to the previous discussion is that now there are no link discontinuities. Instead, the spins at both ends of the link at the selected entrance leg are flipped simultaneously. This introduces two ladder vertices. Here one has to

assign a value v_l to the matrix elements of the ladder operators (i.e., the new operators are $v_l S_i^+$ and $v_l S_i^-$). The initial loop segment, an example of which is shown in Fig. 5(b1), is then generated only with a probability $\min[1, v_l^2/(W_1 W_2)]$, where W_1 and W_2 are the matrix elements corresponding to the two vertices that are considered for replacement by ladder vertices. If this first step is accepted, the next step is again to choose an exit leg. As before, the propagation of the path is carried out according to a heat-bath algorithm, with probabilities proportional to the matrix element when the entrance and exit spins have been flipped. In the example, paths (b2) and (b3) lead to closed loops (back to the space with no ladder operators), whereas in (b4) the ladder operators are moved further away from each other. Note that both spins on the link corresponding to the exit leg are flipped at every step, so that no link discontinuities appear. The process continues until the two ladder operators are on the same vertex, which then becomes equal to one of the original bond-operator vertices. This brings the system back into the original configuration space.

The link-discontinuity and ladder operator pictures of the loop construction are clearly completely equivalent, although the probabilities associated with starting (or closing) the loop are different. In actual simulations it is typically more convenient to use the link-discontinuities view. The ladder operator picture explicitly relates the extended configuration space to that of correlation functions involving these operators, but the link discontinuities can be easily related to them as well. The issue of measuring off-diagonal correlation functions using the SSE operator loops has been considered in Ref. [44].

In Sec. III we will give a more formal and complete proof of detailed balance. We will show that the heat-bath algorithm is not the only, and also not the most efficient, way to satisfy detailed balance when constructing the operator loop. We will introduce the concept of a directed loop to form a general framework for loop updating schemes, both in SSE and path-integral simulations. In the SSE scheme, the directed loop simply leads to different probabilities of choosing among the four exits from a vertex, all other aspects of the method remaining as has been discussed in this section. Before introducing the directed-loop concept, we first consider special cases in which the bounce process can be excluded.

D. Excluding backtracking in special cases

In the general operator-loop algorithm discussed above, the probability of the bounce process is always nonzero, because the vertex remains unchanged and has a nonzero value (otherwise, it would not appear in the configuration in the first place). In some special cases, it is possible to modify the algorithm in such a way that the bounce is completely excluded. This has very favorable effects on the simulation dynamics, since there is then no backtracking and all segments of the loop accomplish changes in the configuration.

The most important of the special cases is the isotropic Heisenberg model ($\Delta=1, h=0$) [18]. A very similar algorithm exists for the ferromagnet ($J<0$) [23]. For the antiferromagnet, choosing the constant $\epsilon=0$ in Eqs. (18) implies that the vertices with all spins up or all spins down vanish and the remaining four matrix elements all equal 1/2. As a result, the matrix element product in Eq. (17) is simply $(1/2)^n$ and is not affected by the operator-loop update. If the bounce process is excluded, the only remaining process is the switch-and-reverse shown in Fig. 6(a) and the path is hence deterministic. The actual loop structure is only changed by the diagonal update. The deterministic loop process is clearly symmetric with respect to flipping or flipping back the spins at all vertex legs covered by the loop, and hence it obeys detailed balance. For the ferromagnet, the bounce can be excluded if $C=-1/4$ in Eq. (6) [for the isotropic ferromagnet $\Delta=-1$ and there is no minus sign in Eq. (5)], and the only remaining process is then the switch-and-continue process shown in Fig. 6(b).

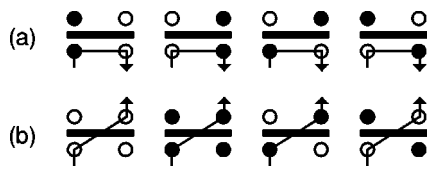

FIG. 6. All allowed vertices in the deterministic operator-loop algorithm in the case of the Heisenberg antiferromagnet (a) and ferromagnet (b). Operator-loop segments starting at the lower left leg are also shown.

In the deterministic case, each vertex leg can be uniquely assigned to a loop, and the loops can be flipped independently of each other. Instead of randomly choosing starting points and constructing a fixed number of loops, one can then construct all possible loops exactly once, by always picking a starting point which does not belong to a loop already constructed. The loops should then be flipped with probability 1/2. The random decision of whether or not to flip can be made before the loop is constructed, but even if the decision is not to flip one has to construct the whole loop and set flags on the vertex legs visited, so that one does not attempt to construct the same loop again. Loops are constructed this way until all $4n$ vertex legs have been visited. This method of constructing all the loops is analogous to the classical Swendsen-Wang multicluster method [46], whereas, as was already mentioned above, the operator-loop construction in the general nondeterministic case is more similar to the Wolff single-cluster algorithm [45].

It should be noted that in the deterministic case an algorithm including only operator updates (diagonal updates and loops) is not completely ergodic. In the antiferromagnet, states with all spins up or down are isolated from the other states since no operators can act on them. These two states are important only at very high temperatures and they can then be reached by also performing random flips of free spins. In simulations with $\epsilon>0$ all states can be reached even without such spin flips.

Another special case is the XY model [18,26] ($\Delta=0, h=0$). In this case all matrix elements in Eq. (18) equal 1/2 if the constant $\epsilon=1/2$. The weight is then again only dependent on n and does not change in the operator-loop update. The bounce can therefore be excluded also in this case, leaving

two remaining allowed exits from each vertex. Although these paths are not deterministic, one can still subdivide the system into loops that can be flipped independently of each other.

The loop structure in the general operator-loop algorithm, which includes bounce processes, is similar to that in the worm algorithm for continuous-time path integrals [3], although the two methods are quite different in other respects (the actual processes used to construct the SSE operator loops and the worms are different—see Sec. VII). In the special cases where the bounce process can be excluded, the SSE operator loops are analogous to the world-line loops (in discrete [16] or continuous [4,19] imaginary time). The close relationships between the Euclidean path integral in continuous time and the discrete representation on which the SSE method is based has been discussed in previous papers [2,43,47] and will also be further elucidated here in Sec. IV.

III. DIRECTED LOOPS

In the operator-loop update discussed in Secs. II B and II C, detailed balance is satisfied using a heat-bath algorithm for propagating the path between connected vertices. In this section we will present a more general set of equations that have to be satisfied for detailed balance to hold in such a process. We will show that these equations have an infinite number of solutions, some of which can lead to a more efficient sampling than the heat bath. We construct a particular solution based on the intuitive hypothesis (for which we have no rigorous proof) that the probability of bounces (back tracking) should be minimized. We show that the bounces can in fact be completely excluded in a much wider range of parameters than at the two isolated points (isotropic XY and Heisenberg) discussed in Sec. II D.

We call the entities involved in the more general scheme directed loops, because the detailed balance equations that we construct (the directed-loop equations) explicitly take into account the fact that the construction of the path of vertices is directional, i.e., the probability of exiting at a particular leg, given the entrance leg, is not the same as the probability of the reverse process. The original operator-loop update with the heat-bath probabilities [18] discussed in the preceding section corresponds to a particular solution of the directed-loop equations. We stress that if another solution is used, the only difference in the actual simulation with respect to the original scheme is a different set of probabilities for exiting at a given vertex leg, given the entrance leg and the four spin states. Before we explicitly construct new solutions in the context of the XXZ model we begin by describing more generally how the directed-loop equations arise.

A. Conditions for detailed balance

Let us first recall that the detailed balance requirement reads

$$P(s\rightarrow s')W(s)=P(s'\rightarrow s)W(s'), \tag{22}$$

where s denotes a configuration having weight $W(s)$, which in the SSE method is expressed as a product over vertex weights, Eq. (17), and $P(s\rightarrow s')$ is the probability of changing the configuration from s to s'. While the weights are given by the Hamiltonian, the probability for how to update the configuration depends on the actual algorithm used.

The algorithm for constructing an operator loop to update an SSE configuration is quite general for any form of the two-body interaction (and can be extended also to multiparticle interactions). With the configuration mapped onto a linked vertex list, an initial entrance vertex leg is first picked at random among all $4n$ legs. Then an exit leg belonging to the same vertex is chosen in a probabilistic way and the spins on the entrance and exit legs are flipped with unit probability. More generally, the states at these legs are updated, with nonzero probabilities only for changes leading to vertices corresponding to nonzero matrix elements. For simplicity, we here assume that the change at the exit leg is uniquely dictated (through conservation laws) by the change at the entrance leg (generalization to cases where the uniqueness does not hold are straightforward). The process continues using the leg linked to the exit leg as the new entrance leg. The process stops when the initial starting leg is reached. The probability for arriving at a new configuration s' can therefore be written as

$$P(s\rightarrow s')=\sum P(e_0)P(s,e_0\rightarrow s_1,e_1) \times P(s_1,e_1\rightarrow s_2,e_2)\times\cdots\times P(s_{n-1},e_{n-1}\rightarrow s',e_0), \tag{23}$$

where $P(e_0)$ is the probability for choosing the vertex leg e_0 as the initial starting point and $P(s_i,e_i\rightarrow s_{i+1},e_{i+1})$ is the probability given a configuration s_i and the entrance leg e_i to exit the vertex at x_i, which is connected to the next entrance leg e_{i+1}, resulting in a new configuration s_{i+1}. The intermediate configurations s_i belong to the extended space of configurations with two link discontinuities, as discussed in Sec. II C. The exit legs x_i do not explicitly appear in the probabilities since they are uniquely linked to the following entrance legs e_{i+1}. The sum is over all possible closed loops which result in the updated configuration being the particular configuration s'. To find a convenient way of choosing the probabilities on the right-hand side of Eq. (23) one needs an expression for the inverse process where the spin configuration s' is transferred into s. This can be written down quite simply by realizing that for each of the terms in Eq. (23) there is a corresponding term that describes the "time-reversed" path, which contributes to the reverse probability. Thus one can write

$$P(s'\rightarrow s)=\sum P(e_0)P(s',e_0\rightarrow s_{n-1},e_{n-1}) \times\cdots\times P(s_2,e_2\rightarrow s_1,e_1)\times P(s_1,e_1\rightarrow s,e_0), \tag{24}$$

where the sum is over the *same* closed loops as in Eq. (23). By inserting these expressions into the detailed balance equation (22) we see that balance is satisfied if

$$W(s_i)P(s_i,e_i\rightarrow s_{i+1},e_{i+1})=W(s_{i+1})P(s_{i+1},e_{i+1}\rightarrow s_i,e_i) \tag{25}$$

for all possible SSE configurations and entrance legs. Because the update $(s_i,e_i\rightarrow s_{i+1},e_{i+1})$ changes only one particular vertex, all except one of the factors in the product of vertex weights in Eq. (17) factor out and cancel. Writing $W_s P(s,e\rightarrow s',x)=W(s,e,x)$, where we have slightly changed the notation so that W_s denotes the matrix element corresponding to a *single* vertex with its four leg states coded as s, e is the entrance leg, and x is the exit leg on the same vertex, one can formulate the detailed balance criterion Eq. (25) as

$$W(s,l_1,l_2)=W(s',l_2,l_1), \tag{26}$$

which should be valid for all possible vertex types which can be converted into each other by changing the states at the entrance and exit legs. This equation implies many relations between the unknown probabilities of how to choose an exit leg given a particular vertex and an entrance leg. There are additional relations which must be satisfied. Requiring that the path always continues through a vertex translates into

$$\sum_x P(s,e\rightarrow s_x,x)=1, \tag{27}$$

where the sum is over all legs on the vertex. We have emphasized in the notation s_x that the resulting spin configuration depends on the exit leg. In terms of the weights $W(s,l_1,l_2)$ this requirement translates into

$$\sum_x W(s,e,x)=W_s, \tag{28}$$

which must be valid for all vertices and entrance legs. These equations, Eq. (28) together with the relations in Eq. (26), form the directed-loop equations, the foundations of our approach to construct valid probability tables for the operator-loop update.

B. SSE directed loops for the *XXZ* model

For the XXZ model there are just three possible exits for any given entrance leg as one choice always leads to a zero-weight state when spins connected by the loop segment are flipped (due to violation of the z-magnetization conservation of the model). Figure 3 illustrates the possibilities for placing directed-loop segments for different vertices. In order for our updating process to satisfy detailed balance we recall that according to Eq. (26) we must relate vertices in which the two spins connected by the loop segment are flipped *and* the direction of the loop segment is reversed. Such related configurations are illustrated in Fig. 7. Furthermore, Eq. (28) relates vertices with different exit legs having the same spin configuration and entrance legs. We then make the key observation that all possible vertex configurations can be divided into eight subsets that do not transform into each other. Half of these subsets are shown in Fig. 8, where only configurations within the *same* quadrant are transformed into each other. These configurations form closed sets under the flipping operation. It is therefore sufficient to derive the detailed balance conditions, Eq. (26), for transitions between vertex configurations in the same set independently of other configurations.

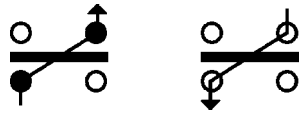

FIG. 7. Example of two vertices with directed-loop segments that transform into each other in the flipping process.

A row in any of the quadrants in Fig. 8 contains all three configurations which can be reached by entering a certain vertex from a certain entrance leg. For instance, in the upper left quadrant the entrance leg for the first row is the lower left one, for the second row, the lower right one; and for the third row, the upper right one. According to Eq. (28), the sum of the weights of all possible configurations that can be reached from a certain in-leg, keeping the spin configuration fixed, should equal the vertex weight alone. Thus taking the upper left quadrant of Fig. 8, we have for rows 1–3 from the top,

$$\begin{aligned} W_1&=b_1+a+b,\\ W_2&=a+b_2+c,\\ W_3&=b+c+b_3, \end{aligned} \tag{29}$$

where the symbols on the left-hand sides are the vertex weights, Eqs. (18), in the spin configuration space, i.e.,

$$W_1=\langle\uparrow\downarrow|H_b|\downarrow\uparrow\rangle=\langle\downarrow\uparrow|H_b|\uparrow\downarrow\rangle=1/2,$$

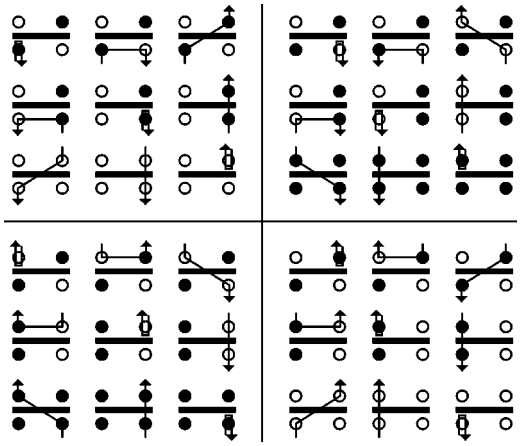

FIG. 8. Possible assignments of directed-loop segments for half of the different combinations of vertices and entrance legs. The other half of the vertex configurations can be obtained by interchanging up and down spins (solid and open circles) while keeping the arrows. The lines with arrows are the directed-loop segments. The configurations are divided into four sets (one in each quadrant). On flipping the spins connected by the loop segment and reversing the direction of the arrow, only configurations within the *same* quadrant are transformed into each other.

$$W_2=\langle\downarrow\uparrow|H_b|\downarrow\uparrow\rangle=\langle\uparrow\downarrow|H_b|\uparrow\downarrow\rangle=\Delta/2+h_b+\epsilon,$$

$$W_3=\langle\downarrow\downarrow|H_b|\downarrow\downarrow\rangle=\epsilon, \tag{30}$$

$$W_4=\langle\uparrow\uparrow|H_b|\uparrow\uparrow\rangle=\epsilon+2h_b,$$

while those on the right are weights in the enlarged configuration space of spins and directed-loop segments. We have assigned equal weights to the configurations that are related by flipping, in accordance with Eq. (26). The order of the symbols on the right-hand sides of Eqs. (29) follows the order in the upper left quadrant of Fig. 8, so that, e.g., the weight of the two configurations in Fig. 7 is b and the weight of the middlemost configuration in the upper left quadrant of Fig. 8 is b_2. We use b with a subscript to denote a weight of a configuration where the exit equals the entrance (bounce process).

