THE LOOP ALGORITHM[1]

H.G. Evertz
Institut für Theoretische Physik
Technische Universität Graz, 8010 Graz, Austria
e-mail: evertz@itp.tu-graz.ac.at

Abstract

A review of the Loop Algorithm and its generalizations is given, including new results. The loop algorithm is a Monte Carlo procedure which performs *nonlocal* changes of worldline configurations, determined by *local stochastic decisions*. It is based on a formulation of physical models in an extended ensemble of worldlines and graphs, and is related to Swendsen-Wang cluster algorithms. It overcomes many of the difficulties of traditional worldline simulations. Autocorrelations between successive Monte Carlo configurations are reduced by orders of magnitude. The grand-canonical ensemble (e.g. varying winding numbers) is naturally simulated. The continuous time limit can be taken directly. Off-diagonal operators can be measured. Improved Estimators exist which further reduce the errors of measured quantities. For a large class of models, the fermion sign problem can be overcome. The algorithm remains unchanged in any dimension and for varying bond-strengths. It becomes less efficient in the presence of strong site disorder or strong diagonal magnetic fields. Transverse fields are handled efficiently. It applies directly to locally XYZ-like spin, fermion, and hard-core boson models. It has been extended to the Hubbard and the tJ model and generalized to higher spin representations. It has been used in many large scale applications, especially for Heisenberg-like models, where lattices of a million sites and temperatures below $0.001J$ have become accessible, allowing the precise calculation of asymptotic behavior and of quantum critical exponents.

[1]Second edition, January 2000. To be published in *"Numerical Methods for Lattice Quantum Many-Body Problems"*, ed. D.J. Scalapino, Perseus books, Frontiers in Physics.

Contents

1 Introduction and Summary

A pedagogical review of the loop algorithm, its generalizations, and the range of present applications is given, including some new results. The loop algorithm [1, 2, 3, 4] is a Monte Carlo procedure. It is applicable to numerous models in worldline formulation [5]. It overcomes many of the difficulties of traditional worldline simulations. Its main feature is that as Monte-Carlo updates it performs *nonlocal* changes of worldline configurations. These nonlocal changes are determined by *local* stochastic decisions. The loop algorithm is based on a formulation of the worldline system in an extended ensemble which consists of both the original variables (spins or occupation numbers) and of graphs (sets of loops) [1, 2, 3, 6]. It is related to Swendsen-Wang [7] cluster algorithms in classical statistical systems. It has been applied and generalized by a large number of authors. Before we delve into the technical details, let us summarize the main features.

(a) Autocorrelations between successive Monte Carlo configurations are drastically reduced, thereby reducing the number of Monte Carlo sweeps required for a given system, often by orders of magnitude.

(b) The grand-canonical ensemble (e.g. varying magnetization, occupation number, winding numbers) is naturally simulated.

(c) The continuous time limit can be taken [8], completely eliminating the Trotter-approximation. In fact, the loop algorithm can be formulated directly in continuous time.

(d) Observables can be formulated in terms of loop-properties, as so called Improved Estimators, reducing the errors of measured quantities.

(e) Off-diagonal operators can be measured through improved estimators [9].

(f) Transverse fields can be simulated efficiently [10, 11].

(g) For a large class of models, including fermions ones, it has recently been discovered [12] that by clever use of improved estimators the sign problem can be overcome.

(h) Bond disorder and depleted lattices can be trivially included. The algorithm remains completely unchanged in any dimension. Generalizations to higher spin representations [6, 13, 14, 15, 16, 17, 18], and to fermionic models [19, 20, 21, 22, 23, 24] exist.

Each of the points (a)-(g) can save orders of magnitude in computational effort over the traditional local worldline method. In addition, the algorithm is easier to program than traditional worldline updates. The method has some limitations:

(a) Site disorder as well as some other asymmetries in the Hamiltonian will make the original algorithm less efficient. This includes large magnetic field (or chemical potential) and other non "particle-hole-symmetric" terms like e.g. quadratic

interactions of softcore bosons. The difficulty disappears when the disorder or field can be put into transverse direction (section 3.5).

(b) A technical complication, called "freezing", occurs in some models and might possibly reintroduce autocorrelations into the simulations, though this has not been tested for realistic cases.

(c) Long range interactions make the algorithm more complicated and less effective.

Some of the usual limitations of worldline methods also remain in the loop algorithm. The most serious limitation remains (so far) the sign problem, which still occurs in many fermionic models as well as in frustrated spin systems. Generalizations of the meron idea (section 3.10) may help here in the future.

The loop algorithm has already been used for many physical models. The original formulation [1, 2, 3, 4] of the algorithm (in vertex language) applies directly to general spin $\frac{1}{2}$ quantum spin systems in any dimensions, e.g. the 2D Heisenberg model [25], where improved estimators for this algorithm were first used. At the root of the loop algorithm is an exact mapping of the physical model to an extended phase space which includes loops in addition to the original worldlines. In ref. [6] it was shown in a general framework that this mapping is a Fortuin-Kasteleyn-like representation. A related mapping was independently used in a rigorous study of spin models [26]. The general anisotropic XYZ-model (including for example the XY-model quantized along the x-axis) maps to an eight-vertex-like situation for the shaded plaquettes of the worldline lattice. For the corresponding loop algorithm [3], explicit update probabilities are given in ref. [14] and in section 2.7. The method has been adapted and extended to fermion systems like the Hubbard model [19, 24] and to the tJ model [20, 21, 22, 23]. Recently, the meron method [12] to overcome the fermion sign problem was developed and also applied to antiferromagnets spins in a field [11]. The loop algorithm was extended to quantum spin systems with higher spin representation [6, 13, 15, 16, 17, 18], also for the XYZ-case [14], and recently to cases with transverse fields [10]. The extension to more than (1+1) dimensions is immediate [1, 2]: the algorithm remains completely unchanged, only the geometry of the plaquette lattice changes. In ref. [8] it has been shown that the continuous time limit can be taken, and in ref. [9] that all off-diagonal operators can be measured. A recent related development along a somewhat different line is the "Worm" algorithm in continuous time [27], which is applicable to a larger class of models.

There have already been many successful large scale applications of the loop algorithm, to fermionic and especially to numerous Heisenberg-like models, for spin $\frac{1}{2}$ and higher spins, with and without anisotropy, disorder, or impurities, from spin chains up to three dimensional systems, including, e.g., a high statistics calculation of quantum critical exponents on regularly depleted lattices of up to 20000 spatial sites at temperatures down to $T = 0.01$ [28].

Section 2 describes the algorithm, with a brief review of the worldline representation, an intuitive outline of the loop algorithm, and a detailed and partially new step by step formal derivation of the algorithm, followed by a brief summary. We

compute explicitly the update probabilities for the XXZ-model, and give a concise recipe for the Heisenberg antiferromagnet. Ergodicity is treated, and it is shown that in some important cases a transformation of the worldline model to a pure loop model can be done. In section 2.11 we introduce improved estimators, and in section 2.13 we discuss the performance of the loop algorithm, its possibilities and limitations. Section 3 describes a number of generalizations, some of them immediate. The state-of-the-art continuous time version is described (including a brief recipe) in section 3.7. Section 4 mentions related algorithms, and section 5 points to some of the physics problems to which the loop algorithm has been applied so far. Section 6 contains concluding remarks. The appendix provides a prescription to ensure the essential (yet often neglected) requirement for correct Monte Carlo simulations that convergence and statistical errors are properly determined.

2 Algorithm

The loop algorithm acts in the worldline representation [29], which is reviewed e.g. in ref. [5]. We will develop the formal procedure for the general anisotropic (XYZ-like) case. As an example we shall use the particularly simple but important case of the one-dimensional quantum XXZ model [5]. It includes the *Heisenberg model* and *hard core bosons* as special cases. We will see that the same calculation is valid for the loop algorithm in any spatial dimension and already covers most of the important applications. The simplest and most important case is the loop algorithm for the isotropic Heisenberg antiferromagnet, which will be summarized in section 2.8.

2.1 Setup: Worldline representation and equivalent vertex models

Let us first recall the worldline representation for the example of the XXZ model on a one-dimensional chain of N sites [5]. The Hamiltonian is

$$\begin{aligned} \hat{H} \;= \sum_{\langle \mathbf{ij} \rangle} \hat{H}_{i,j} \;&= \sum_{\langle \mathbf{ij} \rangle} J_x (\hat{S}^x_{\mathbf{i}} \hat{S}^x_{\mathbf{j}} + \hat{S}^y_{\mathbf{i}} \hat{S}^y_{\mathbf{j}}) + J_z \hat{S}^z_{\mathbf{i}} \hat{S}^z_{\mathbf{j}} - h \sum_{\mathbf{i}} \hat{S}^z_{\mathbf{i}} \\ &= \sum_{\langle \mathbf{ij} \rangle} \tfrac{J_x}{2} (\hat{S}^+_{\mathbf{i}} \hat{S}^-_{\mathbf{j}} + \hat{S}^-_{\mathbf{i}} \hat{S}^+_{\mathbf{j}}) + J_z \hat{S}^z_{\mathbf{i}} \hat{S}^z_{\mathbf{j}} - h \sum_{\mathbf{i}} \hat{S}^z_{\mathbf{i}} \;, \end{aligned} \tag{1}$$

where $\vec{\hat{S}}_i$ are quantum spin $\frac{1}{2}$ operators at each site i, and $\hat{S}^+_i$, $\hat{S}^-_i$ are the associated raising and lowering operators, and h is a magnetic field. We use periodic boundary conditions.
After splitting the Hamiltonian into commuting pieces

$$\begin{aligned} \hat{H} \quad &= \quad \hat{H}_{even} + \hat{H}_{odd} \\ \hat{H}_{even,odd} \quad &= \quad \sum_{i:\,\text{even,odd}} \hat{H}_{i,i+1} \;, \end{aligned} \tag{2}$$

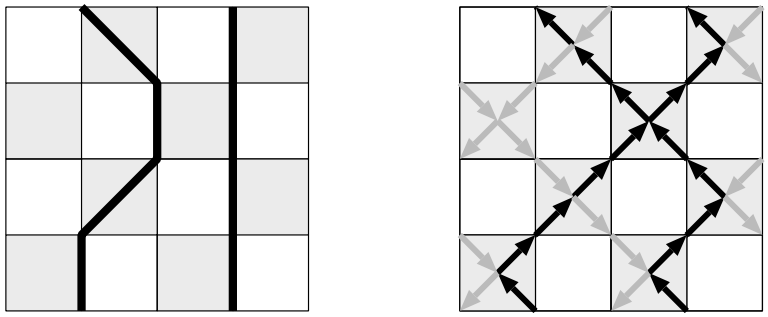

Figure 1: Example of a worldline configuration on a checkerboard lattice of shaded plaquettes. (a) Worldline picture. (b) The *same* configuration as a vertex picture. Space direction (index i) is horizontal, imaginary time direction (index l) vertical. The variables $S^z_{i,l}$ are defined on each lattice site. Worldlines (arrows upwards in time) denote $S^z_{il} = +1$, empty sites (arrows downwards in time) denote $S^z_{il} = -1$. The Hamiltonian $\hat{H}_{i,i+1}$ acts on the shaded plaquettes.

performing a Trotter-Suzuki breakup [30]

$$Z^{XXZ} = tr\, e^{-\beta \hat{H}} = \lim_{L_t \to \infty} Z^{XXZ}_{tr} = \lim_{L_t \to \infty} tr \left(e^{-\frac{\beta}{L_t} \hat{H}_{even}}\, e^{-\frac{\beta}{L_t} \hat{H}_{odd}} \right)^{L_t} , \qquad (3)$$

and inserting complete sets of $\hat{S}^z$ eigenstates, we arrive at the worldline representation

$$Z^{XXZ}_{tr} = \sum_{S^z_{il}} W(\{S^z_{il}\}) = \sum_{S^z_{il}} \prod_p W_p(\{S_p\}) \,, \qquad (4)$$

where the summation $\sum_{\{S^z_{il}\}}$ extends over all "configurations" $\mathcal{S} = \{S^z_{il}\}$ of "spins" $S^z_{il} = \pm 1$, which live on the sites (i,l), $i = 1..N$, $l = 1..2L_t$, of a (1+1)-dimensional checkerboard lattice. The index $l = 1, .., 2L_t$ corresponds to imaginary time. The product $\prod_p$ extends over all shaded plaquettes of that lattice (see fig. 1), and S_p stands for the 4-tupel of spins at the corners of a plaquette $p = (\,(i,l),(i+1,l),(i,l+1),(i+1,l+1)\,)$. The weight W_p at each plaquette [2]

$$W_p(S_p) = \langle S^z_{i,l} S^z_{i+1,l} | e^{-(\Delta\tau)\hat{H}_{i,i+1}} | S^z_{i,l+1} S^z_{i+1,l+1} \rangle \,, \qquad (5)$$

[2] We keep a plaquette index p with W_p to cover spatially varying Hamiltonians.

where $\Delta\tau \equiv \beta/L_t$, is given by the matrix elements[3] [4] of $\hat{\mathcal{H}} = (\Delta\tau)\hat{H}_{i,i+1}$:

$$\begin{aligned}
W(1^+) \equiv a_+ &\equiv \langle ++|e^{-\hat{\mathcal{H}}}|++\rangle = && e^{-\frac{\Delta\tau}{4}J_z} && e^{+\frac{\Delta\tau}{2}h} \\
W(1^-) \equiv a_- &\equiv \langle --|e^{-\hat{\mathcal{H}}}|--\rangle = && e^{-\frac{\Delta\tau}{4}J_z} && e^{-\frac{\Delta\tau}{2}h} \\
W(2^\pm) \equiv c &\equiv \langle +-|e^{-\hat{\mathcal{H}}}|+-\rangle = \langle -+|e^{-\hat{\mathcal{H}}}|-+\rangle = && e^{+\frac{\Delta\tau}{4}J_z} && \cosh\left(\tfrac{\Delta\tau}{2}|J_x|\right) \\
W(3^\pm) \equiv b &\equiv \langle +-|e^{-\hat{\mathcal{H}}}|-+\rangle = \langle -+|e^{-\hat{\mathcal{H}}}|+-\rangle = && e^{+\frac{\Delta\tau}{4}J_z} && \sinh\left(\tfrac{\Delta\tau}{2}|J_x|\right).
\end{aligned} \tag{6}$$

Since $[\hat{H}_{i,i+1}, \hat{S}^z_{tot}] = 0$, there are only the *six* nonvanishing matrix elements given in eq. (6), namely those that conserve $\hat{S}^z_i + \hat{S}^z_{i+1}$, as shown pictorially in fig. 2. Therefore, the locations of $S^z_{il} = 1$ in fig. 1(a) can be connected by continuous worldlines. The worldlines close in imaginary time-direction because of the trace in eq. (3).