As mentioned above there are in all eight sets of vertex configurations which close under the flipping process. These sets are, in principle, independent of each other and have their own equation sets. However, one can easily convince oneself that because of symmetry reasons there are only two different types of sets. One of these symmetries is that of permuting the two spins acted upon by H_b. This implies that the equations derived for the set in the upper left quadrant in Fig. 8, Eqs. (29), are the same as for the set (not shown) that can be obtained from the upper right quadrant by interchanging up- and down-spins, keeping the orientation of the directed-loop segments. The other symmetry is that of imaginary time inversion, which in the figures corresponds to switching the pairs of spins below and above the horizontal bar representing the operator H_b. This symmetry together with the previous one identifies the rules for the upper left quadrant of Fig. 8 with those of the lower right quadrant. Thus, one only has to consider two independent sets of equations, Eqs. (29) and the corresponding equations that can be derived from the lower left quadrant in Fig. 8:

$$W_1=b_1'+a'+b',$$

$$W_2=a'+b_2'+c', \tag{31}$$

$$W_4=b'+c'+b_3'.$$

This latter set takes the form of the set (29) but with primed symbols to denote the weights and W_4 instead of W_3. There is a further reduction in the case of zero magnetic field, where the two equation sets become identical.

Before discussing solutions to these sets of equations it should be stressed that the actual probabilities for selecting the exit leg are given by dividing the weight in the extended configuration space by the weight of the bare vertex, so that, e.g., the probability for choosing the ''bounce'' process, given that the entrance leg is the lower left one on a vertex with weight W_1, as shown in the uppermost left-hand corner of Fig. 8, is b_1/W_1.

It is clear that there are many solutions with only positive weights to the above equation sets as they are underdetermined; both sets have six unknowns and there are three equations with the additional requirement of non-negative weights. A particular symmetric solution is that corresponding to the heat-bath probabilities used in the original scheme [18], which we will henceforth refer to as solution *A*. It is given by

$$a=W_1W_2/(W_1+W_2+W_3),$$

$$b=W_1W_3/(W_1+W_2+W_3),$$

$$c=W_2W_3/(W_1+W_2+W_3), \tag{32}$$

$$b_i=W_i^2/(W_1+W_2+W_3).$$

For the primed weights, W_3 is replaced by W_4. Clearly the probabilities for choosing the exit leg are here proportional to the weights of the resulting bare vertices, which are obtained by flipping the two spins on the loop segment, an example of which was given in Eq. (20). This solution is valid in the full parameter space of the XXZ model. However, it generally assigns a relatively large weight to the bounce processes where the exit leg equals the entrance leg. These are ineffective in updating the configurations. In particular, when the field $h\to 0$ and the anisotropy $\Delta\to 1$, the bounce probability approaches 1/2. Although the method still is reasonably efficient (we are not aware of any method that has been more successful for models including external fields), this is bothersome since the SSE algorithm exactly at $h=0$ can be formulated entirely without any bounce processes [18,43], as reviewed in Sec. II D, and is then considerably more efficient. The fact that the $h=0$ scheme has no bounces and is completely deterministic, whereas the $h\to 0$ method has bounce probabilities approaching 1/2, inspires us to look for solutions where the bounce probability instead vanishes continuously as $h\to 0$. This will eliminate the algorithmic discontinuity of the previous approach.

For the discussion of other solutions to the directed-loop equations (29) and (31) it is convenient to express these equations in terms of the bounce weights $b_1,\ldots,b_3'$:

$$a=\frac{1+\Delta}{4}+\frac{h_b}{2}+\frac{-b_1-b_2+b_3}{2},$$

$$b=\frac{1-\Delta}{4}-\frac{h_b}{2}+\frac{-b_1+b_2-b_3}{2},$$

$$c=\frac{\Delta-1}{4}+\frac{h_b}{2}+\epsilon+\frac{b_1-b_2-b_3}{2}, \tag{33}$$

$$a'=\frac{1+\Delta}{4}-\frac{h_b}{2}+\frac{-b_1'-b_2'+b_3'}{2},$$

$$b'=\frac{1-\Delta}{4}+\frac{h_b}{2}+\frac{-b_1'+b_2'-b_3'}{2},$$

$$c'=\frac{\Delta-1}{4}+\frac{3h_b}{2}+\epsilon+\frac{b_1'-b_2'-b_3'}{2},$$

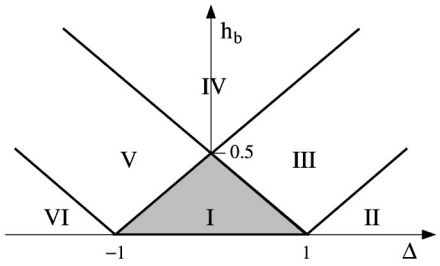

FIG. 9. ''Algorithmic phase diagram'' showing regions where various bounce weights must be nonzero. The actual values of these weights are given in Table I. In the shaded region all bounce weights can be set to zero. Other bounce weights are listed in Table I.

where we have explicitly inserted the expressions for the vertex weights, Eq. (30). We seek positive solutions to these equations. Being underdetermined the set has many solutions, so we will try to find the solutions that yield the most effective algorithms. As a general principle for finding efficient rules, we will attempt to minimize the bounce weights $b_1, \ldots, b_3'$. The solution so obtained will be termed solution *B*. Inspecting the equations, it is clear that there is one region in parameter space where one can avoid bounces altogether. This region is shown as the shaded region in Fig. 9. From the requirement of non-negative vertex weights we already have the restriction $\epsilon \geqslant 0$. In the shaded region, the requirement of non-negative weights also in the enlarged configuration space when all the bounce weights are zero imposes an additional constraint on ϵ: $\epsilon \geqslant (1-\Delta)/4 - h_b/2$. We have no rigorous principle of finding the optimal value of ϵ, in general, but as can be inferred from our simulation tests (presented in Sec. V) it is often advantageous to choose a small but nonzero value in cases where $\epsilon_{\min}=0$.

For the Heisenberg antiferromagnet at zero magnetic field $(\Delta=1, h=0)$ the deterministic algorithm constructed in Ref. [18] is recovered for the choice $\epsilon=0$. The nonzero weights are then $a=a'=1/2$, while the nonzero matrix elements are $W_1=W_2=1/2$, which correspond to the switch-and-reverse process illustrated in Fig. 6(a). This is a deterministic algorithm as the only probabilities different from zero are unity. There is a subtlety here as the ratio c/W_3 is undetermined for $\epsilon=0$. However, the value of this probability can be left undetermined as the vertex with all spins down will not be generated as a consequence of the vanishing of the weight W_3. This is actually more general—whenever a probability cannot be defined because of a zero denominator, it can be left undetermined because the probability of reaching such a vertex is zero in the first place. For the *XY* model $(\Delta=0)$ at zero magnetic field the choice $\epsilon=1/4$ gives a different set of zero-bounce rules than that proposed in Ref. [18], which, however, also is a solution of our equations (but with $\epsilon=1/2$). It is quite remarkable that for the *XY* model one can in fact find rules with no bounces for all magnetic field strengths up to the saturation field. We expect this to be very useful.

Outside the shaded region in Fig. 9 one or more bounce weights must be nonzero. In these regions we will again choose the smallest possible values for the bounce weights.

TABLE I. Nonzero bounce weights and minimum values of ϵ for the different parameter regions of solution *B* of the directed-loop equations. The Roman numerals correspond to those in Fig. 9. We have defined $\Delta^{\pm}=(1\pm\Delta)/2$.

	Bounce weights		$\epsilon_{\min}$
I			$(\Delta^- - h_b)/2$
II	$b_2 = h_b - \Delta^-$	$b_2' = -h_b - \Delta^-$	0
III	$b_2 = h_b - \Delta^-$		0
IV	$b_2 = h_b - \Delta^-$	$b_3' = h_b - \Delta^+$	0
V		$b_3' = h_b - \Delta^+$	$(\Delta^- - h_b)/2$
VI	$b_3 = -h_b - \Delta^+$	$b_3' = h_b - \Delta^+$	$-h_b - \Delta/2$

Table I shows these values for the different regions in Fig. 9, along with the minimum value of ϵ allowed. Selecting a value for ϵ, the remaining weights can be obtained using Eqs. (33).

At the boundary between regions in Fig. 9, one of the bounce weights vanishes continuously. In particular, this means that the rules for the Heisenberg antiferromagnet in a magnetic field approaches the rules in zero field *continuously* as $h_b \to 0$. This is to be contrasted to the symmetric solution *A*, Eqs. (32), where the bounce probabilities approach 1/2 as $h_b \to 0$. Hence, the algorithmic discontinuity is indeed removed as the special deterministic solution at the isotropic point is recovered automatically with solution *B* (when $\epsilon=0$).

In Sec. V, the performance of simulations using solutions *A* and *B* will be quantified in terms of calculated autocorrelation functions. It will be shown that the new solution *B* can lead to autocorrelation times more than an order of magnitude shorter than with solution *A*. The improvements are most dramatic for weak but nonzero fields and weak Ising anisotropies $(\Delta>1)$. In Sec. IV we will describe how the directed loops also can be adapted to simulations in the path-integral formalism. Below we first briefly discuss the form of the detailed balance equations for more general Hamiltonians.

C. General form of the directed-loop equations

The SSE operator-loop update with the heat-bath probabilities [18] has already been applied to several different systems, including spin systems with $S>1/2$ [35], various boson models [36–38], as well as the 1D extended Hubbard model [40]. The directed-loop approach can also be easily applied to a much wider class of models than the $S=1/2$ *XXZ* model discussed in the preceding section, and minimum-bounce solutions can be expected to lead to significant improvements in efficiency. We here briefly outline the general form of the directed-loop equations and their solutions for a general two-body interaction.

When the operator-loop update is applied to models with higher spins, boson or fermion models, it is clear that the simple notion of flipping a spin in the $S=1/2$ *XXZ* model must be extended to a change in the state at a vertex leg where the final state is one out of several possible ones.

Consider as an example a spin-1 model where a loop can change the state on a leg by one or two units of spin. This is simplified when the total S^z is conserved as then these different changes can be considered as two *independent* loop updates. This is because changing the state on the exit leg by two units of spin when the state on the entrance leg is changed by one unit, or vice versa, violates the S^z conservation law. Thus, with such a conservation law the state change of the exit leg is uniquely determined given the state change at the entrance leg. For simplicity we will here consider only those cases where this uniqueness holds, although this is by no means a necessary condition.

In order to describe the general form of the directed-loop equations for this type of general two-site interaction it is convenient to change the labeling somewhat from that used in the preceding section. To define this new labeling, we start by selecting a reference vertex (which can be any of the allowed vertices) and label its weight W_1. We then choose an entrance leg and label this leg as leg 1, and then number the rest of the legs on this vertex 2,3, and 4. Distributing the weight over all possible exit legs according to Eq. (28) gives

$$W_1 = a_{11} + a_{12} + a_{13} + a_{14}, \tag{34}$$

where we have labeled the weights a_{ij} in the extended space by their entrance (i) and exit (j) legs. On changing the states at both the entrance and exit legs one arrives at a new vertex. If the entrance and exit legs are the same the vertex stays the same. Now label the weight of the vertex reached by exiting at leg i as W_i. Thus if the exit was on leg 2 we would label that vertex W_2. W_2 has a similar decomposition as W_1,

$$W_2 = a_{21} + a_{22} + a_{23} + a_{24}, \tag{35}$$

where now the entrance is on leg 2 on the vertex which differs from vertex 1 by having changed the states at leg 1 and 2. The weight a_{21} corresponds to the process where the path enters at leg 2 and exits at leg 1. The states are changed in the *opposite* way to that when arriving at W_2 from W_1, and hence the process is undoing the changes and we arrive back at W_1. From Eq. (26) it follows that $a_{21} = a_{12}$. Now one can asks the question of whether exiting at leg 3 or 4 yields the same vertex type when starting from W_2 as it does starting from W_1. The answer to this is yes, because starting from W_1 one would change the state at leg 1 and 3 while starting from W_2 one would change the states at legs 2 and 3. But W_2 differs from W_1 only by having different states at legs 1 and 2 and thus the state at leg 2 is *changed twice* in opposite directions resulting in the same configuration W_3. The weights are hence uniquely defined by this procedure, and one is guaranteed that the only vertices that are related by the detailed balance equations are those that can be reached by changing the state on the entrance leg together with the state on any exit leg of the reference vertex. The directed-loop equations can therefore be written as

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{12} & a_{22} & a_{23} & a_{24} \\ a_{13} & a_{23} & a_{33} & a_{34} \\ a_{14} & a_{24} & a_{34} & a_{44} \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} W_1 \\ W_2 \\ W_3 \\ W_4 \end{pmatrix}, \tag{36}$$

where the matrix on the left-hand side is a real symmetric 4×4 matrix with all entries non-negative for a useful algorithm. The magnitudes of the diagonal elements determine the bounce probabilities. This is the general structure of the directed-loop equations for two-site interactions. There are, in general, several such sets of equations, which can be generated one by one by changing the reference vertex and the type of change at the entrance leg. The reference vertex should then of course be chosen among vertices that have not yet been generated starting from another reference vertex, in order not to generate the same equation sets several times. Some of the different sets are typically identical to each other by symmetry, as in the case of the $S=1/2$ XXZ model, where there are eight sets falling into two classes. In that case the structure of the equations changes into 3×3 forms because there are only three allowed exit possibilities for each entrance leg. To explain this with an example in the scheme used here, we can consider the vertex with all spins down as the reference vertex. Then $W_2=0$ as this configuration corresponds to the case where the lower legs (1 and 2) are flipped, resulting in a vertex with weight zero. This immediately implies that all a's (being all non-negative) with an index 2 must be zero and so the result is that row 2 and column 2 is taken out resulting in a 3×3 matrix. In general, there can be a large numbers of 4×4 equation sets, some or all of which reduce into 3×3 and 2×2 sets (e.g., for Hubbard-type electron models there are both 2×2 and 3×3 sets, but no 4×4 sets).

Let us consider the 3×3 case in greater detail and ask when one can do without bounces, as we saw was possible in a region in parameter space of the $S=1/2$ XXZ model discussed in the preceding section. To do this, it is convenient to first relabel the equations so that $W_3 \geqslant W_2 \geqslant W_1$. We then set all the diagonal entries (the bounce weights) to zero and find the region of different W's for which the equation set has strictly positive solutions. In this case the solution is unique as there are three equations and three unknowns and it is easy to see that the a's are positive only when $W_3 \leqslant W_1 + W_2$, and hence one finds a directed-loop solution without bounces only when this condition is satisfied.

Allowing bounces, it is also easy to see that one can always do with only one bounce, the one that bounces off the vertex with the largest weight. If W_3 is the largest weight, one can set $a_{11}=a_{22}=0$ and $a_{33}=W_3-W_1-W_2$, which gives $a_{12}=0$, $a_{13}=W_1$, and $a_{23}=W_2$. This means that the probability for moving between the configurations with the smallest weight is zero while that of moving from the largest weight configuration to the smaller ones is the ratio of the smaller weight to the larger weight and unity for the reverse process. The bounce probability is unity minus the probability for moving to the smaller weight configurations. A similar analysis can be carried out for the 4×4 equation sets appearing for $S>1/2$ spin models and soft-core boson models.

The equation sets involving larger matrices, as encountered when dealing with interactions involving more than two sites, can also be studied in a similar manner. It should be pointed out, however, that there is nothing that guarantees *a priori* that the operator-loop update is ergodic (in combination with the diagonal updates), for any solution of the directed-loop equations. Ergodicity requires that all allowed vertices can be generated through a series of loop updates, and this is typically the case with two-particle terms (although one could, in principle, construct models where it is not the case). However, simple one-dimensional loops such as those discussed here cannot always accomplish this alone when the interaction includes more than two particles, even in the case of relatively simple models. The SSE method has recently been applied to an XY model with a standard two-spin interaction J and a four-spin term K [48]. In that particular case, an ergodic operator-loop update could be used for $|J|>0$, but for $J=0$ another cluster-type update had to be carried out. In practice, a combination of the two updates had to be used for large K/J.

IV. PATH-INTEGRAL FORMULATION

In this section we will discuss how the directed loops can be applied to the path-integral Monte Carlo method (PIM) formulated in imaginary time. Such methods are known as world-line methods in discrete [9] or continuous [3,4] imaginary time. The close relationships between the SSE and PIM representations of quantum statistical mechanics have been explored in previous works [47,43]. Here we will show that also the directed-loop ideas can be almost directly translated from SSE into the PIM formalism.

A. Construction of the path integral

We start by writing the partition function as

$$Z=\mathrm{Tr}\{e^{-\beta H}\}=\mathrm{Tr}\left\{\prod_{t=1}^{L} e^{-\Delta\tau H}\right\}, \tag{37}$$

where $\Delta\tau=\beta/L$ and L is a large integer. The Hamiltonian is generally a sum of noncommuting pieces, and in order to deal with the exponential it is convenient to employ the Suzuki-Trotter trick [6]. This involves dividing the Hamiltonian into several sets of terms, where all terms within a set are commuting while the sets themselves are noncommuting. Because the Hamiltonian is multiplied by the small quantity $\Delta\tau$ it is possible to split the exponential into a product of exponentials, each having one set in the exponent. The errors arising from this approximation vanish as $\Delta\tau\to 0$ [6,20]. Consider as an example the XXZ chain. Then the Hamiltonian can be divided into two sets, one involving the operators that act on sites $2n$ and $2n+1$, while the other set involves the operators acting on sites $2n+1$ and $2n+2$. It is then possible to insert complete sets of states, which can be chosen to be written in terms of S^z components, between all the exponentials and the partition function can be written [7–10] as

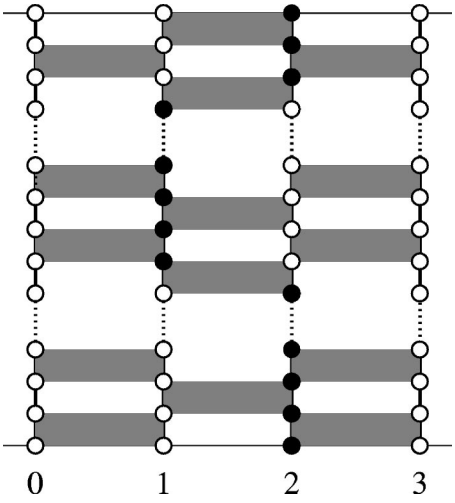

FIG. 10. The checkerboard breakup of the space-time for a spin chain with four sites with open boundary conditions. H_1 has terms acting on the links between site 0 and 1 and the link between site 2 and 3. H_2 acts on the link between site 1 and 2. The shaded plaquettes show where the Hamiltonian acts.

$$Z=\sum_{\{\sigma\}}\prod_{t=1}^{L}\langle\sigma_{t+1}|e^{-\Delta\tau H_2}|\sigma_{t+1/2}\rangle\langle\sigma_{t+1/2}|e^{-\Delta\tau H_1}|\sigma_t\rangle, \tag{38}$$

where σ is a shorthand for a spin configuration in the S^z basis of all sites in the chain. The sum is over all possible sets of spin configurations, two complete sets of states for each time step t, and the trace implies $\sigma_{L+1}=\sigma_1$. This is called the checkerboard breakup, as one can visualize it as a checkerboard pattern (see Fig. 10) where all the matrix elements are pictured as shaded plaquettes. This breakup is completely general and can also be used for higher-dimensional lattices. Because each set H_1 and H_2 consists of individually commuting terms it suffices to consider the interaction on one shaded plaquette only and the matrix elements can easily be written down. Keeping only terms to first order in $\Delta\tau$, one finds

$$\begin{aligned}
W_1&=\langle\uparrow\downarrow|e^{-\Delta\tau H}|\downarrow\uparrow\rangle=\langle\downarrow\uparrow|e^{-\Delta\tau H}|\uparrow\downarrow\rangle=\Delta\tau/2,\\
W_2&=\langle\uparrow\downarrow|e^{-\Delta\tau H}|\uparrow\downarrow\rangle=\langle\downarrow\uparrow|e^{-\Delta\tau H}|\downarrow\uparrow\rangle=1+(C+\Delta/4)\Delta\tau,\\
W_3&=\langle\downarrow\downarrow|e^{-\Delta\tau H}|\downarrow\downarrow\rangle=1+(C-\Delta/4-h_b)\Delta\tau,\\
W_4&=\langle\uparrow\uparrow|e^{-\Delta\tau H}|\uparrow\uparrow\rangle=1+(C-\Delta/4+h_b)\Delta\tau.
\end{aligned} \tag{39}$$

These matrix elements differ from the matrix elements (18) in the SSE method only in that the Hamiltonian is multiplied by the factor $\Delta\tau$ and the diagonal matrix elements also come with the zeroth-order term of the exponential. The weight W_1 comes with a minus sign, which here is omitted by implicitly performing a π rotation about the S^z axis for spins on one sublattice. This can be done whenever the lattice is bipartite. One can of course also calculate the matrix elements (39)

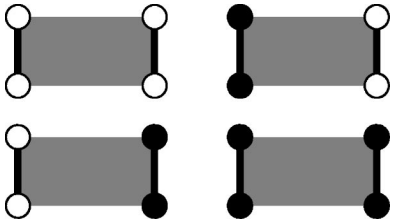

FIG. 11. Loop and spin configurations which should have the same weight when allowing the loops to be flipped independently.

exactly, but since we will take the continuum limit here, it is sufficient to go to linear order in $\Delta\tau$, where the similarity to the SSE expressions are most evident.

In the ordinary world-line loop algorithm (for a review see Ref. [49]), two loop segments are assigned to each and every shaded plaquette in a stochastic way. The shaded plaquettes are corner sharing so that when all shaded plaquettes have been assigned segments one can identify closed loops. Given that the probabilistic rules for the assignment of loop segments for each shaded plaquette follows the analogy of Eqs. (26) and (28), one can flip a loop with any probability. In particular, one can pick a random site and a random imaginary time and flip the loop that includes this point with unit probability. One can also turn this around and first, before any loop is constructed, pick a random point in space-time and then construct the loop starting at this point and flipping spins with unit probability as the loop is being constructed.

When assigning loop segments to each shaded plaquette one needs two loop segments for each plaquette in order to fill the lattice completely. Then many configurations can be reached, as one should be able to independently flip spins along one or both the loop segments. Thus one gets relatively many constraints of the type (26). This is illustrated in Fig. 11. In fact, in zero field there are just as many equations as unknowns, and this set has only non-negative solutions in the XY-like case, $-1\leqslant\Delta\leqslant 1$. In a magnetic field there is one additional equation and the set does not have any solutions. Within the standard loop algorithm this is repaired by introducing additional processes which ''freeze'' loops together, i.e., if spins on one loop are flipped, spins on any loop frozen together with the first one will also be flipped. This increases the number of unknowns in the equation set, making a solution possible. While we are not aware of any systematic studies of the effects of the freezing process, it tends to freeze all loops together, resulting in the trivial spin update where all spins are flipped. It is therefore not very effective. However, in the extreme Ising limit the freezing is responsible for the fact that the loop algorithm becomes equivalent to the Swendsen-Wang algorithm, and hence the freezing of loops has some merits.

Another method to make the loop algorithm work in a magnetic field is to apply the field in the x direction, thereby changing the matrix elements and introducing a minus sign. Using the concept of merons, the resulting sign problem can be solved [21,22], but the simulation algorithm is not very efficient for large systems. If one relaxes the condition that the loops should be flipped with unit probability and instead chooses weights such that the flipping probability is maximized, it is possible to find rules that work very well at extreme fields [19]. However, this success at extreme fields must be regarded as a lucky circumstance and is not generally valid for lower fields. Yet another and perhaps the simplest loop method in the presence of a magnetic field is to construct the loops as if the field was absent and then include a Metropolis decision whenever attempting to flip a loop that changes the magnetization. This method is, however, very ineffective [44] [except at extremely weak fields; $h/J \lesssim 1/(\beta N)$ [25]] as is to be expected, as it does not take into account the actual physics of the model which is the competition between the magnetic field and the exchange energy.

None of the above methods for treating external fields has proven as useful in practice as the SSE operator-loop algorithm [18]. The worm algorithm for path-integral simulations in continuous imaginary time [3] shares some important features with the SSE operator loops (specifically, there is an analog to the backtracking feature) and has also been used successfully. However, its autocorrelation times seem to be much longer (as can be seen in comparing our results in Sec. V with those presented in Ref. [41]). We will discuss differences between the procedures used to construct directed loops and worms in Sec. VII. Because the directed loops are a further improvement of the SSE approach, it is natural to investigate if these concepts can also be implemented in the path-integral formulation.

B. Directed loops in the PIM

To implement the notion of directed loops in the path-integral formulation we note the similarities of the vertices in the SSE and the shaded plaquettes in the PIM. We can identify a corner of a shaded plaquette with a vertex leg in the SSE. Both have a spin attached, and each corner (leg) is connected to another corner (leg) on another shaded plaquette (vertex). To construct a directed loop, we first choose a random entrance corner at a random shaded plaquette. Then, depending on the spin configuration, we choose an exit corner and place a directed-loop segment between the entrance corner and the exit corner. The spins connected by the loop segment are flipped with unit probability. The spin on the exit corner is then the entrance spin of the next shaded plaquette and the process continues until the loop closes. In contrast to the usual loop algorithm there is no notion of freezing loops, but there is the necessary (at least in some regimes) process of bouncing, where the ''loop head'' backtracks some distance along its path and reverses spin flips.

Because of the relation between the SSE vertices and the shaded plaquettes, and the similarity of the matrix elements (30) and (39), one can immediately write down the detailed balance equations for the PIM using Fig. 8 and interpreting the vertices as shaded plaquettes. As in the SSE, there are eight sets of directed-loop equations which are reduced to two by symmetries. Substituting the plaquette weights and expressing the extended configuration weights in terms of the bounce weights, we get

$$
\begin{aligned}
a &= \left(\frac{1+\Delta}{4}+\frac{h_b}{2}\right)\Delta\tau+\frac{-b_1-b_2+b_3}{2},\\
b &= \left(\frac{1-\Delta}{4}-\frac{h_b}{2}\right)\Delta\tau+\frac{-b_1+b_2-b_3}{2},\\
c &= 1+\left(C-\frac{1}{4}-\frac{h_b}{2}\right)\Delta\tau+\frac{b_1-b_2-b_3}{2},\\
a' &= \left(\frac{1+\Delta}{4}-\frac{h_b}{2}\right)\Delta\tau+\frac{-b_1'-b_2'+b_3'}{2},\\
b' &= \left(\frac{1-\Delta}{4}+\frac{h_b}{2}\right)\Delta\tau+\frac{-b_1'+b_2'-b_3'}{2},\\
c' &= 1+\left(C-\frac{1}{4}+\frac{h_b}{2}\right)\Delta\tau+\frac{b_1'-b_2'-b_3'}{2}.
\end{aligned}
\tag{40}
$$

Non-negative weights are required to avoid sign problems. This implies that there are regions where bounces must be nonzero. In fact the same algorithmic phase diagram as shown in Fig. 9 applies here, with the exception that in this case there are no restrictions on C (or $\epsilon = C-\Delta/4-h_b$) as it always occurs multiplied by $\Delta\tau$ in a combination where there also is the zeroth-order term of the exponential. In fact, in the construction of the loops in continuous imaginary time, where only quantities to order $\Delta\tau$ matter, the value of C drops out completely as we will consider *ratios* where it turns out that C does not occur to order $\Delta\tau$. Thus in contrast to the SSE, there is nothing gained by adjusting C in the path-integral representation. Whenever in a region of parameter space where bounces are needed, one can choose them to be the minimum values as summarized in Table I, with the only modification that the bounce weights should be multiplied by $\Delta\tau$. As in the SSE method the actual probability for choosing an exit corner, given an entrance corner and a spin configuration on a shaded plaquette, is obtained by dividing one of the weights above by the appropriate matrix element from Eqs. (39).

In the limit $\Delta\tau\to 0$ this method might seem very slow as one needs to make a choice for every plaquette of which there are infinitely many in this limit. However, one can use the method employed in the continuous-time implementation of the standard world-line loop algorithm [4], which is based on the fact that the c,c' weights are of order unity. The c,c' weights describe the process of continuing the loop construction in the imaginary time direction on the same site. Being of order unity means that this will be the dominating process. The other processes are multiplied by $\Delta\tau$ and will therefore occur much less frequently.

To illustrate in detail how a loop is constructed in the limit $\Delta\tau\to 0$, consider as an example the situation shown in Fig. 12. This figure shows the full imaginary-time spin configurations for four sites. The dotted (solid) lines correspond to spin down (up). The figure can be understood as the limit $\Delta\tau\to 0$ of Fig. 10. The loop construction consists of moving the loop head. This motion begins at a random site and time

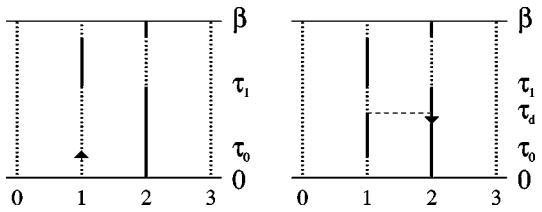

FIG. 12. Left: Continuous imaginary time construction of a loop. This figure can be understood as the limit $\Delta\tau\to 0$ of Fig. 10, dotted (solid) lines correspond to spin down (up). Starting at an arbitrary site and time (indicated by the arrow) a probability of ''decay'' dependent on the spin states of the neighbors is calculated, and the loop head is moved to the point of decay. Right: A resulting a' decay at a time τ_d where the segment up to the decay has changed orientation and a new arrow is placed.

in a random direction. In Fig. 12 the starting point and direction is marked by an arrow. From the arrow at time τ_0 to the time τ_1, the spin configuration on site 1 and its neighbors 0 and 2 stay unchanged. At time τ_1 there is a spin-flip process exchanging the spins on sites 1 and 2. This means that half of the $2\tau/\Delta\tau$, $\tau=\tau_1-\tau_0$, shaded plaquettes (the factor 2 is from the fact that there are two neighbors) between the starting point τ_0 and τ_1 are of the type W_2, while the other half is of the type W_3. The loop head will therefore enter alternately the lower left corner on shaded plaquettes having weight W_2 and the lower right corner on shaded W_3 plaquettes. On exiting the shaded W_2 plaquette, one of the three processes a', b_2', or c' can happen, while for each of the W_3 plaquettes, one of the processes b, c, or b_3 can happen. The c and c' processes are by far the most probable as they are of order unity while the others are of order $\Delta\tau$. Therefore until one of the other processes of order $\Delta\tau$ occurs, the loop head will just continue its motion in the upward direction on site 1. The probability for the first occurrence of one of the processes of order $\Delta\tau$ within an interval $\Delta\tau$ after time τ' is given by

$$
\begin{aligned}
P(\tau')\Delta\tau &= \left(\frac{c'}{W_2}\frac{c}{W_3}\right)^{\tau'/\Delta\tau}\left(1-\frac{c'}{W_2}+1-\frac{c}{W_3}\right)\\
&= e^{-(\alpha_0+\alpha_2)\tau'}(\alpha_0+\alpha_2)\Delta\tau,
\end{aligned}
\tag{41}
$$

where in the last equality we have taken the limit $\Delta\tau\to 0$, and the quantities α_i are finite as $\Delta\tau\to 0$;

$$
\alpha_0 = \frac{b+b_3'}{W_3\Delta\tau}, \tag{42}
$$

$$
\alpha_2 = \frac{a'+b_2'}{W_2\Delta\tau}, \tag{43}
$$

where the subscript on α indicates which neighbor is considered. Recall that by definition $W_3=b+c+b_3'$ and $W_2=a+b_2+c$. Thus, with a random number generator one can generate ''decay'' times according to the distribution (41) and take the random decay time generated as the point where one of the processes a', b_2', b, or b_3 occurs. If the decay

time so generated is bigger than $\tau_1 - \tau_0$ the loop head can be moved directly all the way to time τ_1, while flipping all the spins on site 1 up to time τ_1. There it enters a shaded plaquette from the lower left corner. This plaquette has weight W_1, and the possible choices for exit corners are determined by the ratio of the weights b_1', a', and b' to W_1 which are all finite as $\Delta\tau \to 0$. One can hence just use the random number generator to select the exit corner. Given that the outcome of this choice is, for instance, a', the loop head would move to site 2 while flipping spins, which changes the shaded plaquette of type W_1 to be of type W_2. The process would then continue in the downward direction on site 2. If the decay happens before τ_1, the loop head moves to the decay point while flipping spins and then a choice between the possible decay types is made. Given that a decay occurs, the choice of different types of decays is again independent of $\Delta\tau$ as only the ratios matter. As an example, the probability of selecting a' is $a'/(a' + b_2' + b + b_3)$. This type of process is illustrated in Fig. 12. Having made the choice, the process continues, and the loop closes when the loop head reaches the original starting point.