For models with fermions or hard core bosons one inserts occupation number eigenstates instead of S^z_{ij}. Nearest neighbor hopping then again leads to the six-vertex case [5] of fig. 2. The term "worldline" derives from this case, since here they connect sites occupied by particles.

We will find it useful to also visualize worldline configurations in a slightly different way, namely as configurations of a *vertex model* [31]. To do this, we perform a one-to-one mapping of each worldline configuration to a vertex configuration. We stay on the same lattice of shaded plaquettes. We represent each spin S^z_{il} by an arrow between the centers of the two shaded plaquettes to which the site (i, l) belongs. The arrow points upwards (downwards) in time for $S^z_{il} = +1(-1)$. The worldline-configuration in fig. 1(a) is thus mapped to the vertex configuration of fig. 1(b). The one-to-one mapping of the worldline-plaquettes is shown in fig. 2. The conservation of $\hat{S}^z_{tot}$ on each shaded plaquette means in vertex language that for each vertex (center of shaded plaquette) two arrows point towards the vertex and two arrows point away from it. If one regards the arrows as a vector field, then this means a

$$\text{condition} \quad \text{“}\mathbf{divergence = zero}\text{”} \quad \text{for the arrows.} \tag{7}$$

Note again that vertex language and worldline language refer to the same configurations; they differ only in the pictures drawn.

We have now mapped the XXZ quantum spin chain to the *six-vertex model* of statistical mechanics [31], though with unusual boundary conditions, since the vertex lattice here is tilted by 45 degrees with respect to that of the standard six-vertex model. Let us look more closely at the case of vanishing magnetic field, $h = 0$. This model has been exactly solved in (1+1) dimensions [31]. The exact phase diagram is shown in fig. 4, in terms of the plaquette weights given in eq. (6) and in fig. 2.

[3] The notation a, b, c is standard for vertex models [31], the notation $1, 3, 2$ (in different order) is that used in refs. [32, 6, 14].

[4] The matrix element b is originally proportional to $\sinh(-\frac{\Delta\tau}{2}J_x)$. It is positive for ferromagnetic XY couplings $J_x < 0$. For antiferromagnetic XY couplings $J_x > 0$, the minus signs cancel on bipartite lattices. Equivalently, b can be made positive on a bipartite lattice by rotating $\hat{S}^{x,y} \to -\hat{S}^{x,y}$ on one of the two sublattices. We have already assumed such a rotation in eq. (6).

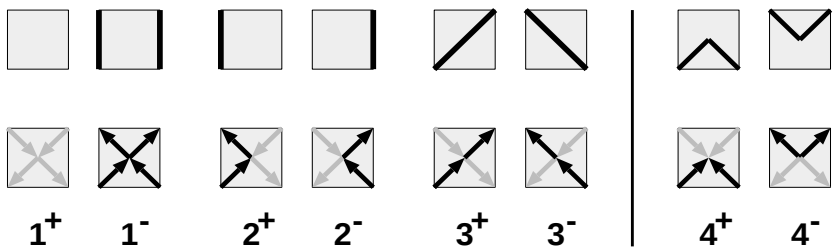

Figure 2: Allowed plaquette configurations S_p. Top line: worldline picture. Bottom line: the same plaquettes in vertex picture. In the XXZ case (= six-vertex case), only the plaquettes with continuous worldlines, $S_p = 1^{\pm}, 2^{\pm}, 3^{\pm}$, have nonzero weight, eq. (6). In the anisotropic XYZ case (= eight-vertex case) the plaquettes $4^{\pm}$ also have nonzero weight.

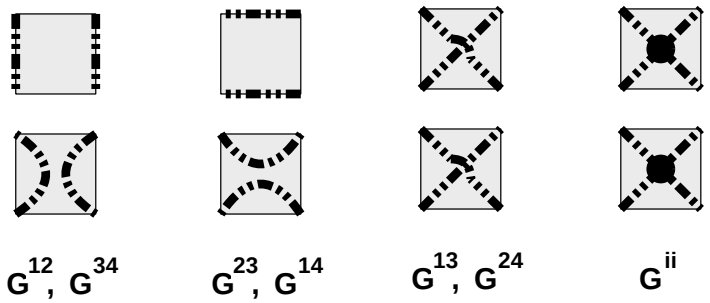

Figure 3: Graphical representations of plaquette breakups. Top row: worldline picture. Middle row: the same graphs in the vertex picture. A "breakup" specifies the direction in which the two loop segments that enter each plaquette will continue. This direction can be vertical, horizontal, or diagonal. Each graphical representation applies in general to two breakups "G^{ij}", as denoted below the pictures. In the six-vertex-case, breakups G^{i4} do not occur, thus the three non-freezing breakups G^{ij}, $i \neq j$ are one-to-one equivalent to the three graphical representations. Breakup "G^{ij}" is allowed (i.e. compatible with continuity of the arrow directions of the worldlines) in plaquette configurations $S_p = i^{\pm}$ or $j^{\pm}$ (see fig. 2). Namely, flipping the two spins on either one of the two lines in the graphical representation of G^{ij}, $i \neq j$, maps between configurations i and j. Breakup G^{ii}, called "freezing", forces all four spins to flip together, thus mapping between i^+ and i^-.

It is interesting to note where the couplings of the Trotter-discretized XXZ-model at $h = 0$ are located in this phase diagram (see eq. (6)): For the Heisenberg antiferromagnet $J_z = |J_x| > 0$ we have $a + b = c$, i.e. we are *on* the Kosterlitz-Thouless line. As $\Delta\tau \to 0$, we approach the point $a/c = 1$, $b/c = 0$. For the Heisenberg ferromagnet $J_z = -|J_x| < 0$ we have $a - b = c$, i.e. we are on the KDP transition line, approaching the same point $a/c = 1$, $b/c = 0$ as $\Delta\tau \to 0$. When $|J_z| < |J_x|$, the same point is approached from inside the massless (XY-like) region. When $|J_z| > |J_x|$, it is approached from below the respective transition line, i.e. from the massive (Ising-like) phase IV when $J_x > 0$ (AF) and from phase I when $J_x < 0$ (FM). Note that the local couplings a, b, c do not change in higher dimensions (see section 3.1).

Anisotropic XYZ case: For generality later on, let us briefly describe the anisotropic

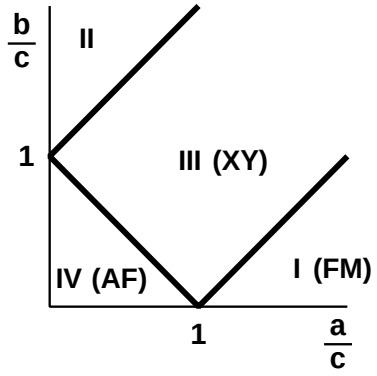

Figure 4: Exact phase diagram of the classical six-vertex model [31] at $h = 0$. The weights a, b, c are defined in eq. (6). Phase III is massless (infinite correlation length). At $\frac{a}{c} + \frac{b}{c} = 1$ there is a Kosterlitz-Thouless phase transition [33] to the massive (finite correlation length) phase IV, and at $\frac{a}{c} - \frac{b}{c} = 1$ there is a first order KDP phase transition [34] to the ferroelectric phase I. The parameter regions of the Heisenberg model are denoted in brackets: $J_z \geq |J_x|$ (AF), $J_z \leq -|J_x|$ (FM), and $|J_z| < |J_x|$ (XY-like). The weights corresponding to the XY-model (or free fermions), i.e. $J_z = 0$, are located on the circle $a^2 + b^2 = c^2$. When $\Delta\tau \to 0$, the point $(a/c = 1, b/c = 0)$ is approached in all cases.

case without magnetic field, in which $J_x \neq J_y$ in the Hamiltonian

$$\hat{H} \;\; = \sum_{\langle \mathbf{ij} \rangle} J_x \hat{S}^x_{\mathbf{i}} \hat{S}^x_{\mathbf{j}} \; + \; J_y \hat{S}^y_{\mathbf{i}} \hat{S}^y_{\mathbf{j}} \; + \; J_z \hat{S}^z_{\mathbf{i}} \hat{S}^z_{\mathbf{j}} \; . \tag{8}$$

We also get this case if we quantize the XXZ-model along an axis different from the z-axis. The treatment is the same as for the XXZ-model. Again we use $\hat{S}^z$ eigenstates to insert complete sets, and arrive at the following nonvanishing matrix elements on the (1+1)-dimensional checkerboard lattice,

$$\begin{aligned}
W(1^{\pm}) \equiv a \;\; &:= \;\; \langle ++|e^{-\hat{\mathcal{H}}}|++\rangle \;=\; \langle --|e^{-\hat{\mathcal{H}}}|--\rangle \;=\; e^{-\frac{\Delta\tau}{4}J_z} \quad \mathrm{ch}(\tfrac{\Delta\tau}{4}(J_x - J_y)) \; , \\
W(2^{\pm}) \equiv c \;\; &:= \;\; \langle +-|e^{-\hat{\mathcal{H}}}|+-\rangle \;=\; \langle -+|e^{-\hat{\mathcal{H}}}|-+\rangle \;=\; e^{+\frac{\Delta\tau}{4}J_z} \quad \mathrm{ch}(\tfrac{\Delta\tau}{4}(J_x + J_y)) \; , \\
W(3^{\pm}) \equiv b \;\; &:= \;\; \langle +-|e^{-\hat{\mathcal{H}}}|-+\rangle \;=\; \langle -+|e^{-\hat{\mathcal{H}}}|+-\rangle \;=\; e^{+\frac{\Delta\tau}{4}J_z} \quad \mathrm{sh}(-\tfrac{\Delta\tau}{4}(J_x + J_y)) \; , \\
W(4^{\pm}) \equiv d \;\; &:= \;\; \langle ++|e^{-\hat{\mathcal{H}}}|--\rangle \;=\; \langle --|e^{-\hat{\mathcal{H}}}|++\rangle \;=\; e^{-\frac{\Delta\tau}{4}J_z} \quad \mathrm{sh}(-\tfrac{\Delta\tau}{4}(J_x - J_y)) \; ,
\end{aligned} \tag{9}$$

which reduce [5] to eq. (6) when $J_x = J_y$. We see that now there is an additional type of vertex with weight d, shown as type $4^{\pm}$ in fig. 2, in which all four arrows point either towards or away from the center. This vertex type may be thought of as a source (resp. sink) of arrows. Eq. (7) becomes the

$$\text{condition} \quad \text{“}\mathbf{divergence = zero \; mod \; 4}\text{”} \quad \text{for the arrows.} \tag{10}$$

The vertices and their weights now correspond to the eight-vertex model [31]. We will see that very little changes for the loop algorithm in this case [3].

[5] On bipartite lattices. See footnote 4.

2.2 Outline of the Loop Algorithm

The traditional way to perform Monte Carlo updates on a worldline configuration [5] consists of proposing *local deformations* of worldlines and accepting/rejecting them with suitable probability. In contrast, the updates for the loop algorithm are very nonlocal. We will first describe the basic idea for the example of the XXZ case and outline the resulting procedure. We postpone the formal discussion and the calculation of Monte Carlo probabilities to the next section.[6]

Two observations lead to the loop algorithm:

(1) The Hamiltonian acts locally on individual plaquettes. Thus the detailed balance condition on Monte Carlo probabilities can be satisfied locally on those plaquettes.