In practice it is convenient to store the spin-flip events in a doubly linked list for each lattice site so that spin flips can be added and removed efficiently. The main computational cost is then to search the site of the loop head and its neighbors for spin transitions.

In zero magnetic field the directed PIM loop algorithm proposed here corresponds exactly to the single-cluster formulation of the ordinary loop algorithm for $-1 \leqslant \Delta \leqslant 1$ [4,16]. This can be seen by setting all bounce weights to zero and $C = -\Delta/4$, and then comparing our weights to Eq. (39) in Ref. [49]. In the language of the usual loop algorithm, our weight a corresponds to horizontal breakups, b to diagonal breakups, and c to vertical breakups. The general algorithm with bounces is more similar to the worm algorithm [3], but the processes by which the worm is propagated through space-time are different and do not correspond to a solution of our directed-loop equations. This will be further discussed in Sec. VII. In Sec. V we will demonstrate that the directed-loop processes, especially with solution B (in both SSE and PIM implementations) lead to much more efficient simulation algorithms.

V. AUTOCORRELATIONS

Autocorrelation functions provide quantitative measures of the efficiency of a Monte Carlo sampling scheme in generating statistically independent configurations. For a quantity Q, the normalized autocorrelation function is defined as

$$A_Q(t) = \frac{\langle Q(i+t)Q(i)\rangle - \langle Q(i)\rangle^2}{\langle Q(i)^2\rangle - \langle Q(i)\rangle^2}, \tag{44}$$

where i and t are Monte Carlo times, for which we will use the unit of 1 MCS (as defined in Sec. II D in the case of SSE, and with an analogous definition for the PIM). The brackets indicate the average over the time i. Asymptotically, the autocorrelation function decays exponentially as $\sim e^{-t/\tau_Q}$, where the asymptotic autocorrelation time τ_Q is given by the slowest mode of the simulation (the transition matrix of the Markov chain) to which the observable Q couples. For short times, the behavior is typically different for different quantities, even if τ_Q is the same. The integrated autocorrelation time is defined according to

$$\tau_{\mathrm{int}}[Q] = \frac{1}{2} + \sum_{t=1}^{\infty} A_Q(t) \tag{45}$$

and is the autocorrelation measure of greatest practical utility [49].

In this section we will present integrated autocorrelation times for some important quantities in several regions of the parameter space of the anisotropic Heisenberg model (1). We cannot present a completely exhaustive study, however, since in addition to the field h and the anisotropy Δ, the autocorrelations also depend on temperature $T/J = \beta^{-1}$ and the lattice size. In addition, in SSE simulations the autocorrelations depend on the constant ϵ in the matrix elements (18). One of our aims here is to find the optimum value of ϵ. We compare simulations with the original general (nondeterministic) SSE operator-loop update [18] (solution A) and the new solution of the directed-loop equations discussed in Sec. III B (solution B). We also present some results obtained with solution B in continuous-time PIM simulations.

The physical quantities that we will focus on here are the magnetization

$$M = \frac{1}{N}\sum_{i=1}^{N} \langle S_i^z \rangle, \tag{46}$$

the uniform magnetic susceptibility

$$\chi_u = \frac{\beta}{N}\left\langle \left(\sum_{i=1}^{N} S_i^z \right)^2 \right\rangle, \tag{47}$$

the staggered susceptibility

$$\chi_s = \frac{1}{N}\sum_{k,l} (-1)^{x_k - x_l + y_k - y_l} \int_0^\beta d\tau \langle S_k^z(\tau) S_l^z(0)\rangle, \tag{48}$$

and the spin stiffness

$$\rho_s = \frac{\partial^2 E(\phi)}{\partial \phi^2}, \tag{49}$$

where $E(\phi)$ is the internal energy per spin in the presence of a twist ϕ in the boundary condition. These quantities and their SSE estimators have been discussed in detail in Ref. [42].

We note again that the definition of an MCS in the generic SSE operator-loop scheme involves some degree of arbitrariness, as was discussed in Sec. II D. There is also a statistical uncertainty due to the statistical determination of the number N_l of operator loops constructed per MCS. In all the SSE simulations discussed here, N_l was adjusted during the equilibration of the simulation so that on an average, $2M$ vertex legs (excluding bounces) were visited in each MCS. The maximum expansion power M was increased if needed

after each equilibration MCS, so that $M=1.25n_{\max}$, where $n_{\max}$ is the highest power n generated so far in the simulation. The statistical uncertainties in N_l and $n_{\max}$ imply some fluctuations in the definition of an MCS. This, in turn, results in fluctuations in the results for the integrated autocorrelation times that can be larger than their statistical errors. Typically, these fluctuations are only a few percent, however, and are hence not problematic.

In the PIM simulations, we adjusted N_l so that on average the total length (again excluding bounces) of all N_l loops in a MCS is equal to βN, the space-time volume. The definitions of an MCS in SSE and PIM simulations are hence similar but not identical. One reason why it is difficult to construct exactly comparable MCS definitions in the SSE and the PIM is that the diagonal single-operator updates carried out separately in SSE are in effect accomplished during the loop construction in the PIM. Another difference is that there is no adjustable constant ϵ in the PIM. In Ref. [44] an alternative approach of normalizing the autocorrelation times by the actual number of operations performed was used. However, also this definition may be ambiguous since it depends on the details of the implementation, and there are also differences in the actual CPU time consumed, depending on the mix of operations (integer, floating point, boolean, etc.). These issues are not of major significance in the calculations we present below, but should nevertheless be kept in mind when comparing autocorrelations for the two methods.

The reminder of this section is organized as follows. In Sec. V A we first discuss SSE simulations of the 1D Heisenberg model in an external field. In Sec. V B we consider SSE simulations of 2D systems in fields and with anisotropies. PIM results for both 1D and 2D systems are presented in Sec. V C. We have also studied several isotropic systems at critical points and extracted the dynamic exponent of the simulations. We discuss these results in Sec. V D.

A. SSE simulations in 1D

When the constant $\epsilon=0$, the vertices with all spins up or all spins down are excluded from the SSE configuration space when $h=0$, since the corresponding matrix elements (18) then vanish. When $h>0$, the all-up vertex is again allowed. With $\epsilon>0$ all vertices are allowed and the propagation of the loop is then more random. We here begin by studying how the simulation efficiency depends on ϵ in the case of the 1D Heisenberg model ($\Delta=1$) in a field.

Figures 13 and 14 show the field dependence of the integrated autocorrelation time of the magnetization and the staggered susceptibility in simulations of chains with 64 sites at inverse temperature $\beta=16$. As shown in the inset of Fig. 13, at the $T=0$ saturation field ($h_{\rm sat}/J=2$ in 1D), the magnetization is about 10% from saturation at this temperature. The staggered susceptibility is peaked at $h=0$, reflecting the fact that the staggered spin-spin correlation function for spin components parallel to the external field is dominant only in the absence of a field. In the case of solution *A* simulations, the effect of increasing ϵ from 0 is an initial small drop in $\tau_{\rm int}[M]$ for fields $h\lesssim 0.8$ and a small increase at higher fields. There is a substantial increase in $\tau_{\rm int}[\chi_s]$ for weak fields. As ϵ is further increased there is a small increase in $\tau_{\rm int}[M]$ also for weak fields. In contrast, with solution *B*, increasing ϵ has favorable effects on both autocorrelation times up to the highest ϵ studied here. The effects are very small for high fields, however, since there the autocorrelation time is already close to its lower bound 0.5 when $\epsilon=0$. For

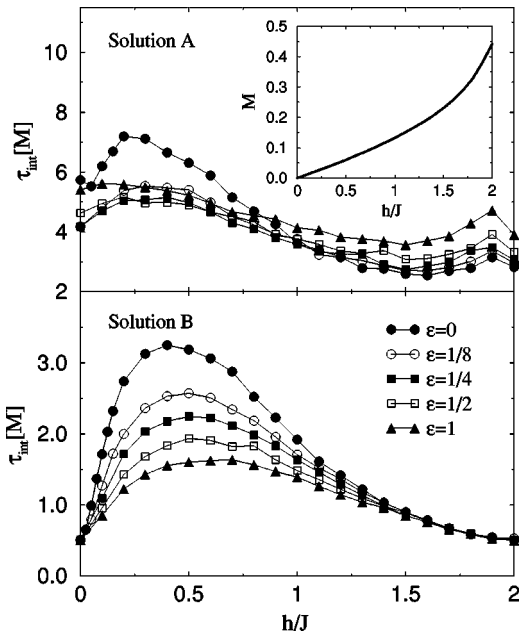

FIG. 13. Integrated autocorrelation times vs external field for the magnetization of an $N=64$ Heisenberg chain at $\beta=16$. The upper and lower panels show results of simulations using solutions *A* and *B*, respectively. Several values of the constant ϵ were used, as indicated by the legends in the lower panel. The inset shows the magnetization itself.

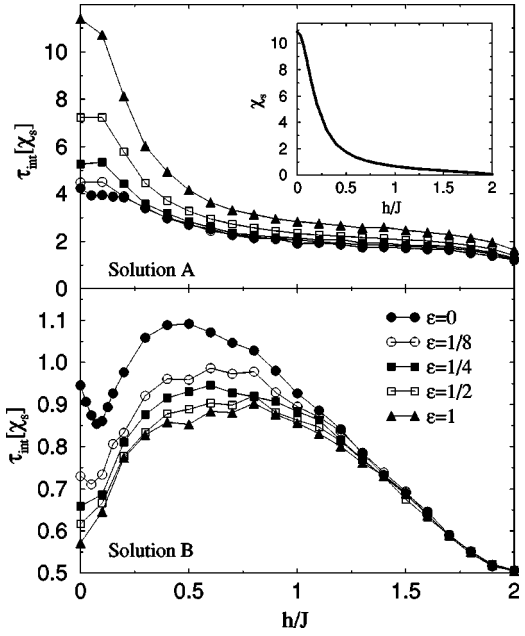

FIG. 14. Integrated autocorrelation times vs external field for the staggered susceptibility of an $N=64$ Heisenberg chain at $\beta=16$. The inset shows the staggered susceptibility.

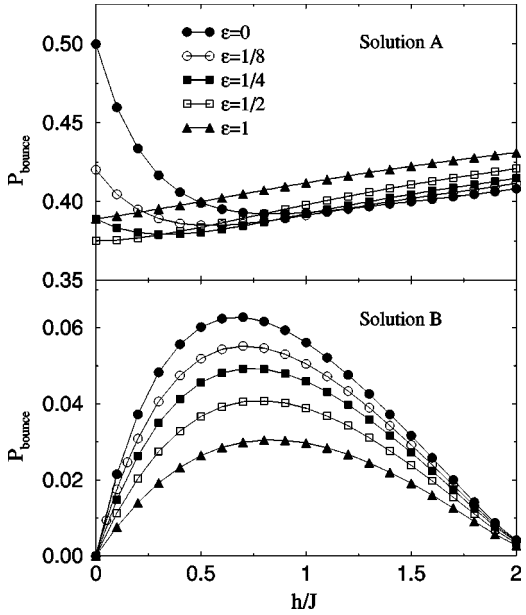

FIG. 15. Bounce probabilities in solution-*A* and -*B* simulations of an $N=64$ Heisenberg chain at $\beta=16$, using different values of ϵ.

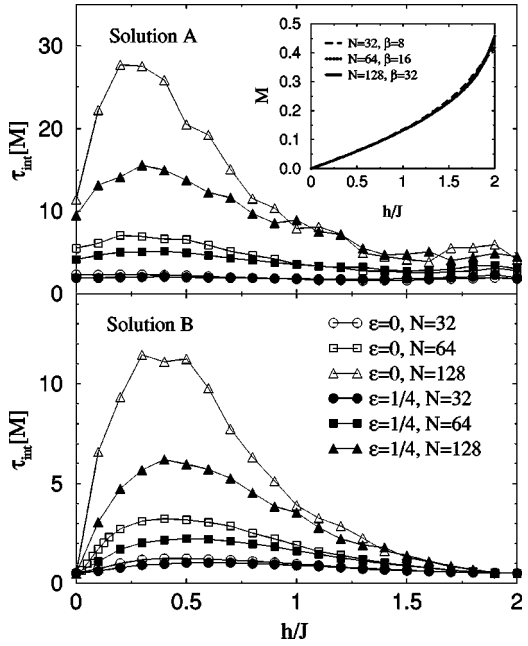

FIG. 16. Integrated autocorrelation times for the magnetization in simulations of chains of different lengths N at inverse temperature $\beta=N/4$. The inset shows the magnetization.

all ϵ values, the autocorrelation times are considerably shorter with solution *B* than with solution *A*. This shows that the strategy of decreasing the probability of the bounce processes in the operator-loop construction is working. The effects are particularly pronounced at and close to $h=0$, where the shortest autocorrelation times with solution *B* are only about 10% of those with solution *A*.

In Fig. 15 we show the probability of bounces in the simulations (P_{bounce} is the fraction of bounces, including length-0 loops). The behavior reflects that of the autocorrelation times. With solution *B*, P_{bounce} decreases monotonically with ϵ for all fields, whereas with solution *A* the behavior is nonmonotonic. In solution *B*, the vanishing of P_{bounce} both in the limits $h\to 0$ and $h\to h_{\mathrm{sat}}/J$ (at $T=0$) follows by construction, as discussed in Sec. II. With solution *A* the bounce rate is large in these limits.

For weak fields, a small $\epsilon>0$ has favorable effects on the magnetization autocorrelations both with solutions *A* and *B*. In the case of solution *B*, both $\tau_{\mathrm{int}}[M]$ and $\tau_{\mathrm{int}}[\chi_s]$ continue to decrease also when $\epsilon\approx 1$, as seen in Figs. 13 and 14. Nevertheless, it is not practical to use a very large ϵ since the average expansion order $\langle n\rangle$ (and hence the operator sequence size M) has a contribution $\epsilon\beta N_b$, and there is a similar increase in the number of operations needed to carry out 1 MCS. However, Figs. 13 and 14 indicate that even a small value ($\epsilon\sim 1/4$) gives a significant improvement of the magnetization autocorrelations relative to $\epsilon=0$ simulations. We find that this behavior persists also for larger system sizes and lower temperatures. Figure 16 shows $\tau_{\mathrm{int}}[M]$ for different system sizes N at inverse temperature $\beta=N/4$, using both $\epsilon=0$ and 1/4. The advantage of $\epsilon=1/4$ becomes more pronounced with increasing system size. For $N=128$ the maximum $\tau_{\mathrm{int}}[M]$ is reduced by about 50% for both solutions *A* and *B*. The relative advantage of solutions *B* over *A* is again the most dramatic in the limit $h\to 0$. In both solutions, the autocorrelation time is rather strongly peaked, with the peak position for the largest systems at slightly higher fields for solution *B*. The reason for this type of field dependence is not clear and deserves further study. It cannot be ruled out that a still more efficient directed-loop solution could be found at intermediate field strengths (which would imply that minimizing the bounce probability does not necessarily lead to the most efficient algorithm).

When the temperature becomes small compared to the finite-size gaps in the system, a step structure in the magnetization versus field curve can be clearly resolved, as is shown in Fig. 17(a). These steps are also reflected in the autocorrelation time, as shown in Fig. 17(b). There are sharp maximas in the regions where the magnetization switches between two values. Exactly at $T=0$, the autocorrelation function (44) for the magnetization is ill defined, since there are then no fluctuations in M on the magnetization plateaus. However, we find that the limit $T\to 0$ is well behaved in the simulations. Exactly at the switching fields, $\tau_{\mathrm{int}}[M]$ appears to diverge, however, showing that tunneling between the two equal-probability magnetization sectors becomes rare. Figure 17(c) shows the average size of the operator loops. There are maxima at the switching fields, with the peak heights growing as the temperature is lowered. On the plateaus, the loop size does not change much with β. A divergence of the average loop size with β at the switching fields can be expected, since in order for the magnetization to change, the loop has to wrap around the system in the SSE propagation (or imaginary time) direction, which is of length $\sim\beta$. The convergence of the average loop size on the plateaus can be understood on the same grounds. Apart from the oscillations, there is also a significant increase in the loop size as the field increases.

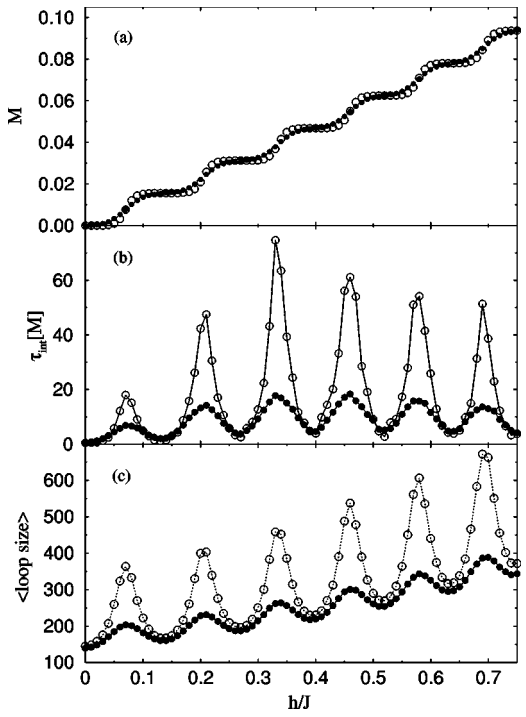

FIG. 17. Magnetization vs field of an $N=64$ chain (a), the corresponding integrated autocorrelation time (b), and the average size of the operator loops (c). Solid and open circles show results at $\beta=64$ and 128, respectively. The simulations were carried out with solution B of the directed-loop equation with $\epsilon=1/4$.

Distributions of loop sizes at $\beta=128$ are shown in Fig. 18 for field strengths corresponding to magnetization plateaus ($h/J=0$ and 0.14) and switching fields ($h/J=0.07$ and 0.21). At $h=0$, there are no bounce processes and this appears to be reflected as a qualitatively different loop size distribution than for $h>0$, with no very large loops and a larger probability of sizes in the range 2^8-2^{11}. For all fields, there is quite a sharp crossover beyond which the probability

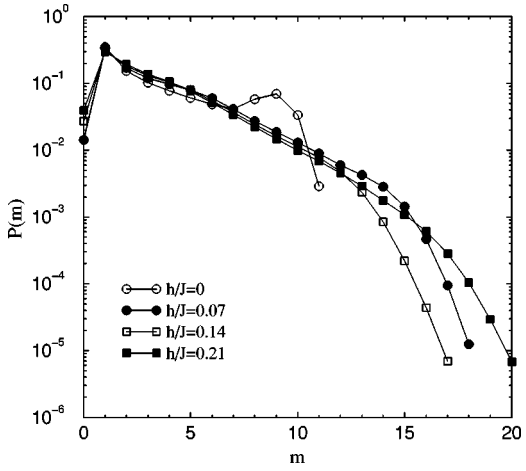

FIG. 18. Loop size distribution for $N=64$ chains at $\beta=128$ and different field strengths (solution B simulations with $\epsilon=1/4$). $P(m)$ is the cumulative probability of loop sizes between 2^m (0 for $m=0$) and $2^{m+1}-1$.

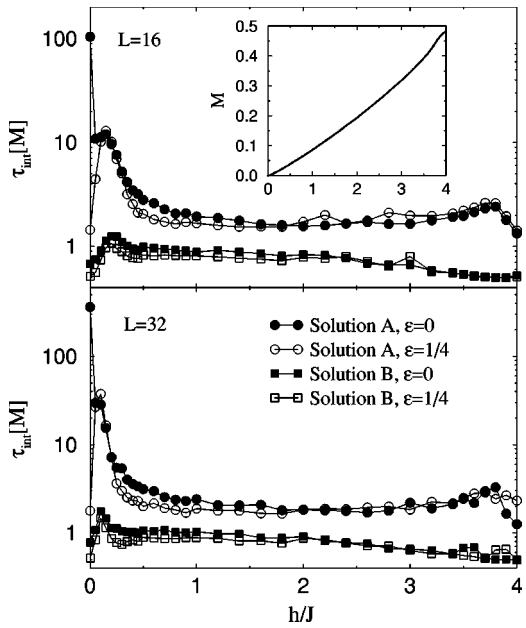

FIG. 19. Integrated autocorrelation times for the magnetization in simulations of the 2D Heisenberg model in a magnetic field, using solutions A (circles) and B (squares) and constants $\epsilon=0$ (filled symbols) and 1/4 (open symbols). The inset shows the magnetization (the differences in M between $L=16$ and $L=32$ are very small at the inverse temperature $\beta=8$ used here).

becomes very small. Problems with loops that do not close [40,44] are therefore absent in this case. We did not have to impose any maximum size during the loop construction in any of the simulations discussed in this paper.

In the studies of the 1D Heisenberg model in a field that we have presented here, the new solution B is clearly better than solution A, although the difference is very large only for h close to 0 (but significant also for $h\rightarrow h_{sat}$). Already with solution A the autocorrelation time for the magnetization is very short compared to other approaches. With the continuous-time worm algorithm $\tau_{int}[M]$ is close to 100 even for system sizes as small as $N=10$ and $N=20$ [41].

B. SSE simulations in 2D

For the 2D XXZ model (on periodic $L\times L$ lattices), we have calculated autocorrelation times versus the field strength in systems with isotropic couplings ($\Delta=1$, $0\leqslant h\leqslant h_{sat}=4J$), Ising-anisotropic systems in zero field ($\Delta\geqslant 1$, $h=0$), and the XY model in zero and finite field ($\Delta=0$, $h/J=0,1/2$).

Figure 19 shows the field dependence of the autocorrelation time for the magnetization of $L=16$ and $L=32$ systems at inverse temperature $\beta=8$. With solution A at $\epsilon=0$, a sharp drop in the autocorrelation time can be noted immediately when h becomes nonzero. It is not surprising that the algorithm at $h=0$ is inefficient, since the only processes occurring here are the switch-and-reverse and the bounce (see Fig. 3). The bounce probability is high if it is not excluded ''by hand,'' which would yield the much more efficient deterministic loop rules. With the bounce included, the actual closed loop is still deterministic, but during its construction

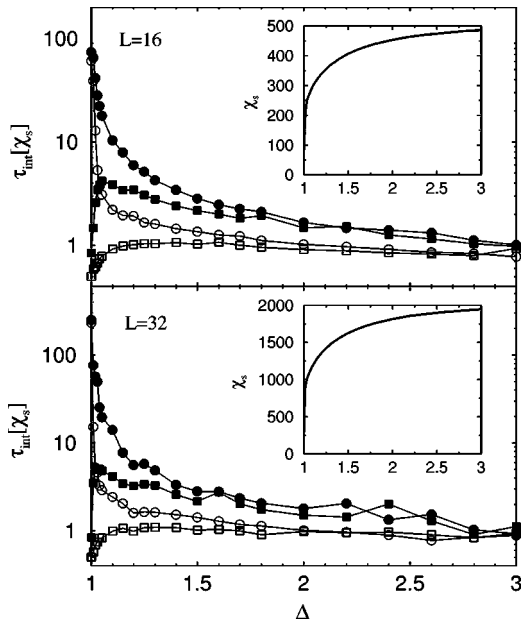

FIG. 20. Integrated autocorrelation times for the staggered susceptibility in simulations of the 2D anisotropic Heisenberg model at $\beta=8$. The symbols indicate solutions *A*, *B*, and $\epsilon=0,1/4$ in the same way as in Fig. 19.

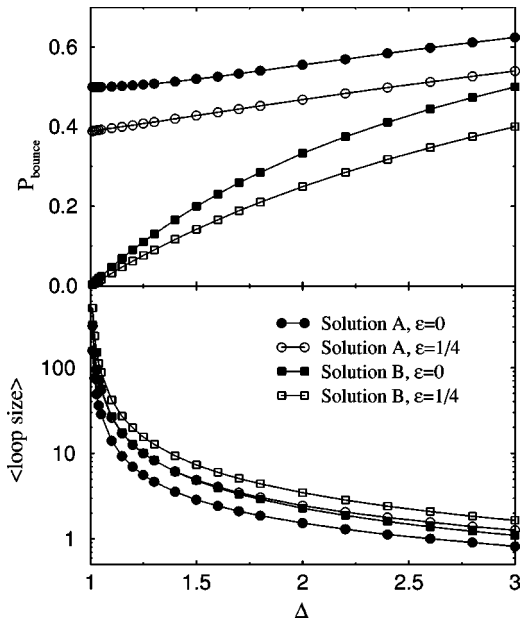

FIG. 21. Upper panel: bounce probabilities in simulations of a 32×32 anisotropic Heisenberg model at $\beta=8$. Lower panel: the average loop size in the same simulations.

the propagating open end oscillates randomly back and forth along the *de facto* deterministic trajectory until the loop closes or is annihilated via backtracking all the way to the starting point. Once h is nonzero, the loops become manifestly nondeterministic (since an additional vertex path becomes allowed) and apparently, as seen in Fig. 19, even for a very small h the simulation is much more efficient. This is in contrast to the 1D case (see Fig. 16), where solution *A* with $\epsilon=0$ is reasonably efficient even for $h=0$ and $\tau_{\mathrm{int}}[M]$ increases when h is turned on. This difference between the 1D and 2D simulations may be related to the loop sizes (although the full explanation probably is more complex and is related to the different physical properties of the systems, which are reflected in the loop structures). In one dimension, the loops are relatively small, and for a small h a large fraction of the constructed loops are then identical to the deterministic ones at $h=0$. In two dimensions the loops are much larger, and then even a small h can allow most paths to ''escape'' from the $h=0$ deterministic loop trajectories so that there are not as many propagations back and forth along the same path as at $h=0$. Using a nonzero ϵ also makes the path nondeterministic, and Fig. 19 shows very favorable effect of using $\epsilon=1/4$ in solution *A* at $h=0$. For higher fields, there are only very minor advantages of a nonzero ϵ, which is also in contrast to the 1D case. As in the 1D case, solution *B* reduces the autocorrelation times very significantly at weak fields, and substantially also at higher fields. The differences between $\epsilon=0$ and 1/4 in solution *B* are small at all fields, however.

Figure 20 shows autocorrelation times for the staggered susceptibility of Ising-anisotropic systems in zero field at $\beta=8$. Solution *B* performs significantly better than solution *A* for $\Delta\lesssim1.5$, but only marginally better at higher Δ. In this system $\epsilon=0$ implies that the closed loops are *de facto* deterministic for all anisotropies (the only allowed $\epsilon=0$ vertex processes are again the bounce and the switch-and-reverse). However, the symmetry of flipping and flipping back loops is broken when $\Delta>1$ and the *de facto* fixed structure of the closed loops is not taken into account during their construction, neither with solution *A* nor *B* (doing this would correspond to neglecting the bounces, constructing a deterministic loop, and then taking Δ into account in a Metropolis acceptance probability for actually flipping the loop, in a way analogous to what has been done with the standard worldline loop method for weak magnetic fields [25]). Solution *B* minimizes the bounce probability and hence leads to more directed paths and, therefore, closing of the loops in fewer steps (and hence a larger number of completed loop in an MCS as defined here). Bounce probabilities are shown in the upper panel of Fig. 21. When $\epsilon>0$ the loops become manifestly nondeterministic, leading to significantly reduced autocorrelation times. The bounce probabilities are also reduced, but for both solutions, P_{bounce} still becomes large as Δ is increased. Nevertheless, the autocorrelation times continue to decrease. We do not expect this to be the case as $\Delta\to\infty$, where the model at fixed β reduces to the classical Ising antiferromagnet at temperature $T\to0$. In that limit, a classical single-spin flip would correspond to flipping spins on all SSE vertices on a given site (the number of which scales as $\beta\Delta$), which would be a slow process since the bounce probability is high. The lower panel of Fig. 21 shows that the average loop size becomes very small for large Δ. The algorithm clearly does not reduce to a classical Swendsen-Wang or Wolff cluster algorithm as $\Delta\to\infty$ (in the classical algorithms the cluster size $\to N$ as $T\to0$). However, at higher temperatures the algorithm could easily be supplemented with a cluster update which corresponds exactly to the clas-

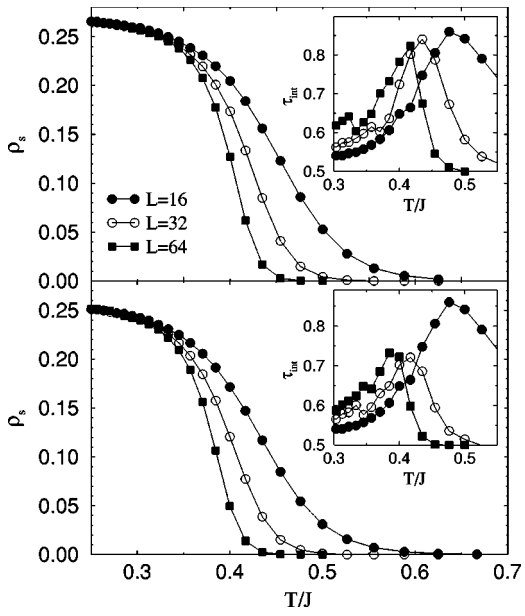

FIG. 22. Spin stiffness of the XY model at zero external field (upper panel) and at $h/J=0.5$ (lower panel). The insets show the corresponding integrated autocorrelation times. Solution B with $\epsilon=\epsilon_{\min}$ (see Table I) was used in all cases.