(2) The allowed configurations of arrows in the six-vertex-case have zero divergence, eq. (7). Therefore any two allowed configurations can be mapped into each other by changing the arrow-direction on a set of closed **loops**, where *along each of these loops, the arrows have to point in constant direction*. These are the loops that are constructed in our algorithm. The path of the loops will be determined locally on each plaquette (see below).

An example is given in fig. 5. Note that the loops are not worldlines; instead they consist of the locations of (proposed) *changes* in the worldline occupation number (=arrow directions = spin directions). Also, the loops are not prescribed, instead they will be determined stochastically, with probabilities that depend on the current worldline configuration. Since both the zero divergence condition and detailed balance can be satisfied locally at the plaquettes, we will be able to construct the loops by *local* stochastic decisions on the plaquettes, yet achieve potentially very nonlocal worldline updates.

Let us construct a loop (see fig. 5). This is most easily done using the vertex picture, where the loop follows the arrows of the spin-configuration. We start at some arbitrary site (il) of the plaquette lattice and follow the arrow of the current spin configuration into the next plaquette. There we have to specify the direction in which we will continue. For any allowed spin configuration (see fig. 2) there are two possibilities to continue along an arrow. We choose one of these directions and follow the arrow into the next plaquette. Then we continue to choose a direction at each of the plaquettes which we traverse. If we reach a plaquette a second time, there is only one direction left to go, since the loop should not overlap itself. Eventually we will (on a finite lattice) reach the starting point again, thus closing the loop. If we flip all

[6] *Notation:* From now on we will synonymously use "plaquette" or "vertex" to refer to the shaded plaquettes of the checkerboard lattice. We also use interchangeably the terms "spin direction", "arrow direction", and "occupation number" to refer to the 2 possible states S^z_{il} at each site (il) of the checkerboard lattice. We denote both probabilites and plaquettes by the letter p. S_p and W_p are the spin configuration at plaquette p and its weight, and G_p will be a breakup at p. "Six-vertex-case" (="XXZ-case") and "eight-vertex-case" will refer to the local plaquette constraints (i.e. nonzero weights), *not* necessarily to the respective models of statistical mechanics themselves.

the arrows (=spins) along the loop, we maintain the continuity of arrows (worldlines) at each plaquette, and thus reach a new valid spin configuration.

We can now start to construct another loop (not to overlap the first one), starting at some other arbitrary site which has not been traversed by the first loop. Note that by deciding the directions in which the first loop travelled, we have at those plaquettes also already determined the direction in which a second loop entering the same plaquettes will move, namely along the two remaining arrows. Thus, what we actually decide at each plaquette through which a loop travels, is a **"breakup"** of the current arrow configuration into two disconnected parts, denoting the paths that the two loop segments on this plaquette take. The possible breakups of this kind are shown and described in fig. 3. For each of the six possible arrow configurations $i^{\pm}$, fig. 2, there are two breakups which are compatible with the constraint that the arrows along the loop have to point in a constant direction; namely those breakups labelled $G^{ij}, (i \neq j)$ in fig. 3.

There is another kind of "breakup" that maintains condition (7). Here all 4 spins on a plaquette are forced to be flipped together. We call this choice "freezing" (labeled G^{ii} in fig. 3), since for a flip-symmetric model like the six-vertex model at $h = 0$ it preserves the current weight $W(i^{\pm})$. Freezing G^{ii} can also be viewed as consisting of either one of the two breakups G^{ij} $(j \neq i)$, with the two loop segments on this plaquette being "glued" together. In this view freezing causes sets of loops to be glued together. We shall call such a set (often a single loop) a "**cluster**". All loops in a cluster have to be flipped together.

Overall, we see that *by specifying a breakup for every shaded plaquette, the whole vertex configuration is subdivided into a set of clusters which consist of closed loops.* Each site of the checkerboard lattice is part of one such loop. We shall call such a division of the vertex configuration into loops a "**graph**" G. Flipping the direction of arrows (= spin or occupancy) on all sites of one or more clusters of a graph (a "cluster flip" which consists of "loop flips"), leads to a new allowed configuration. In the loop algorithm the loops are constructed by specifying breakups with suitable probabilities that depend on the current configuration (see below). In the vertex picture, the graph G resides on the same arrows as the spins. In the worldline picture, the elements of G look slightly differently, as seen in figures 5 and 3 . Note that by introducing loops, we have effectively extended the space of variables, from spins, to spins and breakups. This point will be formalized in the next section.

In summary, the *basic procedure* for one Monte Carlo update then consists of two *stochastic* mappings: First from spins to spins plus loops, and then to new spins. I.e., starting with the current configuration of worldlines:

(1) **Select a breakup** (i.e. specify in which direction the loops will travel) for each shaded plaquette with a probability that depends on the current spin configuration at that plaquette. These probabilities are discussed below. Identify the clusters which are implicitely constructed by these breakups. This involves a search through the lattice.

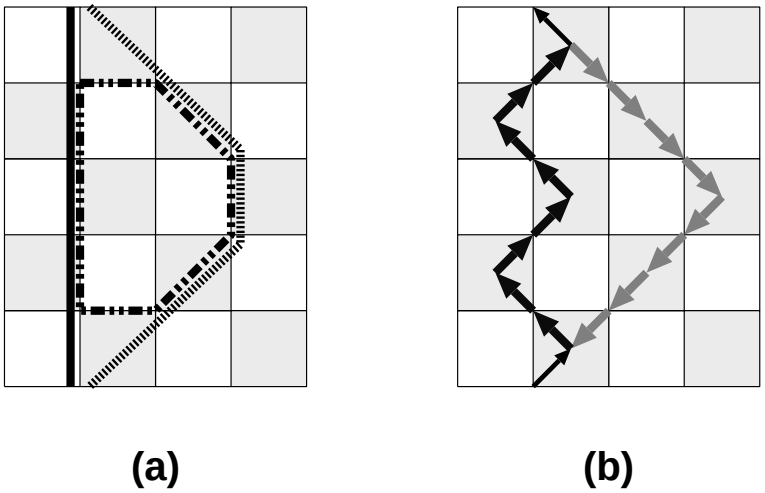

Figure 5: Example of a loop update. (a) Worldline picture. The thick solid line denotes a single worldline, the dot-dashed line depicts a possible loop. By "flipping the loop", i.e. flipping the spin direction (arrow direction) on all sites along the loop, the worldline will be deformed into the dashed shape. Since the loop can potentially be very large, this deformation can be very nonlocal. (b) The same situation as a vertex picture. The loop is represented by the thick arrows. By construction, in the vertex picture each loop follows arrows of the spin-configuration. This means that the loop runs upwards in time-direction along sites with spin $S^z_{il} = +1$, i.e. along worldlines, and downwards in time-direction where there is no worldline.

(2) **Flip each cluster with suitable probability**, where "flipping a cluster" means to change the direction of all arrows along the loops in this cluster (or, equivalently, changing spin direction or occupation number, respectively). The combined cluster flips result in a new spin configuration. The flip probabilities depend in general on the Hamiltonian and on the current spin configuration. In the ideal case, for example the isotropic Heisenberg model in any dimension, each individual loop can be flipped independently with probability 1/2.

An example is given in fig. 6. Notice that in this example the flip of a loop which happened to wind around the lattice in spatial direction led to a change in spatial winding number of the worldline configuration, i.e. an update that *cannot* be done by local deformations of worldlines.

Little changes in the general XYZ-like (eight-vertex-like) case [3]. The loops now have to change direction [3] at every breakup of type $(i, 4)$. Alternatively, one can also omit assigning a direction to loops.

Let us now cast the general ideas into a valid procedure. Sections 2.3 to 2.6 are formal and comprehensive, with detailed explanations. A summary is given in section 2.6, and explicit weights for XXZ and XYZ cases in section 2.7. A recipe for the most important (yet particularly simple) case, the Heisenberg antiferromagnet, is provided in section 2.8. Previous formal expositions can be found in the original loop algorithm papers [2, 3] (the best formal description there is that for the eight-

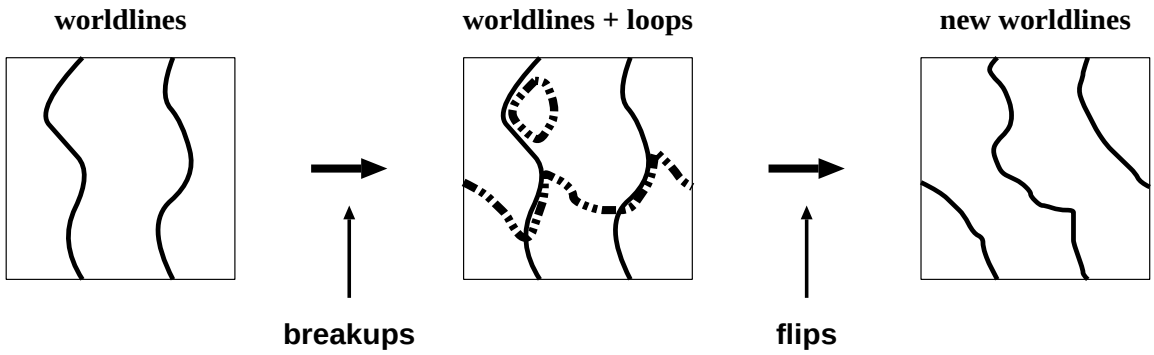

Figure 6: Example of a worldline update with the loop algorithm. For clarity, we show a situation with only two worldlines. We start with the worldline configuration $\mathcal{S}$ in the left picture. The stochastic breakup decision on each plaquette, with probabilities depending on this worldline configuration, defines loops (in general clusters of loops), only two of which are drawn. In this example we then flip both loops, i.e. flip the spin direction (= worldline occupancy) along the loops. This results in the new worldline configuration $\mathcal{S}'$ in the right picture, which, as in this example, can be very different from the original one. Since one of the loops happened to wind around the lattice in spatial direction, its flip produced a worldline configuration with nonzero winding number. Note that for the next worldline update, the current loop configuration is discarded, and a completely new set of breakups will be determined with probabilities depending on the new worldline configuration.

vertex case in ref. [3]), as well as, in a general setting and in a more suitable language closer to the Fortuin-Kasteleyn mapping of statistical mechanics, in the papers by Kawashima and Gubernatis [6, 14]. We shall use both the worldline picture and the vertex picture of ref. [2, 3], in order to provide a bridge between the existing formulations and to make the simple geometry of the problem as obvious as possible. In section 2.10 we point out that for many models it is possible to sum over all spin variables to obtain a pure loop model. Finally, we introduce improved estimators, the single cluster version, and describe the performance of the loop algorithm.

2.3 Kandel-Domany framework

A brief overview of the basics of Monte Carlo algorithms is given in Appendix A. The derivation [2, 3] of the loop algorithm is similar to that for the Swendsen Wang cluster algorithm in statistical mechanics [7] which uses the Fortuin-Kasteleyn mapping of the Ising model to an extended phase space. (For an excellent review see e.g. ref. [35]). A general formalism for such a mapping was given by Kandel and Domany [36]. Here we use a more suitable language similar to that of the general framework of Kawashima and Gubernatis [6], who made the Fortuin-Kasteleyn-like nature of the mapping obvious.

For future use, we first write down the general scheme, without yet making refer-

ence to individual spins, loops, or plaquettes. We start with a set $\{\mathcal{S}\}$ of configurations $\mathcal{S}$ and a set $\{G\}$ of graphs G, which together constitute the extended phase space. The partition function

$$Z = \sum_{\mathcal{S}} W(\mathcal{S}) \tag{11}$$

depends only on $\mathcal{S}$. In addition we now *choose* a new weight function $W(\mathcal{S}, G)$ which must satisfy

$$\begin{array}{lcl} \sum_G W(\mathcal{S}, G) & = & W(\mathcal{S})\,, \\ W(\mathcal{S}, G) & \geq & 0\,. \end{array} \tag{12}$$

Thus we have a Fortuin-Kasteleyn-like [37] representation:

$$Z = \sum_{\mathcal{S}} \sum_{G} W(\mathcal{S}, G) \tag{13}$$

A Monte Carlo update now consists of 2 steps:

i) Given a configuration $\mathcal{S}$ (which implies $W(\mathcal{S}) \neq 0$), choose a graph G with probability

$$p(\,\mathcal{S} \to (\mathcal{S}, G)\,) \;=\; \frac{W(\mathcal{S}, G)}{W(\mathcal{S})}\;. \tag{14}$$

ii) Given $\mathcal{S}$ and G (this implies $W(\mathcal{S}, G) \neq 0$), choose a new configuration $(\mathcal{S}', G')$ with a probability $p\,(\,(\mathcal{S}, G) \to (\mathcal{S}', G')\,)$ that satisfies detailed balance with respect to $W(\mathcal{S}, G)$:

$$W(\mathcal{S}, G) \times p\,(\,(\mathcal{S}, G) \to (\mathcal{S}', G')\,) \;=\; W(\mathcal{S}', G') \times p\,(\,(\mathcal{S}', G') \to (\mathcal{S}, G)\,)\;, \tag{15}$$

for example the heat-bath-like probability

$$p\,(\,(\mathcal{S}, G) \to (\mathcal{S}', G')\,) \;=\; \frac{W(\mathcal{S}', G')}{W(\mathcal{S}, G) + W(\mathcal{S}', G') + const}\;. \tag{16}$$

Then the mapping $\mathcal{S} \to \mathcal{S}'$ also satisfies detailed balance with respect to the original weight $W(\mathcal{S})$. Proof:

$$\begin{array}{rcll} W(\mathcal{S})\, p(\mathcal{S} \to \mathcal{S}') & = & W(\mathcal{S}) & \sum_{G,G'}\; p(\,\mathcal{S} \to (\mathcal{S}, G)\,)\;\; p\,(\,(\mathcal{S}, G) \to (\mathcal{S}', G')\,) \\ & = & W(\mathcal{S}) & \sum_{G,G'}\; \frac{W(\mathcal{S},G)}{W(\mathcal{S})}\;\; p\,(\,(\mathcal{S}', G') \to (\mathcal{S}, G)\,)\;\; \frac{W(\mathcal{S}',G')}{W(\mathcal{S},G)} \\ & = & W(\mathcal{S}') & \sum_{G,G'}\; \frac{W(\mathcal{S}',G')}{W(\mathcal{S}')}\;\; p\,(\,(\mathcal{S}', G') \to (\mathcal{S}, G)\,) \\ & = & W(\mathcal{S}') & \sum_{G,G'}\; p(\,\mathcal{S}' \to (\mathcal{S}', G')\,)\;\; p\,(\,(\mathcal{S}', G') \to (\mathcal{S}, G)\,) \\ & = & W(\mathcal{S}') & p(\mathcal{S}' \to \mathcal{S})\;. \end{array} \tag{17}$$

(Within a Monte Carlo simulation the denominators in eq. (17) cannot vanish.)

2.4 Exact mapping of plaquette models

We apply the Kandel-Domany formalism to a model defined on plaquettes, with

$$\begin{array}{rcl} W(\mathcal{S}) & = & A_{global}(\mathcal{S}) \times W^{plaq}(\mathcal{S}) \,, \\ & = & A_{global}(\mathcal{S}) \times \prod_p W_p(S_p) \,. \end{array} \tag{18}$$

To cover the general case, we have split off a global weight factor A_{global}. This split is not unique. We devise an algorithm for $W^{plaq}(\mathcal{S}) \equiv \prod_p W_p(S_p)$. Because of its product structure, we can perform the decomposition into graphs separately on every plaquette. Thus in analogy with eq. (12) we look for a set of "breakups" G_p and new weights $W_p(S_p, G_p)$ on every plaquette p which satisfy

$$\begin{array}{rcl} \sum_{G_p} W_p(S_p, G_p) & = & W_p(S_p) \,, \\ W_p(S_p, G_p) & \geq & 0 \,. \end{array} \tag{19}$$

This implies eq. (12) again, both for the plaquette part

$$W^{plaq}(\mathcal{S}) = \prod_p \sum_{G_p} W_p(S_p, G_p) = \sum_{\bigcup_p G_p} \prod_p W_p(S_p, G_p) \equiv \sum_G W^{plaq}(\mathcal{S}, G) \tag{20}$$

(where $G \equiv \bigcup_p G_p$, $W^{plaq}(\mathcal{S}, G) \equiv \prod_p W_p(S_p, G_p)$) and for the total weight $W(\mathcal{S}) = \sum_G W(\mathcal{S}, G)$ with

$$\begin{array}{rcl} W(\mathcal{S}, G) & = & A_{global}(\mathcal{S}) \times W^{plaq}(\mathcal{S}, G) \\ & = & A_{global}(\mathcal{S}) \times \prod_p W_p(S_p, G_p) \,. \end{array} \tag{21}$$

Thus we can apply the Kandel-Domany procedure. Restricting ourselves to $G' = G$, the two steps i), ii) in the previous section now become the procedure for the loop algorithm. Starting with a configuration $\mathcal{S}$, a Monte Carlo update consists of:

(1) Breakup: For each plaquette, satisfy eq. (14) by choosing G_p with probability

$$p(\, S_p \to (S_p, G_p) \,) \;=\; \frac{W_p(S_p, G_p)}{W_p(S_p)} \,. \tag{22}$$

(2) Flip: Choose a new configuration $(\mathcal{S}', G)$ with a probability $p\,(\,(\mathcal{S}, G) \to (\mathcal{S}', G)\,)$ that satisfies detailed balance with respect to $W(\mathcal{S}, G)$.

In the next section we shall explicitely find a suitable set of breakups G_p and plaquette weights $W_p(S_p, G_p)$.

2.5 Structure of plaquette weight functions

Given a graph G, we demand that $W^{plaq}(\mathcal{S}, G)$ does not change

$$W^{plaq}(\mathcal{S}, G) = W^{plaq}(\mathcal{S}', G) \tag{23}$$

upon any spin update allowed by $W^{plaq}(\mathcal{S}', G) \neq 0$. Then it cancels in eq. (16), which can now be written as

$$p_{flip}(\mathcal{S} \to \mathcal{S}') = \frac{A_{global}(\mathcal{S}')}{A_{global}(\mathcal{S}) + \sum_{\mathcal{S}' \neq \mathcal{S}, W(\mathcal{S}', G) \neq 0} A_{global}(\mathcal{S}')} \; . \tag{24}$$

The configurations $\mathcal{S}'$ for which $W(\mathcal{S}', G) \neq 0$ will be those reached by cluster flips. By enforcing eq. (23), all cluster flips will become independent of each other, up to acceptance with A_{global}. Eq. (23) is equivalent to

$$W^{plaq}(\mathcal{S}, G) = \Delta(\mathcal{S}, G)\; V(G) \; , \quad \Delta(\mathcal{S}, G) := \begin{cases} 1, \; W^{plaq}(\mathcal{S}, G) \neq 0, \\ 0, \; \text{otherwise} \end{cases} \; , \tag{25}$$

which is the form introduced in ref. [6]. Thus

$$W(S, G) = \Delta(\mathcal{S}, G)\; V(G)\; A_{global}(\mathcal{S}). \tag{26}$$

We shall achieve eq. (23) by enforcing it on every plaquette:

$$W_p(S_p, G_p) = W_p(S'_p, G_p) \, . \tag{27}$$

Then eq. (26) also holds on the plaquette level: $W_p(S_p, G_p) = \Delta(\mathcal{S}_p, G_p) V(G_p)$. The nontrivial part in this point of view is that all allowed plaquette updates $S_p \to S'_p$ match for different plaquettes, to give an overall allowed update $\mathcal{S} \to \mathcal{S}'$. As we have seen in section 2.2, it is the six- (or eight-) vertex constraint, stemming from local conservation of S^z (or $S^z \; mod \; 2$) in the Hamiltonian, that makes these plaquette updates match in the form of loops. In other words, by enforcing eq. (27), we will achieve that all clusters (sets of loops that are glued together at frozen plaquettes) constructed during the breakup-step can be flipped independently, up to acceptance with p_{flip}, eq. (24).

Let us now find weights satisfying eq. (27). Independent cluster flips require that eq. (27) at least include the case $S'_p = \overline{S}_p$, where all four spins at a plaquette are flipped, $W_p(S_p, G_p) = W_p(\overline{S}_p, G_p)$, which implies the requirement

$$W_p(S_p) \equiv \sum_{G_p} W_p(S_p, G_p) = \sum_{G_p} W_p(\overline{S}_p, G_p) = W_p(\overline{S}_p) \tag{28}$$

on the plaquette weights $W_p(S_p)$. The first step in our construction is therefore to

$$\textit{Choose} \;\; A_{global} \;\; \textit{such that} \;\; W_p(S_p) = W_p(\overline{S}_p) \, . \tag{29}$$

Such an A_{global} always exists. It is not unique. The ideal case is $A_{global} = const$, since then for each cluster, $p_{flip} \equiv \frac{1}{2}$ can be chosen. See also section 3.4.

For worldline models, there are a total of eight allowed spin configurations $S_p = i^{\pm} = 1^{\pm}, 2^{\pm}, 3^{\pm}, 4^{\pm}$, as shown in fig. 2. With eq. (28), the plaquette weight $W_p(S_p)$ depends only on i. Following ref. [3], let us

$$\text{Define a different “}\textbf{breakup}\text{”} \;\; G_p := G^{ij} \equiv G^{ji} \text{ for every transition } \; i \leftrightarrow j \; , \tag{30}$$

such that the breakup G^{ij} allows exactly the transitions $i \leftrightarrow j$. Thus we define

$$W_p(S_p, G^{ij}) := \begin{cases} w^{ij}, & \text{if } S_p = i^{\pm} \text{ or } S_p = j^{\pm} \ , \\ 0 & \text{otherwise} \end{cases} \tag{31}$$

with suitable constants $w^{ij} \equiv w^{ji}$. We have satisfied eq. (27) by construction. By inspection of fig. 2 we see that every transition $i \leftrightarrow j$, $i \neq j$, corresponds to the flip of 2 spins on a plaquette (all four spins for $i^+ \leftrightarrow i^-$).

We also see by inspection of fig. 3 that, *given* the current worldline configuration $S_p = i^{\pm}$, we can identify each of the 4 breakups G^{ij}, $j = 1, 2, 3, 4$, with one of the *graphs* in fig. 3. Namely, flipping 2 of the spins connected in the graph for G^{ij}, $i \neq j$, leads to one of the two plaquette configurations $j^{\pm}$, flipping the other two spins leads to the other configuration, flipping all four spins maps from $i^{\pm}$ to $i^{\mp}$. Therefore, *given* a worldline configuration, the combined breakup $G = \bigcup_p G_p$ can be represented[7] as a graph consisting of the plaquette-elements in fig. 3. In many cases we can also transform the worldline model entirely into a loop graph model; see section 2.10.

Since G_p connects pairs of sites, the breakups of all plaquettes will combine to give a set of clusters consisting of loops, as already described in section 2.2. When there is no freezing, i.e. no breakups G^{ii} occur, then all clusters consist of single loops.

We still need to find constants $w^{ij} \equiv w^{ji}$, $i, j = 1, 2, 3, 4$, such that the constraint eq. (19) is satisfied, which now reads [8]

$$\begin{aligned} \textstyle\sum_j w^{ij} &= W(i) \, , \\ w^{ij} &\geq 0 \, , \end{aligned} \tag{32}$$

(with $W(4) = 0$ in the six-vertex-case). This constraint underdetermines the w^{ij}. There are 3 equations for 6 unknowns in the six-vertex case, and 4 equations for 10 unknowns in the eight-vertex case. It can always be solved. One explicit solution is the following: Let $W(k)$ be the smallest of the n weights $W(j)$, $j = 1, .., n$ (n is 3 or 4). Eq. (32) is satisfied by

$$\begin{aligned} w^{ij} &= W(k)/n && \text{for } i \neq j \ , \\ w^{ii} &= W(i) - \textstyle\sum_{j \neq i} w^{ij} && \text{for } i = 1, .., n \ . \end{aligned} \tag{33}$$

Experience tells us that for an efficient algorithm, one should keep the loops as independent as possible. Thus we should minimize the weights w^{ii} which cause loops to be glued together. Let $W(l)$ be the largest of the n weights $W(j)$. Given a solution w^{ij} we can always find another one in which no diagonal element w^{ii} except at most

[7] In general we should distinguish between the breakups G^{ij}, of which there are 6 (10) in the six (eight) -vertex case, and the fewer (4) graphical representation in fig. 3. Kawashima has shown that one can also give a common graphical representation of G^{ij} for all (ij) [14]. This representation requires more than one loop-element per site.

[8] Eqs. (31),(32) are eqs. (15),(16) in ref. [3].

w^{ll} is nonzero [14]. For example, to remove $w^{jj}, j > 1$, define

$$\begin{array}{lll} w'^{j,j} & = 0 & \\ w'^{j,j-1} = w'^{j-1,j} & = w^{j-1,j} & + \quad w^{jj} \\ w'^{j-1,j-1} & = w^{j-1,j-1} & - \quad w^{jj} \; . \end{array} \tag{34}$$

Iterating this transformation leads to the one surviving diagonal element

$$w'^{ll} = w^{ll} - \sum_{i \neq k} w^{ii} \; . \tag{35}$$

More explicit solutions are given in section 2.7.

2.6 Summary of the loop algorithm

Since the detailed derivation of the general formalism was a bit tedious, we summarize the actual procedure here. Start with a model in worldline representation with $Z = \sum_{\mathcal{S}} W(\mathcal{S})$, eq. (11).

(1) Choose a split $W(\mathcal{S}) = A_{global}(\mathcal{S}) \times \prod_p W_p(S_p)$, eq. (18), such that $W_p(S_p) = W_p(\overline{S}_p)$, eq. (29).

(2) Find new weights $w^{ij} = w^{ji} \geq 0$ such that $\sum_j w^{ij} = W(j)$, eq. (32), while preferrably minimizing the "freezing" weights w^{ii}, see eq. (34).