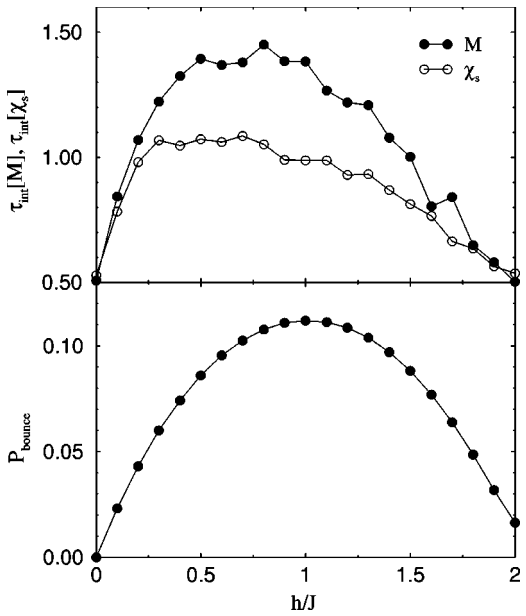

FIG. 23. Upper panel: integrated autocorrelation times vs external field for the magnetization (solid circles) and staggered susceptibility (open circles) in PIM simulations of an $N=64$ Heisenberg chain at $\beta=16$. Lower panel: bounce probability vs external field in the same simulation.

sical one (a multispin generalization of the flips of free spins, where clusters of spins connected to each other by operators in S_M can be flipped simultaneously without changing the weight if $\epsilon=0$ and $h=0$). As in the standard world-line loop algorithm [49], it is also possible to include loop freezing in the deterministic loop algorithm.

Note that there is essentially no structure in the solution B autocorrelation time for $\epsilon=1/4$ in Fig. 20, in spite of the fact that the scan over anisotropies should cross an Ising-type transition to an ordered state. At $\Delta=3$ the antiferromagnetic order is already at $\approx 97\%$ of the maximum (classical $T=0$) value, as can be inferred from the insets of Fig. 20 by using Eq. (48).

For the XY model ($\Delta=0$), the directed-loop equations have a solution without bounces for all fields up to the saturation field. We find that the resulting algorithm is very efficient, with autocorrelation times smaller than 1 for all system sizes and temperatures that we have studied. Figure 22 shows results for the spin stiffness as a function of temperature for zero field as well as at $h/J=0.5$. The corresponding autocorrelation times are peaked around the Kosterlitz-Thouless (KT) transition temperature but do not grow with the system size. The KT transition in the $h=0$ system has been studied to high accuracy using a continuous-time world-line loop algorithm, with the result $T_{\mathrm{KT}}/J\approx 0.342$ [24]. Our $h=0$ data are in complete agreement with the previous results. We find that the data for $h=0.5$ shown in Fig. 22 can be collapsed onto the $h=0$ data if T and ρ_s are both scaled by the same factor (≈ 1.05 for $h/J=0.5$), in accord with the universality of the transition. More extensive results for this model will be presented elsewhere.

C. PIM simulations

Next we will show some results for autocorrelation times obtained using the PIM implementation of the directed-loop algorithm. To make a reasonable comparison with the autocorrelation times for the SSE, we will also define a MCS in the PIM so that it includes N_l loops. N_l is determined such that on average the total path length, excluding the first path segment immediately following each bounce, of all N_l loops in an MCS is equal to the space-time volume βN (in the PIM, each path segment has a length in imaginary time, in contrast to the SSE where the steps are just counted). This definition is chosen so that it corresponds reasonably close to the definition used in the SSE. However, it could be argued that a better definition of the total path length would be to add all the path segments; but instead of excluding the segment immediately following a bounce, one would subtract the part of the path immediately following a bounce that overlaps with the path segment preceding the bounce (with special care taken for consecutive bounces). This would more accurately take into account the fraction of spins actually flipped. We have here used the first definition of the MCS as it corresponds more closely to how we define an MCS in the SSE method (where a different treatment of the bounces could of course also be implemented—see Sec. II D).

Generally speaking the computer implementation of the PIM is more complex than SSE, as it is always necessary to keep track of the spin states on neighboring sites in the PIM. This is not required in the SSE formulation, where the vertices contain all the information needed. Therefore our computer code for the PIM is not as efficient as the SSE code in generating a single MCS, and so we will be content in this section to show just a few PIM autocorrelation results. As solution A of the directed-loop equations was already shown above to be much less effective than solution B, we will in this section just show results for solution B.

Figure 23 shows the integrated autocorrelation times

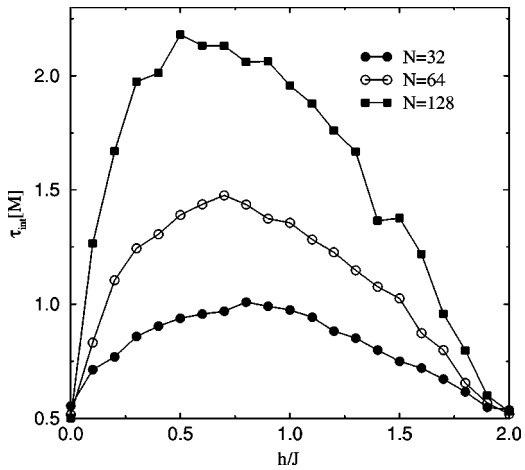

FIG. 24. Field dependence of the integrated autocorrelation times for the magnetization in PIM simulations of chains of different lengths N at inverse temperature $\beta=N/4$.

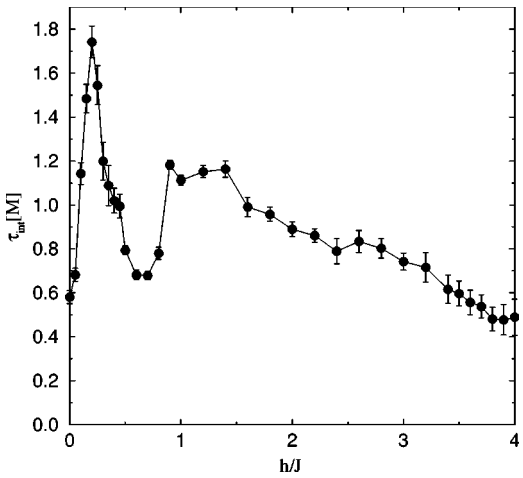

FIG. 25. Integrated autocorrelation times (PIM) vs external field for the magnetization of a 16×16 Heisenberg square lattice at $\beta=8$.

$\tau_{\rm int}[M]$ and $\tau_{\rm int}[\chi_s]$ for a 64-site Heisenberg chain ($\Delta=1$) at inverse temperature $\beta=16$ as functions of the magnetic field. Comparing the results in Figs. 13 and 14 with the SSE, it is seen that $\tau_{\rm int}[M]$ is comparable to the $\epsilon=1$ case while $\tau_{\rm int}[\chi_s]$ is more similar to the $\epsilon=0$ curve, except close to $h=0$, where it also behaves more like the $\epsilon=1$ case. The lower panel of Fig. 23 shows the field dependence of the bounce probability. $P_{\rm bounce}$ is defined here as the number of bounces divided by the total number of times the path building changes, either by moving to a neighbor site or by back tracking. This measure is not directly comparable to the definition in the SSE case, as the moves c and c'—where the path continues on the same site—are not counted in the denominator of $P_{\rm bounce}$ (they are infinitely many in the PIM). Nevertheless, the general behavior of $P_{\rm bounce}$ versus h is the same for the two methods.

In Fig. 24 we have plotted $\tau_{\rm int}[M]$ as a function of magnetic field for different chain sizes N. In all cases $\beta=N/4$. As in the SSE case (Fig. 16) we see an increase in $\tau_{\rm int}[M]$ with system size for small to intermediate fields. However, the maximum PIM autocorrelation times are about 50% smaller than in the SSE $\epsilon=1/4$ case.

We have also carried out simulations of the 2D Heisenberg model using the PIM. In Fig. 25 we show results for $\tau_{\rm int}[M]$ for a 16×16 lattice at $\beta=8$. Here the behavior is almost identical to the SSE results shown in Fig. 19, where there is only a small dependence on ϵ.

From these examples it can be seen that the PIM generally has shorter autocorrelation times than SSE in cases where the SSE results show a significant dependence on the constant ϵ. In some sense the PIM corresponds to the $\epsilon\to\infty$ limit of SSE, as in this limit the continue-straight processes also dominate the loop construction in SSE. In cases where the SSE autocorrelations converge slowly to their $\epsilon=\infty$ limit, the PIM approach may hence be more efficient (since in SSE the computation time for 1 MCS grows linearly with ϵ in this limit). However, in assessing a method's efficiency one should also take into account the cost of performing a single MCS. This of course depends heavily on the actual computer implementation of the directed-loop algorithm. That is, what kind of data structures are used to represent the spin and vertex configurations, what kind of search algorithms are used for finding spin states at a given time in the PIM, etc. While we do not attempt to compare the PIM and SSE in this respect here, it is quite clear that it is often easier to find a fast and effective implementation for the SSE than for the PIM. We also note that the convergence to the $\epsilon\to\infty$ limit in the SSE is relatively fast in all cases we have studied so far. The convergence appears to be slowest in 1D, but even there the reduction of the autocorrelation times becomes small beyond $\epsilon=1$, where they are similar to the PIM autocorrelations.

D. Dynamic exponent

An interesting question is how the autocorrelation time diverges with the system size in simulations at a critical point. The 1D Heisenberg model at $h=0,T=0$ exhibits power-law $(1/r)$ decay of the staggered spin-spin correlation function and is hence a quantum critical system [50]. We have studied the integrated autocorrelation time for the staggered spin susceptibility in this model as the system size N is increased and the inverse temperature $\beta=N/4$. The staggered susceptibility should couple to the slowest mode of the simulation, and its autocorrelation time is therefore expected to diverge asymptotically according to a power law

$$\tau_{\rm int}[\chi_s]\sim\beta^z, \tag{50}$$

where z is the dynamic exponent of the simulation. Note that it is essential here that β and N are taken to infinity at a fixed ratio (as the physical dynamic critical exponent relating space and imaginary time is 1). It is interesting to compare SSE simulations with solution B at different ϵ values (we do not consider solution A here since it is much less efficient than solution B). It is also interesting to compare the two possible ways of flipping the loops when $\epsilon=0$. At $h=0,\epsilon=0$, solution B reduces to the deterministic operator loop [18]. As discussed in Sec. II D, instead of constructing a

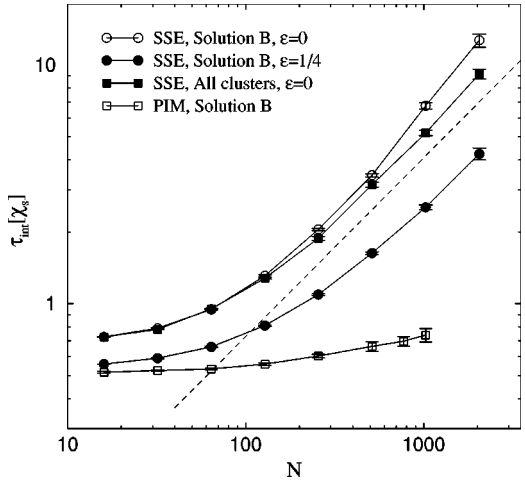

FIG. 26. Autocorrelation times for the staggered susceptibility of the isotropic Heisenberg chain at $\beta=N/4$ obtained in SSE and PIM simulations with solution *B*. The dashed line has slope 0.75.

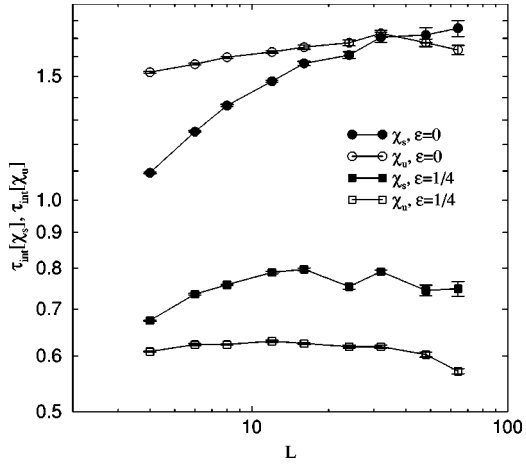

FIG. 27. SSE autocorrelation times for the staggered and uniform susceptibility of the Heisenberg bilayer close to its quantum critical point ($J_\perp/J=2.5225$ was used). The inverse temperature $\beta=L/J$.

fixed number of loops per MCS at random, all loops can then be constructed and flipped independently of each other with probability 1/2. This is analogous to the Swendsen-Wang [46] algorithm for the classical Ising model. For the Ising model, it is known that it is more efficient (i.e., z is smaller) to construct the clusters one by one using the Wolff algorithm [45].

In Fig. 26 we show the results of solution *B* simulations with $\epsilon=0$ and 1/4 along with the results from $\epsilon=0$ simulations, where all clusters were constructed. The autocorrelation times of the two $\epsilon=0$ simulations are very similar, but for large systems, marginally shorter when all clusters are constructed. Hence, here there is no advantage in constructing the clusters one by one. This is most likely related to the fact that in order to change the loop structure in the SSE simulations at $h=0,\epsilon=0$, diagonal updates also have to be carried out. In the scheme used here, diagonal updates are only performed at the beginning of each MCS, and hence the same loop can be constructed several times in 1 MCS if they are constructed at random. It is then more efficient to construct all loops once. In order to achieve an advantage similar to the Wolff algorithm, one would have to construct a new scheme for the diagonal updates, which certainly could be possible but which we have not yet attempted. As in the other cases we have discussed above, there is a significant improvement when $\epsilon=1/4$ is used in solution *B*. However, the dynamic exponent appears to be the same in all cases, $z\approx0.75$. In Fig. 26 we also show PIM results. It is clear that the autocorrelation times here are significantly shorter, but most likely the dynamic exponent is the same as in SSE. The shorter PIM autocorrelation times are consistent with the 1D results shown above in Secs. V A and V C, and clearly we could also reduce the SSE autocorrelations by increasing ϵ further.

In 2D, a well-studied quantum critical system is the Heisenberg model on two coupled layers (bilayer), with intraplane coupling J and interplane coupling $J_\perp$ [51]. The $T=0$ antiferromagnetic long-range order in this model vanishes at a critical interplane coupling $(J_\perp/J)_c\approx2.525$ [28]. Some autocorrelation results for both SSE and PIM simulations of this model at $J_\perp/J=2.524$ have been presented recently [44] and indicate that the dynamic exponent $z\approx0$ in both methods. Our most recent simulations indicate that $(J_\perp/J)_c\approx2.5225$, i.e., slightly lower than the previous estimate [28]. In Fig. 27 we show integrated autocorrelation times for several quantities at this coupling, using both $\epsilon=0$ and 1/4 in solution-*B* simulations. In the $\epsilon=0$ case, all clusters were constructed in each MCS. We again note significant shorter autocorrelation times in the nondeterministic simulation ($\epsilon=1/4$). However, the deterministic simulation is significantly faster. One deterministic MCS at $\epsilon=0$ typically only requires $\approx50\%$ of the CPU time of a generic solution-*B* MCS at $\epsilon=1/4$. The net gain in simulation efficiency with $\epsilon>0$ is therefore only marginal in this case. All our results are consistent with $z=0$, although with $\epsilon=0$ the convergence to a size-independent behavior is rather slow. We have not carried out PIM simulations of this system.

Next we consider the 3D Heisenberg model, which undergoes a phase transition to an antiferromagnetic state at a nonzero temperature [52]. According to recent SSE simulations, using systems with $N=L^3$ sites and L up to 16, the critical temperature $T_c/J=0.946\pm0.001$ [53]. These simulations were carried out using only local updates. With the operator-loop update, much larger systems can be studied. We have carried out simulations for L up to 48 close to the critical temperature. Based on the results, we believe that T_c is at the low end of the previous estimate, likely very close to 0.944. Figure 28 shows autocorrelation times for the staggered susceptibility and the spin stiffness at $T/J=0.944$, obtained using the deterministic SSE algorithm with $\epsilon=0$ (constructing all clusters during each MCS) and solution *B* with $\epsilon=1/4$. Here the $\epsilon=0$ results are initially consistent with a dynamic exponent $z\approx0.25$, but for the largest sizes there seems to be a change in behavior, possibly a convergence corresponding to $z=0$. The $\epsilon=1/4$ simulation is fully consistent with $z=0$.

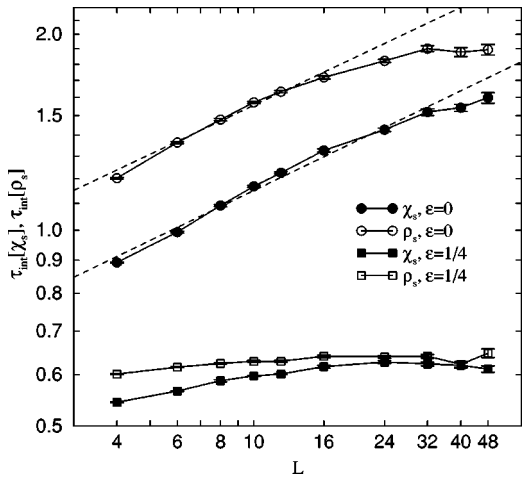

FIG. 28. Autocorrelation times for the staggered susceptibility and the spin stiffness of the 3D Heisenberg model close to its critical temperature ($T/J=0.944$ was used). The lines correspond to scaling $\sim L^{1/4}$.

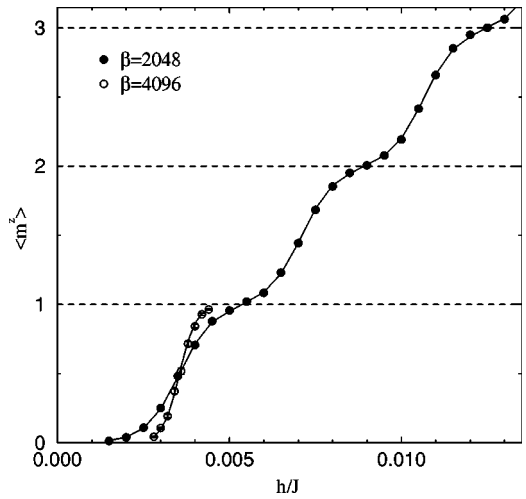

FIG. 29. Total magnetization vs external field in the 2D Heisenberg model with $L=64$ at two inverse temperatures. The curves were calculated using four fitted energies $E_L(S)$ (the same for both curves).

VI. LOW-FIELD MAGNETIZATION OF THE 2D HEISENBERG MODEL

As an example of an application made possible with solution *B* of the directed-loop equations, we here present SSE simulation results for the 2D Heisenberg model in a weak magnetic field. At very low temperatures, the field dependence of the magnetization exhibits a step structure due to the gaps between the lowest-energy states with magnetization $m_z=0,\pm 1,\pm 2,\ldots,\pm N/2$. These gaps can be extracted from the calculated magnetization curve. For the isotropic Heisenberg model, the gaps are exactly the gaps between the degenerate spin multiplets with total spin $S=0,1,\ldots$ in the absence of the field.

In an antiferromagnetically ordered system, such as the 2D Heisenberg model, the energies of the $S>0$ multiplets relative to the $S=0$ ground state should correspond to the excitations of a quantum rotor when $S \ll \sqrt{N}$. The overall energy scale can be related to the uniform (transverse) magnetic susceptibility [54]:

$$E(S)=\frac{S(S+1)}{2L^2\chi_\perp}, \tag{51}$$

where $L^2=N$. The asymptotic validity of this relation has been verified using quantum Monte Carlo estimates for small S and L up to 16 [55,56]. Recently, a slow convergence of the spectrum for $S\sim L$ has been pointed out for the 2D Heisenberg model with spin 1/2 [56]. A systematic study of $E_L(S)$ for systems larger than $L=16$ was not possible, however, because of the large statistical errors in the energy differences.

With the directed-loop algorithm we can instead extract the energy gaps using the field dependence of the magnetization. As we have shown in Sec. V, the new solution *B* shortens the autocorrelation times very significantly for low fields, which is what we need in order to accurately extract the energy levels for S ranging from 0 to $\sim L$. We will present our complete results of such calculations elsewhere. Here we will demonstrate the power of the new method by focusing on the first few levels for system sizes L up to 64, i.e., the number of spins is 16 times larger than in the previous studies [55,56].

In order to see the step structure needed to extract the energy levels $E_L(S)$ for small S, the temperature has to be below the $S=1$ gap, which according to Eq. (51) and previous estimates of the susceptibility ($\chi_\perp \approx 0.065/J$) is $\approx 0.004/J$ for $L=64$. In practice, we have used inverse temperatures β corresponding to roughly (1/10)th of the gap. We have fitted the numerical results to a magnetization curve $\langle m_z\rangle$ calculated using energy levels of the form

$$E_L(S,m_z)=E_L(S)-hm_z, \qquad m_z=0,\pm 1,\ldots,\pm S, \tag{52}$$

at the same temperature as in the simulations. We adjust the energies $E_L(S)$ to give the best match between the calculated and theoretical magnetization curves. Figure 29 shows results for $L=64$ at $\beta=2048$ and 4096. We used the same fitted levels $E_L(S)$ at both temperatures (clearly, the $S=1$ level completely dominates the $\beta=4096$ results, which include only the first magnetization step). As in Ref. [56], we define a spin- and size-dependent susceptibility using the energy levels $E_L(S)$ obtained in this fitting procedure,

$$\frac{1}{2\chi_{L,S}}=\frac{L^2E_L(S)}{S(S+1)}, \tag{53}$$

and extrapolate data for fixed S to infinite size in order to determine the thermodynamic susceptibility $\chi_\perp$. Figure 30 shows our results for $S=1,2,3$ and system sizes ranging from $L=8$ to $L=64$. The results up to $L=16$ agree very well with those presented previously [56], but our statistical errors are considerably smaller. The collapse of the three curves onto each other for large systems demonstrate the validity of Eq. (51) for small S. Extrapolating the three datasets to infinite

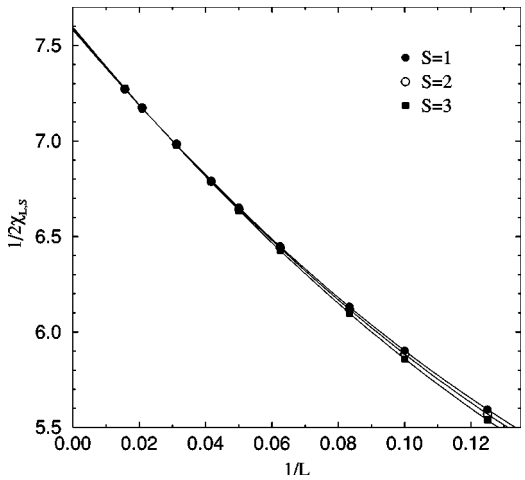

FIG. 30. Inverse susceptibility extracted using the energies of the $S=1$, 2, and 3 multiplets. The curves are quadratic fits.

size gives the susceptibility $\chi_\perp = 0.0659 \pm 0.0002$, again in good agreement with Ref. [56] but with a considerably reduced statistical error.

For the $L=64$ simulations at $\beta=4096$, the CPU time needed to perform one MCS is ≈ 30 s on an Intel Pentium III running at 866 Mhz. The results shown in Fig. 29 are based on $(3-8)\times 10^4$ MCS for each data point.

VII. SUMMARY AND DISCUSSION

We have introduced the concept of directed loops in stochastic series expansion and path-integral quantum Monte Carlo and implemented them for simulations of the S=1/2 *XXZ* model in an external magnetic field. The directionality of the loop reflects the asymmetry between the operation of flipping the spins along the loop and the reverse operation of flipping back those spins. Such an asymmetry is not present in the standard world-line loop algorithm [4,16,49], which as a consequence is restricted to certain regions of the parameter space. Quite generally, there is a hierarchy of three classes of directed loops. In the most general case the loop can backtrack during construction. In some regions of the parameter space the backtracking can be excluded, and in some further restricted regions the loops become symmetric (nondirectional) and reduce to the type of loops previously considered for world-line [4,15,17] and SSE [18] simulations. Hence, the directed-loop framework constitutes a natural generalization of the loop-cluster concept [16]. We have shown that the transitions between the different levels of the hierarchy can be made smooth by minimizing the probability of backtracking when solving the directed-loop equations. We have also demonstrated that the algorithm based on this solution works very well in the full parameter space of the *XXZ* model.

Our scheme appears to be much more efficient than the worm algorithm for continuous-time path-integral simulations [3], which also is applicable in the full parameter space but does not exhibit the three-level hierarchy of the directed loops (at least not in its current formulation). The configuration space involving two moving discontinuities was first introduced in the worm algorithm, where they enabled calculations of off-diagonal correlation functions (Green's functions). It was also the first method that was practically useful in the presence of external fields. It is, however, not the presence of the discontinuities that makes the worm algorithm and directed-loop algorithm applicable in the presence of external fields. One can also think of the construction of the standard world-line loops [16,49] in terms of moving discontinuities, but they are more constrained in their motion and therefore cannot take external fields into account. Hence, it is the rules for moving the discontinuities that determine whether or not a simulation is efficient. The directed-loop equations constitute a framework for optimizing these rules. Below we will comment on the similarities and differences between worms and directed loops.

The operator-loop update previously constructed for SSE simulations [18] corresponds to a particular solution (*A*) of the directed-loop equations. We have here constructed a different solution (*B*), which minimizes the probability of backtracking in the loop construction and therefore is more efficient. The new solution *B* completely eliminates backtracking (bounce processes) in the *XXZ* model for *z* anisotropies $-1<\Delta<1$ up to a finite external field *h* (up to the saturation field for $\Delta=0$ and only exactly at $h=0$ for $|\Delta|=1$). In other interesting parameter regions the bounce probability is typically a few percent or less. Our simulation results show that the new solution can decrease the autocorrelation times by up to an order of magnitude or more in cases where solution *A* is the least efficient (at weak and intermediate magnetic fields and anisotropies). The algorithmic discontinuity of the previous approach (which amounted to using a very efficient deterministic algorithm at $h=0$ and the much less efficient generic solution *A* for $h>0$) is hence avoided with solution *B*, where the bounce probabilities and the autocorrelation times smoothly connect to those of the deterministic algorithm. However, our results also indicate that the deterministic loop construction at $h=0$ is not always the most efficient. With a nondeterministic solution (solution *B* with the constant $\epsilon>0$ in the bond operator) the operator paths becomes more random, which has a favorable effect on the autocorrelations.

In addition to being more efficient in terms of the autocorrelation times measured in units of our defined MCS, solution *B* is also typically faster as the number of operations required to perform 1 MCS is smaller (because of the smaller bounce probability). In terms of ease of implementation, solution *A* is more straightforward as it is directly given in terms of matrix elements of bond operators. In order to implement solution *B* for a new Hamiltonian, one first has to investigate the subclasses of vertices with their directed-loop segments and then minimize the bounce probabilities for all nonequivalent classes. SSE with solution *A* (and other special solutions for Heisenberg and *XY* models) have already been used for a number of different lattices and Hamiltonians [26–29,32–40], but so far we have only investigated solution *B* for the *XXZ* model discussed in this paper. We expect generalizations to a wide range of other models to be relatively straightforward.

In the continuous-time path integral, solution B of the directed-loop equations for zero field and $|\Delta| \leqslant 1$ results in an algorithm identical to the standard world-line loop algorithm [4,49]. The generic algorithm, which includes a probability of back tracking as the loop is constructed, has some features in common with the worm algorithm [3]. The extended configuration space with an open world-line segment (the worm) is the same in the two methods (and is analogous also in the SSE operator-loop construction, although the representation there is discrete rather than continuous). However, there are important differences in the actual processes used to propagate the path (or worm). In the worm algorithm the ''jump'' and ''reconnection'' procedures involve the creation or annihilation of a kink, in which one of the worm ends jumps from one site to another and spins are flipped on finite equal-length segments of imaginary time at both the initial and final sites [3]. The location in time of the worm end does not change in these processes, but is accomplished in separate updates. In the PIM directed-loop scheme, the movement in imaginary time and the creation (or annihilation) of a kink is combined, and in each step spins are flipped on a finite segment of imaginary time at a single site only. This dynamics follows naturally from the vertex representation introduced for the SSE operator-loop algorithm [18], where a single spin is flipped on a link connecting two vertices and the possible sites (the same or a specific neighbor site) and direction (forward or backward) for the next step are dictated by the four legs of the vertex. Here we have directly translated this dynamics into the path-integral simulation by borrowing ides from the continuous-time loop algorithm [4]. The simulation dynamics is hence different from the worm algorithm, and the worm algorithm does not correspond to a solution of the directed-loop equations. Our autocorrelation results show that the directed-loop scheme is much more efficient than the worms in simulations of the Heisenberg chain in a magnetic field, for which our measured autocorrelation times for small systems are almost two orders of magnitude smaller than those reported for the worm algorithm [41]. We expect the superior performance of the directed-loop scheme to be quite general, as the bounce minimization achieved with solution B has no counterpart in the worm algorithm (although it may be possible to develop a generalization). There are, however, very interesting aspects of the worm scheme which could also perhaps be incorporated for the directed loops, e.g., the space-time potential introduced in order to more efficiently measure Green's functions at long distances [3].

Comparing implementations of the directed loops within the SSE and PIM representations, one difference is that in the former there is an adjustable parameter ϵ (a constant added to the bond Hamiltonian operators) which is not present in the latter. We have noted that a nonzero ϵ has generally favorable effects on the autocorrelations in the SSE, but a large value is not practical since the computation time also increases with ϵ. In some sense, the PIM corresponds to SSE with $\epsilon \to \infty$, and one might therefore expect the PIM implementation to be more efficient. However, in practice, the opposite is often true since already a small ϵ in the SSE can give autocorrelation times close to the $\epsilon \to \infty$ limit, and the computation time for 1 MCS can be significantly shorter in SSE. PIM algorithms should be more efficient in cases where the diagonal part of the Hamiltonian dominates in the internal energy, as the PIM configurations (which do not contain diagonal operators) then are smaller than the corresponding SSE configurations [47]. Another important aspect is the ease of implementation and optimization of the simulations for various models. We have found the discrete nature of the SSE configuration space, where the vertices locally contain all information needed to construct the loops, to be a distinct advantage in this respect.

An interesting question is whether the directed-loop approach could be used to further extend the applicability of the meron concept [21] for solving sign problems. We have shown that for the XXZ model, backtracking in the loop construction can be avoided in a larger region of the parameter space than where the loop algorithms previously used for studying merons are applicable (specifically, at nonzero external fields in XY-anisotropic systems). The possibility of generalizing the meron concept to the whole nonbacktracking region should be investigated.

ACKNOWLEDGMENTS

We thank K. Harada and N. Kawashima for pointing out an error in region V of Table I and Fig.9 in an earlier version of this article. A.W.S. would like to thank P. Henelius and M. Troyer for discussions. Both authors acknowledge support from a Nordic network project on Strongly Correlated Electrons at NORDITA. A.W.S. also acknowledges support from the Academy of Finland (Project No. 26175) and the Väisälä Foundation.