Each Monte Carlo update from a worldline configuration to a new one then involves the following steps:

(i) (Breakup) For each shaded plaquettes, with current spin configuration $i^{\pm}$, choose a breakup G^{ij} with probability $p = w^{ij}/W(i)$, proportional to the graph weight w^{ij}. (See eqs. (22,31) and section 2.7).

(ii) (Cluster identification) All plaquette breakups together subdivide the vertex lattice into a set of clusters, which consist of closed loops. Loops which have a frozen vertex ("G^{ii}") in common belong to the same cluster. Identify which sites belong to which clusters. (This is in general the most time consuming task).

(iii) (Flip) Flip each cluster separately[9], one after the other, with (e.g.) heat-bath probability for A_{global}. In case $A_{global} \equiv 1$, this means that one can flip each cluster independently with probability $\frac{1}{2}$. "Flipping" means to change the sign of S^z_{il} on all sites in the cluster. If desired, one can artificially restrict the simulation to some sector of phase space, e.g. to the **"canonical ensemble"** of constant magnetization, by prohibiting updates that leave this sector.

[9] Alternatively one can perform a combined flip of a randomly chosen subset of clusters. However, when A_{global} is not unity, this will in general produce bigger variations of A_{global} and therefore lower acceptance rates.

2.7 Graph weights for the XXZ, XYZ, and Heisenberg model

We now come back to our example and compute [2, 3] one solution for the weights $w^{ij} \equiv w^{ji}$, and thus the breakup and flip probabilities, for the spin-flip symmetric six-vertex case, with weights a,b,c, eq. (6). This includes the Heisenberg model and the XXZ-model at $h = 0$ (eq. (6)) in any dimension (see section 3.1). Some solutions for the general XYZ model are also given. We need to find a solution to eq. (32). Here it reads[10]

$$\begin{array}{lllll} w^{11} + w^{12} + w^{13} & = W(1) & \equiv & a & \simeq & 1 - \frac{\Delta\tau}{2} J_z \\ w^{22} + w^{12} + w^{23} & = W(2) & \equiv & c & \simeq & 1 \\ w^{33} + w^{13} + w^{23} & = W(3) & \equiv & b & \simeq & \frac{\Delta\tau}{2} |J_x| \; . \end{array} \tag{36}$$

From fig. 3 we see that w^{12}, w^{23}, and w^{13} correspond to vertical, horizontal, and diagonal breakups, respectively. The weights w^{ii} correspond to transitions $i^{\pm} \to i^{\pm}$, i.e. to flipping zero or four spins on a plaquette. They freeze the value of the weight $W(i)$. As mentioned above, experience tells us that we should minimize freezing in order to get an efficient algorithm, in which then loops are as independent as possible. We will construct the solution with minimal freezing; others exist.

Eq. (36) has different types of solutions in different regions of the parameter space (a, b, c). Remarkably, these regions are exactly the same [2, 3] as the phases of the two-dimensional classical six-vertex model [34, 31], shown in fig. 4. The regions of fig. 4 have been spelled out in terms of the coupling constants J_x,J_z at the end of section 2.1.

Region IV (AF), has antiferromagnetic couplings $J_z \geq |J_x|$, thus $c \geq a + b$. To minimize the freezing of weight c, we have to minimize w^{22}. From eq. (36), $w^{22} = c - a - b + w^{11} + w^{33} + 2w^{13}$. With $w^{ij} \geq 0$ this implies $w^{22,\min} = c - a - b$. This minimal value of w^{22} is achieved for $w^{11} = w^{33} = 0$, i.e. when we minimize all freezing. The optimized nonzero parameters for region IV are then:

$$\begin{array}{lllll} w^{12} = a & \simeq & 1 - \frac{\Delta\tau}{2} J_z & & \text{(vertical breakup)}, \\ w^{23} = b & \simeq & \frac{\Delta\tau}{2} |J_x| & & \text{(horizontal breakup)}, \\ w^{22} = c - a - b & \simeq & \frac{\Delta\tau}{2} (J_z - |J_x|) & & \text{(freezing of opposite spins)}, \end{array} \tag{37}$$

without any diagonal breakups. (This has to be modified for non-bipartite lattices; see section 2.9).

In region I (FM) with ferromagnetic couplings $J_z \leq -|J_x|$ and $a > b + c$ we get

$$\begin{array}{lllll} w^{12} = c & \simeq & 1 & & \text{(vertical breakup)}, \\ w^{13} = b & \simeq & \frac{\Delta\tau}{2} |J_x| & & \text{(diagonal breakup)}, \\ w^{11} = a - c - b & \simeq & \frac{\Delta\tau}{2} (|J_z| - |J_x|) & & \text{(freezing of equal spins)}, \end{array} \tag{38}$$

[10] We have multiplied the weights in eq. (6) by $\exp(-\Delta\tau J_z/4)$ and also provided the expansion to order $\Delta\tau$ for later use in the continuous time version.

without any horizontal breakups. (This is similar for region II, $b > a + c$, which does not correspond to a quantum model. There we obtain minimal freezing from eq. (37) with indices 2 and 3 interchanged, and no vertical breakups.)

Region III (XY-like) has $|J_z| < |J_x|$, and $a, b, c \leq \frac{1}{2}(a + b + c)$. Here we can set all freezing probabilities to zero, obtaining

$$\begin{array}{rcccll} 2w^{12} & = & a + c - b & \simeq & 2 - \frac{\Delta\tau}{2}\left(|J_x| + J_z\right) & \text{(vertical breakup)}, \\ 2w^{23} & = & c + b - a & \simeq & \frac{\Delta\tau}{2}\left(|J_x| + J_z\right) & \text{(horizontal breakup)}, \\ 2w^{13} & = & b + a - c & \simeq & \frac{\Delta\tau}{2}\left(|J_x| - J_z\right) & \text{(diagonal breakup)}. \end{array} \tag{39}$$

The isotropic Heisenberg model is located on the boundaries of region III. The antiferromagnet $J_z = |J_x|$ has $c - a - b = 0$, thus only vertical (w^{12}) and horizontal (w^{23}) breakups. The ferromagnet $J_z = -|J_x|$ obeys $c - a + b = 0$ and has only vertical (w^{12}) and diagonal (w^{23}) breakups. There is no freezing for the isotropic model.

The classical Ising model is reached in the limit $J_x/J_z = 0$, since then $b = 0$, so that there is no more hopping and all worldlines are straight. Remarkably, in this limit the loop algorithm becomes [32] the Swendsen-Wang cluster algorithm [7] ! Frozen plaquettes connect the sites of clusters in the Ising model, i.e. they correspond to the "freezing" operation [36] of the efficient Swendsen-Wang method.

The classical BCSOS model is simulated for $a = b$ [1]. When $a = b = \frac{1}{2}c$, the loop algorithm constructs [1] the *boundaries* of the clusters which the VMR-cluster algorithm [38, 39] for the (1+1)-dimensional BCSOS model produces, i.e. it constructs these clusters more efficiently. The loop representation was used in ref. [40] to obtain exact analytical results for this model.

General XYZ case: (See also ref. [14] for explicit solutions.) The loop algorithm remains unchanged in the XYZ case (see section 2.2), except that at breakups G^{i4}, the arrows flip direction. In each of the four ordered regions of the XYZ model we have $W(m) \geq \sum_{i \neq m} W(i)$ for one $m \in \{1, 2, 3, 4\}$. The nonzero breakup weights with minimum freezing are then

$$\begin{array}{rcl} w^{mm} & = & W(m) - \sum_{i \neq m} W(i) \\ w^{im} & = & W(i) \;\; (i \neq m). \end{array} \tag{40}$$

This also summarizes the solutions for regions I,II,IV above. In the disordered region $2W(m) < \sum_i W(i)$ we can set all freezing w^{ii} to zero, and in general still have 6 free parameters w^{ij} with only 4 constraints eq. (32).

2.8 Recipe for the spin $\frac{1}{2}$ Heisenberg antiferromagnet

In order to make the loop algorithm as clear as possible, we restate the procedure for the important yet simple case of the isotropic spin $\frac{1}{2}$ Heisenberg antiferromagnet.

A Monte Carlo update leads from a worldline configuration $\mathcal{S}$ of spin variables $S^z_{il} = \pm 1$ to a new configuration $\mathcal{S}'$. On each shaded plaquette p, the local spin

configuration S_p takes one of the six possibilities shown in the left part of fig. 2, with weights $W_p(S_p)$ given in eq. (6), satisfying $a+b=c$ in the isotropic antiferromagnetic case. The weights w^{ij} in eq. (37) are all zero except for $w^{12}=a\,, w^{23}=b$, so that we get only vertical (G^{12}) and horizontal (G^{23}) breakups. The update consists of the following steps:

(i) For each shaded plaquette, choose the horizontal breakup with probability

$$p(S_p,G^{23}) \;=\; (\delta_{S_p,2^\pm}+\delta_{S_p,3^\pm})\;\tfrac{\mathrm{w}^{23}}{\mathrm{W_p(S_p)}} \;=\; \begin{cases} 0, & S_p=1^\pm\,, \\ \tanh(\frac{\Delta\tau}{2}J) & S_p=2^\pm\,, \\ 1, & S_p=3^\pm\,, \end{cases} \tag{41}$$

(see eqs. (6,22,31,37)), otherwise choose the vertical breakup.

(ii) Identify the clusters constructed in step (i). Since there is no freezing here, all clusters consist of single loops.

(iii) Flip each loop with probability $\frac{1}{2}$, where flipping means to change the sign of S^z_{il} on all sites along the loop. This gives the new configuration $\mathcal{S}'$.

This procedure is even simpler than local worldline updates. Moreover, it remains completely unchanged in arbitrary dimensions (see section 3.1). Note that one can and should avoid the Trotter approximation altogether, by working directly in continuous time, for which a modification of this recipe will be given in section 3.7.

2.9 Ergodicity

To establish correctness of the loop algorithm, we still have to show ergodicity for the overall algorithm, including the existence of global configuration changes, Ergodicity is obvious when all $w^{ij}>0$ for $i\neq j$, and when p_{flip} is always nonzero (which is normally the case when we use eq. (24) for p_{flip}). Any two allowed configurations (i.e. $W(\mathcal{S})\neq 0$) are, as always, mapped into each other by a unique set of spin-flips (loop-flips), which are compatible with a set of breakups G^{ij}, $i\neq j$. With $w^{ij}>0$, this set of breakups has a finite probability to occur, and with $p_{flip}>0$, the two configurations will be mapped into each other in a single Monte Carlo step with finite probability. Note that the trivially ergodic case $w^{ij}>0$ can always be constructed, as seen in eq. (33); this may not be an efficient algorithm, however. On the other hand, one can always construct weights w^{ij} such that (for $J_{xy}\neq 0$) ergodicity is not achieved, for example by choosing $w^{ij}=\delta_{ij}W(i)$, i.e. only freezing.

When some of the w^{ij} vanish, ergodicity has to be shown case by case. With the weight choices of section 2.7, region III is trivially ergodic. We have to show ergodicity explicitely in each of the regions I, II, IV, because some w^{ij} vanish there.

Region I (including the Heisenberg FM): $w^{23}=0$, i.e. there are only vertical and diagonal breakups (see fig. 3). These breakups permit a loop configuration which

is identical to any given worldline configuration. That loop configuration will occur with finite probability. Flipping all loops in this configuration leads to the empty worldline configuration. Conversely, any worldline configuration can be generated from the empty one in a single (!) update by such a choice of loops. Therefore the algorithm is ergodic, mapping any two worldline configurations into each other in only two steps.

Region IV (including the Heisenberg AF): $w^{13} = 0$, i.e. there are only vertical and horizontal breakups. On a *bipartite* lattice with open or periodic spatial boundary conditions, ergodicity can be shown easily[41]. Start with any worldline configuration $\mathcal{S} = \{S_{xl}\}$. Our *reference configuration* this time is not the "empty" configuration $S'_{xl} = -1$, but instead the staggered configuration $S'_{xl} = (-1)^x S_{xl}$, i.e. the configuration with straight worldlines on one of the two sublattices. As always, there is a unique set of loops whose flips will map $\mathcal{S}$ into $\mathcal{S}'$. By inspection we see that these loops contain only vertical and horizontal breakups (horizontal where $\mathcal{S}'$ has diagonal worldline parts, vertical elsewhere). Since these breakups have finite probability to occur, the whole set of loops will be constructed with finite probability. Thus, again, any worldline configuration will be mapped to the reference configuration with finite probability, and vice versa, so that on a bipartite lattice the algorithm is ergodic. Furthermore, on any lattice, the loop algorithm is at least as ergodic as the algorithm with the conventional local updates. The latter consist of spin-changes around non-shaded plaquettes, equivalent to the flip of a small loop with two vertical and two horizontal breakups, which will occur with finite probability in the loop algorithm.