APPENDIX: PROGRAM IMPLEMENTATION OF THE SSE METHOD

The computer implementation of a simulation method can of course be done in several different ways and is an issue more technical in nature than the mathematical definition of the underlying algorithm. Nevertheless, for the benefit of readers wishing to quickly construct a simple but efficient simulation program, we here briefly outline the basic aspects of our implementation of the SSE algorithm with the operator-loop update. Some programs are also available online [57].

We first introduce the main data structures used to store the SSE configuration in computer memory. The state $|\alpha\rangle$ is stored as spin$[s]=\pm 1$ representing the up and down spins at the sites s ($s=1,\ldots,N$). The operator-index sequence S_M can be packed into an array sm$[j]$ ($j=0,\ldots,M-1$), with sm$[j]=2b$ and sm$[j]=2b+1$ ($b=1,\ldots,N_b$) corresponding to diagonal and off-diagonal bond-b operators, respectively, and sm$[j]=0$ representing fill-in unit operators. The lattice geometry can be coded into a list of sites $i(b),j(b)$ connected by the bonds b, i.e., site$[1,b]=i(b)$, site$[2,b]=j(b)$. The linked vertices are stored in the form of two lists, one containing the links and one the vertex types. The vertex types vtx$[p]=1,\ldots,6$ ($p=0,\ldots,n-1$) correspond

to the six vertices shown in Fig. 1. The links link[j] ($j=0,\ldots,4n-1$) are arranged such that link[$4p+i$] ($p=0,\ldots,n-1$, $i=0,1,2,3$) contains the link (which is an integer referring to another element in link[ ]) for leg $i+1$ of vertex p [the leg numbers 1,2,3,4 are defined in Eq. (19)]. The double-linked nature of the list implies that if link[a] $=b$ then link[b]$=a$.

The diagonal update is straightforward: For $j=0,\ldots,M-1$, a bond b is generated at random for each sm[j]$=0$, which is changed to sm[j]$=2b$ with the probability (14). If the change is made, the number of bond operators present increases by 1, i.e., $n\rightarrow n+1$. For each diagonal element, i.e., sm[j]>0 and even, the change to sm[j]$=0$ and $n\rightarrow n-1$ is carried out with the probability (15), where $b=$sm[j]/2. If sm[j] is an odd integer, it corresponds to an off-diagonal operator at bond $b=$sm[j]/2 and the corresponding spin states should propagate, i.e., for $a=1,2$, spin[site[a,b]]$\rightarrow-$spin[site[a,b]].

To understand the implementation of the linked vertex list, it is useful to keep in mind Fig. 2 and the numbering of the vertex legs exemplified in Eq. (19). In order to construct the lists link[ ] and vtx[ ], two temporary arrays first[s] and last[s] ($s=1,\ldots,N$) are needed. The element first[s] will contain the first vertex leg on site s, i.e., first[s]$=4p+i$ ($p=0,\ldots,n-1$, $i=0,\ldots,3$) means that the first operator acting on site s is the pth bond operator in sm[ ] and the vertex leg acting on the site is $l=i+1$ (where l will always be 1 or 2, as these are the legs before the operator has acted). In an analogous way, last[s]$=4p+i$ refers to the last operator acting on site s (where $l=i+1$ now will always be 3 or 4, since these are the legs after the operator has acted). All elements are initialized to first[s]$=$last[s]$=-1$ before the construction of the linked list starts. Whereas first[s] will be set at most once (never if no operator acts on site s), last[s] can be updated several times as the operator list sm[j] is searched from $j=0$ to $M-1$. For each sm[j]$\neq 0$, a counter p of the number (minus 1) of bond operators encountered is incremented by 1 and the bond $b=$sm[j]/2 is extracted, giving also the corresponding sites $s_0=$site[1,b] and $s_1=$site[2,b]. Links can be set whenever these sites have already been encountered, i.e., for $a=0,1$, if last(s_a)$\neq-1$, link[$4p+a$]$=$last[s_a] and link[last[s_a]]$=4p+a$. The last occurrence is updated to last[s_a]$=4p+a+2$. If, on the other hand, last(s_a)$=-1$, only the last and first occurrences are recorded, i.e., last[s_a]$=4p+a+2$ and first[s_a]$=4p+a$. The spin list spin[ ] is propagated whenever off-diagonal operators are encountered, so that the vertex types vtx[p] can be recorded (using a map from four leg states to the integers $1,\ldots,6$). After the whole list sm[ ] has been traversed, the list of first occurrences is used in order to connect the links across the propagation boundary, i.e., for each s for which last[s]$\neq-1$, link[last[s]]$=$first[s] and link[first[s]]$=$last[s].

The loop update is repeated N_l times. Each loop starts at a random position $j_0\in\{0,\ldots,4n-1\}$ in the list link[ ]. We will move in link[ ] and the current position will be referred to as j. We hence begin at $j=j_0$ and keep j_0 in order to check at each stage whether the loop has closed or not. The current position corresponds to the vertex number $p=j/4$ and the leg index is $l_i=$mod(j,4) (we can now for convenience number the legs $0,\ldots,3$). This is the entrance leg, and the vertex type is vtx[p]. The exit probabilities given the entrance leg depend on the vertex type and should be stored in a pregenerated table. The probabilities correspond to the particular solution of the directed-loop equations used. It is convenient to use a list of cumulative exit probabilities instead of the individual probabilities, so that for a given entrance leg l_i the exit leg can be obtained by successively comparing the cumulative probabilities prob(l_e,l_i,vtx[p]) for exiting at leg $l_e=0,\ldots,3$ with a random number in the range [0,1]. A corresponding list with updated vertex types is also stored, so that after the exit leg has been fixed the vertex is updated as vtx(p)$\rightarrow$newvtx[l_e,l_i,vtx[p]]. After this, the current position in link[ ] is changed to that corresponding to the exit leg, i.e., $j\rightarrow 4p+l_e$. The loop closes at this stage if $j=j_0$. If it does not close, we move to the leg linked to j, i.e., $j\rightarrow$link[j]. The loop closes also at this stage if $j=j_0$. The two different types of closings, from within the same vertex or from a different vertex, are illustrated in Fig. 4. The possibility of aborting loop updates that become excessively long can be simply taken into account by exiting the loop update routine without mapping the already accomplished changes in the vertex list vtx[ ] back into a new operator list sm[ ] and state spin[ ]. For the XXZ model the loops typically do not become excessively long in practice however, as was demonstrated in a few examples in Sec. V.

After all the N_l loops have been constructed, the updated vertex list vtx[ ] is mapped onto the corresponding new operator list sm[ ]. The bond indices do not change, and therefore one can simply cycle through the positions $j=0,\ldots,M-1$ in the old list one by one, and for each nonzero occurrence, extract the bond $b=$sm[j]/2 and increment an operator counter $p\rightarrow p+1$ (the corresponding position in the vertex list vtx). The operator type, diagonal, or off-diagonal can be coded in a list optype[v]$=0,1$, where $v=1,\ldots,6$ is the vertex type and 0,1 correspond to diagonal and off-diagonal, respectively. The updated operator element is then sm[j]$=2b+$optype[vtx[p]]. The spin list spin[ ] is updated using the list of first occurrences that was generated during the construction of the linked list. For each site s, if first[s]$=-1$ no operator acts on that site and the spin can be flipped, spin[s]$\rightarrow-$spin[s], with probability 1/2. Otherwise, the updated spin state is obtained by extracting the vertex number $p=$first[s]/4 and the leg $l=$mod(first[s],4) corresponding to the site in question. The corresponding spin state can be stored as a pregenerated map, so that spin[s] $\rightarrow$legspin[l,vtx[p]].

We have now described all the basic procedures involved in carrying out 1 MCS using the general operator-loop update. In the special ''deterministic'' cases, where the exit leg is given uniquely by the entrance leg, a number of rather self-evident and trivial simplifications are possible (see discussion in Sec. II D).

The expansion cutoff M is adjusted during equilibration of the simulation by keeping it at $a\times n_{\max}$, where $n_{\max}$ is the largest n reached so far in the simulation and a suitable value

for the factor is $a \approx 1.25$. The number of loops N_l is also adjusted during equilibration, to keep the average total number of vertices visited in 1 MCS close to some reasonable number, e.g., $2\langle n \rangle$, as discussed in Sec. II B. We will not discuss the procedures for measuring operator expectation values here, but published forms for several types of estimators [2,42,44] can be easily translated into the data structures used above.

[1] A.W. Sandvik and J. Kurkijärvi, Phys. Rev. B **43**, 5950 (1991).
[2] A.W. Sandvik, J. Phys. A **25**, 3667 (1992).
[3] N.V. Prokofév, B.V. Svistunov, and I.S. Tupitsyn, Pis'ma Zh. Éksp. Teor. Fiz. **64**, 853 (1996) [JETP Lett. **64**, 911 (1996)]; Phys. Lett. A **238**, 253 (1998); JETP **87**, 311 (1998).
[4] B.B. Beard and U.-J. Wiese, Phys. Rev. Lett. **77**, 5130 (1996).
[5] S.M.A. Rombouts, K. Heyde, and N. Jachowicz, Phys. Rev. Lett. **82**, 4155 (1999).
[6] M. Suzuki, Prog. Theor. Phys. **56**, 1454 (1976).
[7] M. Suzuki, S. Miyashita, and A. Kuroda, Prog. Theor. Phys. **58**, 1377 (1977).
[8] M. Barma and B.S. Shastry, Phys. Rev. B **18**, 3351 (1977).
[9] J.E. Hirsch, R.L. Sugar, D.J. Scalapino, and R. Blankenbecler, Phys. Rev. B **26**, 5033 (1982).
[10] J.J. Cullen and D.P. Landau, Phys. Rev. B **27**, 297 (1983).
[11] R. Blankenbecler, D.J. Scalapino, and R.L. Sugar, Phys. Rev. D **24**, 2278 (1981).
[12] D.C. Handscomb, Proc. Cambridge Philos. Soc. **58**, 594 (1962); **60**, 115 (1964).
[13] J.W. Lyklema, Phys. Rev. Lett. **49**, 88 (1982).
[14] D.H. Lee, J.D. Joannopoulos, and J.W. Negele, Phys. Rev. B **30**, 1599 (1984).
[15] S. Chakravarty and D.B. Stein, Phys. Rev. Lett. **49**, 582 (1982).
[16] H.G. Evertz, G. Lana, and M. Marcu, Phys. Rev. Lett. **70**, 875 (1993).
[17] N. Kawashima, J.E. Gubernatis, and H.G. Evertz, Phys. Rev. B **50**, 136 (1994); N. Kawashima and J.E. Gubernatis, Phys. Rev. Lett. **73**, 1295 (1994).
[18] A.W. Sandvik, Phys. Rev. B **59**, R14 157 (1999).
[19] O.F. Syljuåsen, Phys. Rev. B **61**, R846 (2000).
[20] J.E. Hirsch and R.M. Fye, Phys. Rev. Lett. **56**, 2521 (1986); R.M. Fye and R.T. Scalettar, Phys. Rev. B **36**, 3833 (1987).
[21] S. Chandrasekharan and U.-J. Wiese, Phys. Rev. Lett. **83**, 116 (1999).
[22] J. Cox, C. Gattringer, K. Holland, B. Scarlet, and U.-J. Wiese, Nucl. Phys. (Proc. Suppl.) **83A**, 777 (2000).
[23] P. Henelius and A.W. Sandvik, Phys. Rev. B **62**, 1102 (2000).
[24] K. Harada and N. Kawashima Phys. Rev. B **55**, R11 949 (1998).
[25] M. Troyer and S. Sachdev Phys. Rev. Lett. **81**, 5418 (1998).
[26] A.W. Sandvik and C.J. Hamer, Phys. Rev. B **60**, 6588 (1999).
[27] A.W. Sandvik, Phys. Rev. Lett. **83**, 3069 (1999).
[28] P.V. Shevchenko, A.W. Sandvik, and O.P. Sushkov, Phys. Rev. B **61**, 3475 (2000).
[29] S. Wessel, B. Normand, M. Sigrist, and S. Haas, Phys. Rev. Lett. **86**, 1086 (2001).
[30] K. Kato, S. Todo, K. Harada, N. Kawashima, S. Miyashita, and H. Takayama, Phys. Rev. Lett. **84**, 4204 (2000).
[31] S. Sachdev, M. Troyer, and M. Vojta, Phys. Rev. Lett. **86**, 2617 (2001).
[32] A.W. Sandvik, Phys. Rev. B **66**, 024418 (2002).
[33] S. Wessel, M. Olshanii, and S. Haas, Phys. Rev. Lett. **87**, 206407 (2001).
[34] S. Yunoki, Phys. Rev. B **65**, 092402 (2002).
[35] P. Henelius, P. Fröbrich, P. J. Kuntz, C. Timm, and P. J. Jensen, Phys. Rev. B **66** 094407 (2002).
[36] F. Hébert, G.G. Batrouni, R.T. Scalettar, G. Schmid, M. Troyer, and A. Dorneich, Phys. Rev. B **65**, 014513 (2002).
[37] A. Dorneich, W. Hanke, E. Arrigoni, M. Troyer, and S.C. Zhang, Phys. Rev. Lett. **88**, 057003 (2002).
[38] G. Schmid, S. Todo, M. Troyer, and A. Dorneich, Phys. Rev. Lett. **88**, 167208 (2002).
[39] R.T. Clay, S. Mazumdar, and D.K. Campbell, Phys. Rev. Lett. **86**, 4084 (2001).
[40] P. Sengupta, A.W. Sandvik, and D.K. Campbell, Phys. Rev. B **65**, 155113 (2002).
[41] V.A. Kashurnikov, N.V. Prokofév, B.V. Svistunov, and M. Troyer, Phys. Rev. B **59**, 1162 (1999).
[42] A.W. Sandvik, Phys. Rev. B **56**, 11 678 (1997).
[43] A. W. Sandvik, in *Computer Simulation Studies in Condensed Matter Physics XIV* (Springer-Verlag, Berlin, 2002).
[44] A. Dorneich and M. Troyer, Phys. Rev. E **64**, 066701 (2001).
[45] U. Wolff, Phys. Rev. Lett. **62**, 361 (1989).
[46] R.H. Swendsen and J.S. Wang, Phys. Rev. Lett. **58**, 86 (1987).
[47] A.W. Sandvik, R.R.P. Singh, and D.K. Campbell, Phys. Rev. B **56**, 14 510 (1997).
[48] A.W. Sandvik, S. Daul, R.R.P. Singh, and D.J. Scalapino, e-print cond-mat/0205018.
[49] H.G. Evertz, e-print cond-mat/9707221.
[50] A. Luther and Peschel, Phys. Rev. B **12**, 3908 (1975); I. Affleck, D. Gepner, H.J. Schulz, and T. Ziman, J. Phys. A **22**, 511 (1989).
[51] K. Hida, J. Phys. Soc. Jpn. **59**, 2230 (1990); A.J. Millis and H. Monien, Phys. Rev. Lett. **70**, 2810 (1993); A.W. Sandvik and D.J. Scalapino, *ibid.* **72**, 2777 (1994).
[52] G. S. Rushbrooke, G. A. Baker, and P. J. Wood, in *Phase Transitions and Critical Phenomena*, edited by C. Domb and M. S. Green (Academic, New York, 1974), Vol. 3.
[53] A.W. Sandvik, Phys. Rev. Lett. **80**, 5196 (1998).
[54] H. Neuberger and T. Ziman, Phys. Rev. B **39**, 2608 (1989); D.S. Fisher, *ibid.* **39**, 11 783 (1989); P. Hasenfratz and F. Niedermayer, Z. Phys. B: Condens. Matter **92**, 91 (1993).
[55] K.J. Runge, Phys. Rev. B **45**, 12 292 (1992).
[56] C. Lavalle, S. Sorella, and A. Parola, Phys. Rev. Lett. **80**, 1746 (1998).
[57] http://www.abo.fi/~physcomp/