Note that for a frustrated antiferromagnet, i.e. on a non-bipartite lattice, the algorithm eq. (37) with only horizontal breakups is not ergodic [21]: Loops switch time-direction at every breakup, thus a loop with an odd number of spatial hops is not possible. To ensure ergodicity, one has to include diagonal breakups with some weight $0 < w^{13} < b$. Then ergodicity is trivial, since all $w^{ij} > 0$ for $i \neq j$. Eq. (36) is now solved with $w^{23} = b - w^{13}$ and demands freezing $w^{22} = 2w^{13}$ of equal spins. The size of w^{13} has to be chosen for optimal performance of the algorithm.

For completeness, we mention region II (which does not occur in worldline models). Here there is no vertical breakup. In case of periodic spatial boundary conditions, interchange of "space" and "time" leads us to the situation of region I, for which we have already shown ergodicity.

2.10 Transformation to a pure loop model

Remarkably, by using the exact mapping eqs. (12,19) on which the loop algorithm is based, we can transform quantum spin and particle models into pure loop representations, i.e. into a completely different setting than the original worldlines. This was first achieved, independently, by Nachtergaele and Aizenman [26] for the one-dimensional Heisenberg antiferromagnet, and was used to prove rigorous correlation inequalities [26]. Kondev and Henley used it to compute the exact stiffness and

critical exponents of a twodimensional vertex model [40, 42].

We get to a loop model [41] by explicitely summing over the spin degrees of freedom in eq. (13). Using eqs. (13,21,25) we see that

$$Z = \sum_{\{\mathcal{S}\}} W(\mathcal{S}) \;\; = \;\; \sum_{\{\mathcal{S}\}} \prod_p \sum_{G_p} \Delta(S_p, G_p)\; V(G_p)\; A_{global}(\mathcal{S})\;. \tag{42}$$

The condition $\Delta(S_p, G_p) \neq 0$ restricts the graph G to consist of clusters, i.e. divergence-free components compatible with the spin configuration.

The summation over spin configurations $\mathcal{S}$ can easily be done if a *reference spin configuration* $\mathcal{S}_0$ exists (see also section 2.9), in which all plaquette breakups G_p with nonvanishing weight $V(G_p)$ are allowed (i.e. $W_p(S_{p0}, G_p) > 0$ whenever $V(G_p) > 0$). Then *all* graphs with finite weight can be constructed from S_0. By design, cluster flips do not change $W(S, G)$ when $A_{global}(S)$ is constant. Each cluster can then be flipped independently and contributes a factor 2. With the weight choices from section 2.7, we see that for the AF region IV, the antiferromagnetically staggered configuration is such a reference configuration on bipartite lattices: it allows all relevant breakups (vertical, horizontal, freezing of unequal spins) on any plaquette. For region I (FM), the ferromagnetic spin configuration serves the same purpose on any lattice.

In these regions (as well as in region II), we can then sum over spin configurations in eq. (42). Without external weight A_{global}, we get

$$Z = \sum_{G=\bigcup_p G^{ij}} \left(\prod_p w^{ij} \right) 2^{N_c(G)} \;\; \equiv \;\; \sum_G W(G)\;, \tag{43}$$

where $N_c(G)$ is the number of clusters in G. When A_{global} is a product over contributions from each cluster (e.g. in case of a magnetic field), the factor 2 for each cluster is replaced by $(A_{global}(\mathcal{S}(G)) + A_{global}(\mathcal{S}'(G)))$.

When there is *no* reference configuration, e.g. for region III (XY-like) or for antiferromagnets on non-bipartite lattices, or for different choices of breakup-weights, we can still obtain a pure loop model. Now there can be clusters which do not correspond to a continuous worldline configuration (i.e. the spin directions mandated by independently chosen breakups at different plaquettes within a cluster may contradict each other). To remove graphs containing such clusters, we temporarily endow each loop with a direction, and introduce a constraint into the sum over graphs in eq. (43) enforcing compatibility of the loop directions of each cluster.

The mapping generalizes immediately to the anisotropic XYZ-like case, where the number of a priori breakup-possibilities per plaquette doubles, though they are graphically still the same as in the XXZ-like case (see fig. 3).

We have thus mapped all XYZ-like quantum spin and particle models, with any choice of breakup weights and in any dimension, to a *loop* model, in complete analogy with the Fortuin-Kasteleyn mapping [37] of statistical mechanics. This mapping is useful for analytical purposes (see above). Note that for the Heisenberg antiferromagnet, the loop model consists of antiferromagnetic selfavoiding polygons, and for the

Heisenberg ferromagnet it has a similar graphical representations as the worldlines themselves. Note also that for a given physical model there are many different loop models, corresponding to the different possible choices of breakup-weights. Remarkably, the graph-decomposition eq. (19), and thus the transformation to a loop model, can also be written on an operator level [9] (see section 3.8). All observables can be measured in the loop representation, as correlation functions (see sections 2.11, 3.8) or as thermodynamic derivatives.

One can perform Monte Carlo simulations purely in the loop representation, analoguously to Sweeny's method for the cluster representation of the Ising model [43]. This has not yet been tried (but the method of section 3.10 comes close). In more than one spatial dimension, it is computationally more difficult than the loop algorithm with graphs and spins, since one has to keep track of the number of clusters, which in general requires traversing complete clusters for each local update.

2.11 Improved Estimators

In addition to the reduction of autocorrelations, the combined representation eq. (13) allows a potentially drastic reduction of statistical errors by using so-called improved estimators [44, 45, 25, 21, 43]. The Monte Carlo procedure provides us with a series of configurations $\mathcal{S}_i$. For each such configuration, we construct a graph, with some n_i clusters. We can then reach any state in a set $\mathcal{F}_i$ of 2^{n_i} worldline configurations by flipping a subset of the clusters. The probability for each of these configurations is determined by the cluster flip probabilities p_{flip}. In the loop algorithm one configuration $\mathcal{S}_{i+1}$ will be chosen randomly according to these probabilities as the next Monte Carlo configuration. The standard thermal expectation value of an observable $\mathcal{O}$ is calculated by averaging over the value of the observable in the configurations $\mathcal{S}_i$:

$$\langle \mathcal{O} \rangle = \sum_i \mathcal{O}(\mathcal{S}_i). \tag{44}$$

An improved estimator with $\langle \mathcal{O} \rangle = \langle \mathcal{O}_{impr} \rangle$ can formally be defined as a weighted average [41] of $\mathcal{O}$ over the states $\mathcal{S}'$ that can be reached from $\mathcal{S}$ with any valid Monte Carlo procedure. With the loop algorithm we sum over the 2^{n_i} states $\mathcal{S}' \in \mathcal{F}_i$. Since this is a sum over many states, it has reduced variance. Ideally, it can be calculated completely and then depends only on the graph G

$$\mathcal{O}_{impr}(G) \;=\; \frac{\sum_{\mathcal{S}'} W(\mathcal{S}', G)\mathcal{O}(\mathcal{S}')}{\sum_{\mathcal{S}'} W(\mathcal{S}', G)} \;=\; \frac{\sum_{\mathcal{S}' \in \mathcal{F}_i} A_{global}(\mathcal{S}')\mathcal{O}(\mathcal{S}')}{\sum_{\mathcal{S}' \in \mathcal{F}_i} A_{global}(\mathcal{S}')}, \tag{45}$$

where we have used eq. (26). Note that $\mathcal{O}_{impr}(G)$ is the representation of $\hat{O}$ in the pure loop model (see sections 2.10, 3.8). Alternatively, the sum in eq. (45) can be evaluated stochastically,

$$\mathcal{O}_{impr}(\mathcal{S}, G) \;=\; \sum_{\mathcal{S}' \in \mathcal{F}_i} p_G(\mathcal{S} \to \mathcal{S}')\; \mathcal{O}(\mathcal{S}')\;, \tag{46}$$

where $p_G(\mathcal{S} \to \mathcal{S}')$ is any probability that satisfies detailed balance for A_{global} (and thus for $W(S,G)$); it need not be equal to the actual update probability. Usually it is a product over suitably chosen cluster flip probabilities p_{flip}. We need to calculate this average over 2^{n_i} states in a time comparable to the time needed for a single measurement. Fortunately that is often possible.

Especially simple improved estimators can be found in the case that $p_{\rm flip} = \frac{1}{2}$ for all clusters. Then eq. (46) simplifies to

$$\mathcal{O}_{impr} \;=\; 2^{-n_i} \sum_{\mathcal{S}' \in \mathcal{F}_i} \mathcal{O}(\mathcal{S}'), \tag{47}$$

since all of the states in $\mathcal{F}_i$ now have the same probability 2^{-n_i}. In order to achieve $p_{flip} = \frac{1}{2}$ when there is a nontrivial global weight A_{global}, we can use a probability function $p_G(\mathcal{S} \to \mathcal{S}')$ that has some clusters fixed in a certain state, and then has a new flip probability of $\frac{1}{2}$ for all other clusters. There are many possibilities. One can for example fix a cluster in its present state with Metropolis probability, or one can [21] fix the state of a cluster with probability $p_{\rm fix} = |2p_{\rm flip} - 1|$, in its present state if $p_{\rm flip} < \frac{1}{2}$, and in its flipped stated otherwise. The improved estimators then contain the usual contributions from both states of the non-fixed clusters, as well as contributions from the constant state of the fixed clusters [21]. In the following we assume for simplicity that no clusters are fixed.

Let us calculate some useful improved estimators. Consider as an example the spin correlation function $\langle \hat{S}^z_j \hat{S}^z_k \rangle$ between two spins at spacetime sites $j = (\mathbf{r}, \tau)$ and $k = (\mathbf{r}', \tau')$. Since each spin can be in one cluster only, the improved estimator eq. (47) is

$$4(\hat{S}^z_j \hat{S}^z_k)_{impr} = \begin{cases} \sigma_j \sigma_k, & \text{if } j \text{ and } k \text{ are in the same cluster,} \\ (1 - 2p_{\rm flip,j})\,(1 - 2p_{\rm flip,k})\;\sigma_j \sigma_k, & \text{otherwise,} \end{cases} \tag{48}$$

where $\sigma_{j,k} = \pm 1$ are the current values of the worldline variables. In case $p_{\rm flip} = \frac{1}{2}$, the improved estimator is extremely simple:

$$4(\hat{S}^z_j \hat{S}^z_k)_{impr} = \begin{cases} \sigma_j \sigma_k, & \text{if sites } j \text{ and } k \text{ are in the same cluster,} \\ 0, & \text{otherwise.} \end{cases} \tag{49}$$

We see that the calculation of improved estimators of correlation functions requires even *less* effort than for non-improved estimators. Remarkably, the spin-spin correlation function corresponds to the size distribution of the clusters. In general one can compute n-point Greens functions as improved estimators from the geometric properties of the loop clusters (section 3.8). Note that for the Heisenberg FM at momentum $q = 0$ and for the Heisenberg AF at momentum $q = \pi$, we always have $\sigma_j \sigma_k = 1$ in eq. (49).

The potential gain from using the improved estimator is easy to see when $\sigma_j \sigma_k = 1$. The expectation value is the same as for the unimproved estimator $\sigma_j \sigma_k = \pm 1$. When

$\langle \mathcal{O} \rangle$ is small (e.g. $\langle \mathcal{O} \rangle \sim \exp(-R/\xi)$ at large distance $R = |\mathbf{r} - \mathbf{r}'|$), then the variance of $\mathcal{O}$ is

$$\langle \mathcal{O}^2 \rangle - \langle \mathcal{O} \rangle^2 \; = \; 1 - \langle \mathcal{O} \rangle^2 \; \approx \; 1 \; , \tag{50}$$

whereas the variance of $\mathcal{O}_{impr}$ is

$$\langle \mathcal{O}_{impr}^2 \rangle - \langle \mathcal{O}_{impr} \rangle^2 \; = \; \langle \mathcal{O}_{impr} \rangle - \langle \mathcal{O}_{impr} \rangle^2 \; \approx \; \langle \mathcal{O}_{impr} \rangle \equiv \langle \mathcal{O} \rangle \ll 1 \; . \tag{51}$$

For a given distance R, this gain is largest at small correlation length ξ, whereas the gain from reducing autocorrelations with the loop algorithm is largest at large ξ. On the other hand the non-improved estimator can have a sizeable amount of self-averaging at small ξ, so that the gain from using improved estimators as just a measurement tool tends to be moderate in practice (saving, e.g., a factor of 2 in computer time). See, however, sections 3.8 and 3.10 for other drastic effects of using improved estimators.

Especially simple estimators can also be derived for magnetic susceptibilities. The uniform magnetic susceptibility at vanishing magnetic field can be expressed as the sum over all correlation functions:

$$\langle \chi \rangle = \frac{\beta}{V} \left\langle \left(\sum_{\mathbf{r}} \frac{1}{2M} \sum_{\tau} S^z_{\mathbf{r},\tau} \right)^2 \right\rangle , \tag{52}$$

where V is the spatial volume (number of sites), $M = 2dL_t$ is the number of time slices in d dimensions, and $S^z_{\mathbf{r},\tau} = \pm 1$. This simplifies [25] in the XXZ case, by using

$$\sum_{\tau} \frac{1}{M} \sum_{\mathbf{r}} S^z_{\mathbf{r},\tau} \; = \; \sum_{(\text{clusters c})} \; \sum_{((\mathbf{r},\tau) \text{in } c)} \frac{1}{M} S^z_{\mathbf{r},\tau} \; = \; \sum_{\text{clusters c}} w_t(c) \tag{53}$$

to the sum of the square of the temporal winding numbers $w_t(c)$ of the clusters c:

$$\langle \chi \rangle = \frac{\beta}{4V} \left\langle \sum_{\text{clusters c}} w_t(c)^2 \right\rangle \tag{54}$$

In the single-cluster variant (section 2.12), the sum over the clusters in eq. 54 is also calculated stochastically. Since we pick a single cluster with a probability $\frac{|c|}{MV}$, where $|c|$ is the cluster size and MV is the number of sites in the space-time lattice, we have to compensate for this extra factor and obtain $\langle \chi \rangle = \frac{\beta}{4V} \left\langle \frac{MV}{|c|} \, w_t(c)^2 \right\rangle$. Similarly, eq. (49) implies that for an antiferromagnet with only horizontal breakups, the *staggered* susceptibility corresponds to the sum over the squares of all cluster sizes,

$$\langle \chi_s \rangle = \frac{\beta}{V} \left\langle \left(\sum_{\mathbf{r}} (-1)^{\mathbf{r}} \frac{1}{2M} \sum_{\tau} S^z_{\mathbf{r},\tau} \right)^2 \right\rangle \; = \; \frac{\beta}{4VM^2} \left\langle \sum_{\text{clusters c}} |c|^2 \right\rangle . \tag{55}$$

Further improved estimators can be constructed, including cases with a sign problem [21]. Moreover, it was recently discovered [12] that by clever use of the improved estimator for the fermion sign, one can perform fermionic simulations without any sign problem; see section 3.10.

2.12 Single-Cluster Variant

As in Swendsen-Wang Cluster updates, there are several ways to perform an update of the worldline configuration $\mathcal{S}$ with the required detailed balance with respect to $W(\mathcal{S}, G)$, eq. (15). There are two important approaches:

(i) *Multi-Cluster Variant:* Determine the whole graph (set of loops) G and flip each cluster (set of mutually glued loops) in G with suitable probability (see section 2.6)

(ii) *Single-Cluster Variant* [44, 1] Pick a site (i_0, l_0) at random, and construct only the cluster that includes that site. This can be done iteratively, by following the course of the loop through (i_0, l_0) until it closes, while determining the breakups (and thus the route of the loop(s)) only on the plaquettes which are traversed. At each plaquette at which a frozen breakup G^{ii} is chosen, the current loop is glued to the other loop traversing this plaquette. That other loop (and any loops glued to it) then also has to be constructed completely. Flip the complete cluster with probability p_{flip} satisfying detailed balance with respect to A_{global} to get to a new spin configuration. Note that when $A_{global} = const$, we can choose $p_{flip} = 1$ instead of $\frac{1}{2}$.

Both approaches satisfy detailed balance in eq. (15). One may think of the single-cluster variant as if all clusters had actually been constructed first, and then one of them chosen at random, by picking a site, to make an update proposal.

The advantage of the single-cluster variant [44, 35] is that by picking a random site, one is likely to pick a *large* cluster, whose flip will produce a big change in the configuration and thus a large step in phase space. This can reduce critical slowing down still further. The effort (computer time) to construct the single-cluster is proportional to its length. Normalized to constant effort, one finds indeed that the single-cluster variant (and the corresponding single-cluster update for Swendsen-Wang-like algorithms) usually have even smaller dynamical critical exponents (see appendix B) than the multi-cluster variant. Note that improved estimators get a different normalization in the single-cluster variant (see eq. (54) and below). In some circumstances, the multi-cluster variant can still be advantageous overall, for example when employing parallel [46] or vectorized [47] computers.

2.13 Performance

The most important advantages and limitations of the loop algorithm have already been summarized in the introduction. Let us be more explicit here. Further aspects of the performance are mentioned in the following sections.

Autocorrelations: The biggest obstacle which the loop algorithm addresses are the long autocorrelation times of worldline algorithms with local updates, as discussed in

the appendices (see eq. (79)). They require a proportional increase in computer time, so that simulations for large systems and/or low temperatures quickly become impossible. The loop algorithm appears to remove these autocorrelations completely in many cases, like the spin $\frac{1}{2}$ Heisenberg AF in any dimension, the two-dimensional spin $\frac{1}{2}$ XY-model, and the spin 1 Heisenberg chain. For large systems and low temperatures this can save many orders of magnitude in computer time. As one striking example, see the gain in autocorrelation time for the one-dimensional Hubbard model in fig. 7 in section 3.9. Autocorrelations and critical slowing down have been carefully determined in the original loop algorithm paper [1] for the nonquantum six-vertex model, with the single-cluster variant of the loop algorithm. In [48], a related study was done in which spatial winding was allowed to vary, with similar results for autocorrelations. In the massless phase (infinite correlation length) at $\frac{a}{c} = \frac{b}{c} = \sqrt{2}$, the loop algorithm completely eliminates critical slowing down, i.e. the autocorrelation times are small and constant, with $z_{int}^{MC} \approx 0$ for all measured quantities, and $z_{exp}^{MC} = 0.19(2)$. On the KT transition line, the exponential autocorrelation times are slightly larger (up to 20 on a 256^2 lattice), with $z_{exp}^{MC} = 0.71(5)$, yet for the integrated autocorrelation times, which are relevant for MC errors, we saw barely any autocorrelations in either case, up to the largest lattices of size 256^2. In contrast, local updates indeed showed very long autocorrelation times, and $z^{MC} = 2.2(2)$, as expected.

Continuous time: In section 3.7 we shall explain how the time continuum limit $\Delta\tau \to 0$ can be taken immediately in the loop algorithm, eliminating the Trotter approximation, reducing storage and CPU-time, and thus further extending the accessible temperature range.

Improved Estimators: The use of improved estimators (section 2.11) provides additional gains. For example, in ref [49] it has been possible to calculate the spin-spin correlation function (which in standard updates has large variance) down to values of 10^{-5}.

Change of global quantities: Since the loops are determined locally by the breakup decisions, they can easily, "by chance", wind around the lattice in temporal or in spatial direction. An example is given in fig. 6. The flip of such a loop then changes a global quantity (magnetization, particle number, spatial winding number). (Of course one can also choose to restrict the simulation to part of the total phase space, e.g. the *canonical ensemble* by not allowing such flips). This kind of configuration change is virtually impossible with standard local methods. It has been used to investigate e.g. the KT transition in the quantum XY model [50].

Strong fields, asymmetries, or site disorder can seriously impair the performance. They are discussed in sections 3.4 and 3.5.

Freezing: For the loop algorithm itself, apart from effects of global weights, models which require finite freezing weights w^{ii} might be (but need not be) difficult. The intuitive argument can easily be understood. If two different loops meet at a "frozen" plaquette (i.e. one for which the breakup G^{ii} was chosen), they are glued together. If this happens at overly many plaquettes, then the cluster of glued loops which must be flipped together can occupy most of the lattice. The flip of such a

cluster is not an effective move in phase space. It is basically equivalent to flipping all of the (few !) spins outside of that cluster. As an example, in ref. [1] we also investigated versions of the loop algorithm in which w^{ii} was (unnecessarily !) chosen finite. Sizeable autocorrelations were the result. Note however that it has *not* been tested whether such an impairment also occurs in cases where the *minimal* freezing is finite. (Ref. [51] includes freezing but also a large magnetic field). As an example that freezing need not always be bad, note that, as mentioned in section 2.7 [32], the limiting case $J_{x,y} \to 0$ of the loop algorithm is the classical Swendsen-Wang cluster algorithm, in which "freezing" is the only operation. Yet this cluster algorithm can also completely eliminate critical slowing down in the corresponding classical models.

Implementation: Implementation of the loop algorithm is actually considerably easier than for local updates, which, especially in more than one dimension, require rather complicated local updates [52].

The loop algorithm can be vectorized and parallelized similarly to the Swendsen Wang cluster algorithm (see e.g. [46, 47]). A vectorized version was used in ref. [50]. Vectorization or parallelization of the breakup process is trivial. The computationally dominant part is to identify the resulting clusters. This is equivalent to the well know problem of connected component labeling. For a brief discussion and references, see ref. [53]. The optimal strategies are likely to be different from the Swendsen Wang case, because loops are linear objects.

3 Generalizations

So far we have purposely restricted ourselves to XYZ-like (1+1)-dimensional models in order to simplify the presentation. This covered both the case of spin $\frac{1}{2}$ quantum spin models, where we have inserted eigenstates $|S^z_{il}\rangle = |\pm 1\rangle$ (or eigenstates along a different quantization axis) [5] and models of fermions or hard core bosons, where we have inserted occupation number eigenstates [5]. We have developed the formalism for the general anisotropic XYZ-like (eight-vertex-like) case. We have computed explicit update probabilities for all XXZ and most XYZ-like cases.

Let us now describe further generalizations, several of which are immediate. For all generalizations here, with a slight modification for continuous time (section 3.7), it remains true that *locally* on the vertices we have a situation like in the six- (or eight-) vertex model, so that the loop formulation described above can be applied directly.

3.1 Arbitrary spatial dimension

There is vitually no change algorithmically in going to higher dimensions [1], if one chooses to stay on a vertex-lattice. Let us look at two spatial dimensions as an

example. The even/odd split of the Hamiltonian in eq. (2) can be generalized to

$$\hat{H} = \sum_{\nu} \hat{H}_{\nu} = \sum_{\nu} \sum_{i} \hat{H}_{i,i+\hat{\nu}} \tag{56}$$

with a separate $\hat{H}_{\nu}$ for each direction of hopping (resp. spin coupling) in the Hamiltonian $\hat{H}$. For a two-dimensional square lattice with nearest neighbor hopping we thus get 4 parts $\hat{H}_{\nu}$, each the sum of commuting pieces living on single bonds, in complete analogy with the one-dimensional case.

After the Trotter-Suzuki breakup, eq. (3), these single bonds again develop into shaded plaquettes. Each Trotter timeslice now has 4 subslices. Locally on each shaded plaquette we have the *identical* situation as in (1+1) dimension. Thus the loop algorithm can be applied *unchanged*. The only thing that changes is the way that different plaquettes connect. (Thus it is easy to write a loop-cluster program for general dimension. This contrasts with the traditional local worldline updates, where a number of different rather complicated updates are necessary [52] to achieve acceptable performance). The same construction can be applied as long as the lattice and the Hamiltonian admit a worldline representation in which commuting pieces of the Hamiltonian live on bonds.

3.2 Long range couplings

Hopping or spin-spin interactions beyond nearest neighbor can be handled by the same approach as higher dimensions, namely by introducing extra parts $\hat{H}_{\nu}$ in the split of the Hamiltonian, i.e. extra "bonds" on the lattice, with a corresponding set of shaded plaquettes living on separate Trotter time subslices.

When the number of additional couplings is large, this approach becomes impractical. It is less cumbersome in the continuous time version of the loop algorithm (section 3.7).

3.3 Bond disorder, diluted lattices, and frustration

Bond disorder refers to spatial variations in the spin couplings J_{ij} (resp. hopping strengths t_{ij} and/or density-density couplings V_{ij}). This modifies the loop-construction probabilities *locally*, making them plaquette-dependent. Otherwise nothing changes ! (One does need to check whether ergodicity is still achieved.)

The same is true for diluted lattices, which can be viewed as a case of bond disorder in which the coupling vanishes completely on some bonds.

The situation is different for frustrated couplings whence some matrix elements become negative and cannot be transformed to a positive representation. Whereas the loop algorithm itself is unaffected, this produces a second type of sign problem which needs to be handled in the same way as the fermion sign problem (section 3.9).

If the strength and/or frequency of frustrated matrix elements is not too large, this sign problem can remain manageable. In general, however, it has so far precluded simulations of strongly frustrated models. It is possible to find improved estimators for frustrated couplings [21], to alleviate this sign problem. Note that for non-bipartite lattices, the breakup probabilities have to be modified to ensure ergodicity of the algorithm; see section 2.9. Promisingly, the strategy outlined in section 3.10 may also be applicable to frustrated systems.

3.4 Asymmetric Hamiltonians: Magnetic field, chemical potential

Asymmetric weights can be caused by site disorder, by a magnetic field in $\hat{S}^z$-direction or equivalently a chemical potential, or by other non "particle-hole-symmetric" terms like e.g. quadratic interactions of softcore bosons. Because of eq. (29), these parts of the Hamiltonian should be taken into the global weight A_{global}, such that W^{plaq} is symmetric with respect to a flip of all four spins on each shaded plaquette. Then they contribute to the flip probabilities p_{flip} of the clusters.

A magnetic field h (or chemical potential μ) affects only clusters which change the number of worldlines, i.e. which wind around the lattice in temporal direction. The acceptance rate for the flip of such clusters is proportional to $\exp(-\beta h n_w)$, where n_w is the temporal winding number of the cluster. At sufficiently *large values of* βh the acceptance rate becomes very small on average and results in an *exponential slowing down* of the simulation. This was indeed observed in ref. [54] for $\beta h \gtrsim 5$. For such large diagonal fields, the "worm-algorithm" (section 4.1, see also 4.2) was shown in the same paper to be much better suited. Slowing down of the loop algorithm was also observed with large site disorder [55]. To minimize the acceptance problem, one should normally choose A_{global} such that its fluctuations are minimized. Somewhat surprisingly, it has also been reported [56] that supplementing the cluster updates with explicit global flips of worldlines helps considerably, even at rather larger values of βh.

The canonical ensemble (or ensemble of constant magnetization) is simulated by disallowing the flip of clusters which would change the number of worldlines. It is also possible, and can reduce autocorrelations, to allow the number of worldlines to fluctuate, and afterwards treat each subset of configurations with a certain number of worldlines as a canonical simulation.

An interesting variant of the loop algorithm which works well for very large diagonal fields was recently introduced by Syljuasen [57]. In some cases, the difficulty of magnetic fields can be overcome completely by using *transverse* instead of diagonal fields, as described below.

3.5 Transverse field

There is a different way to treat magnetic fields and chemical potentials. One can change the axis of quantization to turn a magnetic field into the x-direction. The field operator $\hat{S}^x = \frac{1}{2} \left(\begin{smallmatrix} 01 \\ 10 \end{smallmatrix} \right)$ then flips the direction of a spin. It can act at any spacetime point (il). Therefore the worldlines now become piecewise continuous, with spin flips wherever $\hat{S}^x$ acts. For the isotropic Heisenberg model, the physics is invariant under this rotation.

Rieger and Kawashima [10] have shown how to treat such source operators in a cluster algorithm[11]. They investigate the Ising model in a transverse field, for which $J_x = 0$ and all worldlines are thus straight. Here we apply their method to the loop algorithm. For the transverse Ising model, see also [5].

We enlarge our graph to consist not only of breakups but also of the information, for each spacetime site, whether $\hat{S}^x$ acts there or not. $\hat{S}^x$ occurs in matrix elements like $W_h \equiv \langle + | \exp(\Delta\tau h \hat{S}^x) | - \rangle = \sinh(\frac{\Delta\tau}{2} h)$. Its presence or absence in a graph can be treated with Metropolis probability[12]: Tentatively put additional operators $\hat{S}^x$ with probability[13] $p = W_h$ on any lattice site where there is no $\hat{S}^x$ yet. After choosing breakups as usual we now have *segments* of loops between successive occurences of sources $\hat{S}^x$. Flip each segment[14] with probability $\frac{1}{2}$ (modified with any remaining external weight A_{global}). Finally, remove the sources between segments of equal orientation. This completes one update.

Now the magnetic field (chemical potential) can be of arbitrary size, without acceptance problems (section 3.4). We can measure off-diagonal Greens functions of any order in $\hat{S}^x$ directly in the worldline representation, as well as in the loop representation (section 3.8). When the original model is not isotropic, one can still rotate the magnetic field into x direction, obtaining an anisotropic XYZ model (see end of section 2.1). The loop algorithm for this model (section 2.7) involves breakups for which the arrow direction changes on both loop segment, which is equivalent to the original breakups plus two sources.

The procedure just described works in the ferromagnetic case. For an antiferromagnet, we have to perform a rotation $\hat{S}^{x,y} \to -\hat{S}^{x,y}$ on one sublattice to achieve positive matrix elements in eq. (6). This minus sign stays with the source operators $\hat{S}^x$, multiplies the configurational weight $W(\mathcal{S})$[15] and results in a sign problem. Recently, Chandrasekharan, Scarlet, and Wiese [11] showed that this sign problem can be removed completely with the meron approach described in section 3.10. To include sources, they use a method similar to that of Rieger and Kawashima. They introduce additional time slices and employ the Swendsen-Wang algorithm [7, 35] to

[11] Note that with sources, local updates allow the change of global updates and become ergodic too. See also section 4.1.

[12] I.e. *propose* creation of new operators with $p = W_h$, and deletion of existing ones with $p = 1$.

[13] In continuous time [10] this become constant probability $W_h/\Delta\tau$ per time.

[14] When there is freezing, we similarly have to flip subgraphs of clusters.

[15] I thank Matthias Troyer for this observation.

create or delete bonds (as part of loops) on the additional timelike lattice bonds, which is equivalent to the Metropolis procedure described above. They show that for large fields, the algorithm with transverse fields performs orders of magnitude better than the original one with diagonal fields.

The strategy described here, namely to include operators $\hat{S}^x$ stochastically, resembles their occurence in the stochastic series expansion (section 4.2). Indeed, one can also separate other operators from the Hamiltonian and include them stochastically in the same manner as $\hat{S}^x$ [41]. One might for example separate the Hamiltonian into an isotropic Heisenberg-like part, which is then decomposed into breakups without freezing, plus an interaction proportional to $(2\hat{S}_i^z\hat{S}_j^z + \frac{1}{2})$ such that the latter has matrix elements 0,1. Interestingly, its stochastic occurence is again equivalent to a freezing operation. The effect is the same as if it had been kept in the Hamiltonian to compute breakup probabilities as usual. Similar analogies exist for nondiagonal operators like $\hat{S}_i^x\hat{S}_j^x$.

3.6 Higher Spin representations

The loop algorithm for spin models has so far been formulated for the spin-$\frac{1}{2}$ case. One way to extend it to higher spin representations would be [1] to use the corresponding vertex representation (19-vertex model for spin-1) and to try the same formalism as for spin-$\frac{1}{2}$.

Kawashima and Gubernatis have successfully employed a different approach [6, 13]. They write higher spin representations as a product of spin-$\frac{1}{2}$ representations, with a projection operator onto the proper total spin. They arrive at new "shaded plaquettes", between the different spin-$\frac{1}{2}$ representations at each space-time site. Locally on each plaquette, the situation looks again like a six- or eight- vertex model. By the same approach, Kawashima also treated the anisotropic XYZ case for general spins [14]. (Here the number of different graphs quickly proliferates). This generalization was successfully tested on an antiferromagnetic Heisenberg chain with $S = 1$, finding complete removal of autocorrelations [13].

Since the Hamiltonian commutes with the projection operator, the procedure can be simplified [15, 16, 17] by projecting onto total spin S only at time zero. For general spin S [16, 17] there are again $2S$ spin-variables per site. Breakups with the usual probabilites (section 2.7) are allowed between any pair of spin variables from neighboring sites. Projection is achieved at time zero by symmetrizing over permutations of the worldline variables at each site, subject to worldline continuity. This procedure immediately generalizes to continuous time (section 3.7) [15, 16].

3.7 Continuous time

As one of the most important generalizations, Beard and Wiese [8] have shown that within the loop formulation, one can directly take the time continuum limit $\Delta\tau \to 0$ in the Trotter-Suzuki decomposition, eq. (3). In fact, one can also write the original spin model directly in continuous time when H has a countable basis [58, 59, 26].

In continuous time it is appropriate to describe worldlines by specifying the times t_i at which a worldline jumps to a different site. This jump is now instantaneous. Technically, this can be implemented with, e.g., a linked list of "events" for each spatial site.

Let us re-describe the iterative construction of a single loop for the six-vertex case in a suitable language. We begin with discrete time. Let the loop be moving upwards in time along a worldline at site j. At each plaquette which it traverses, it will either continue to move vertically or, with the prescribed breakup probability, jump to the neighboring site. Depending on the worldline configuration at the current plaquette, a jump can either only be diagonal, so that the loop continues to move upwards along a worldline, or only horizontal, so that the loop changes direction and begins moving downwards in time on an empty site.

While the loop moves upwards in time at site j, and as long as the configuration of the neighboring worldlines does not change, i.e. in a "constant neighborhood" $t_1 < t < t_2$, the breakup probabilities are the same on all traversed plaquettes. In each such region we can take the continuum limit $\Delta\tau \to 0$. The probability of the loop to jump to a neighboring site on a single plaquette then transforms to a constant probability per unit of time for such a jump. The situation is the same as in radioactive decay, with a decay constant $\lambda = \lim_{\Delta\tau\to 0}(p_{breakup}/\Delta\tau)$. For example, in the Heisenberg antiferromagnet we get from eq. (41) the decay constant $\lambda = J/2$ for moving to a specific unoccupied neighbor of site j. The total probability to jump away from the current site is the sum of the probabilities to jump to the unoccupied neighbors. In places where the neighborhood changes, the total jump-probability changes, too. It can also be 1, e.g. for the Heisenberg AF in those places where the worldline itself jumps, see eq. (41). Eventually the single loop closes and can be flipped as usual, with the continuum limit of the flip probability p_{flip} [21]. Cases with freezing can be handled in the same way. In addition to constructing a single-cluster, one can of course also construct all clusters, as in the multi-cluster variant.

We sketch a prescription for single loop simulations of the spin $\frac{1}{2}$ Heisenberg antiferromagnet, section 2.8, in continuous time. (See also ref. [21], appendix). Choose a site i at random. Start a loop at time $t_1 = 0$, moving upwards in time. The loop construction now iterates the following procedure: Determine the time interval $t_1 < t < t_2$ (equations are specified for moving upwards in time) during which the set of sites to which the loop can move does not change. This set consists of site i and those neighboring sites j (sites to which i is coupled by the Hamiltonian), which have the opposite worldline occupation from site i and which have not yet been traversed

by the loop between t_1 and t_2. For each such neighbor j draw a random number τ_{ij} from the distribution $\lambda_{ij} \exp(-\lambda_{ij}\tau_{ij})$, where $\lambda_{ij} = J_{ij}/2$ is the corresponding decay probability per time. Now extend the loop on site i up to time $t_1 + \min_j(\tau_{ij})$ and let it jump to the corresponding site j there. [16] Moving in time-direction on site i corresponds to choosing a vertical breakup for all infinitesimal "plaquettes" connecting site i to its neighbors. Moving to site j corresponds to a horizontal breakup, so that for the Heisenberg antiferromagnet the loop will now move in the opposite time-direction at that site. In case that all transition times $t_1 + \tau_{ij}$ exceed t_2, move the loop along site i up to time t_2. If the worldline at site i jumps at time t_2, the loop must do the same jump an continue in the previous direction (corresponding to a diagonal breakup, see eq. (41)). Now iterate this procedure.[17] Eventually the loop closes and can be flipped as usual.

The continuous time limit has several important advantages over the discrete time case. It removes completely the systematic error from the Trotter breakup, thus also removing the cumbersome need for calculations at several values of $\Delta\tau$ in order to extrapolate to $\Delta\tau = 0$. In addition, worldlines are specified much more economically by just specifying their transition times. This helps especially at low temperatures, by strongly reducing the storage requirements for a simulation. Longer range couplings imply larger sets of neighbors j to treat at each step. This is still cumbersome, but more economical than introducing extra Trotter slices. The advantages of the loop algorithm are preserved.

Other approaches which work without Trotter approximation are the worm algorithm (section 4.1) and the stochastic series expansion (section 4.2). Both are not cluster algorithms per se, but they are directly applicable to many models for which the loop algorithm is not well suited.

3.8 Off-diagonal operators

Brower, Chandrasekharan, and Wiese showed [9] that one can obtain all Greens functions, even off-diagonal ones, by measuring properties of the loop clusters, i.e. improved estimators, instead of properties of the worldline configurations. We first discuss the off-diagonal Greens function $\langle \hat{S}_i^+ \hat{S}_j^- \rangle$ between spacetime sites i, j as an example. It corresponds to a configuration with an additional propagator from i to j, i.e. with a partial worldline. Such a configuration never occurs during the simulation. Thus, in a regular worldline Monte-Carlo with continuous worldlines, off-diagonal Greens functions cannot be measured at all[18]. Instead, one would have to perform

[16] Alternatively one can use the sum of the decay rates λ_{ij} to determine a transition time, and then decide where to move, according to the ratios of the λ_{ij}.

[17] Note that when the transition probability per time is constant, the stochastic determination of a transition *time* can be interrupted and iterated arbitrarily without affecting the outcome. Thus the interruption at t_2 is allowed.

[18] One can measure equal-time correlation functions by introducing open b.c. on one timeslice [29].

a separate simulation with fixed sources for each pair of sites (i,j), and somehow measure and adjust the normalization $Z_{\text{sources } i,j}/Z_{\text{no sources}}$. This is not feasible with the usual worldline method. Note, however, that off-diagonal two-point functions can be measured directly with the worm algorithm (section 4.1). They are also directly accessible when there are transverse fields, with the loop algorithm (section 3.5) and with stochastic series expansion (section 4.2).

In the loop algorithm, $\langle \hat{S}_i^+ \hat{S}_j^- \rangle$ is easy to measure even without transverse fields. Let us look at a given configuration of loops. If spacetime sites i and j are connected by a loop, we could flip the partial loop from i to j (or from j to i, depending on the current spins at i,j) to obtain the desired propagator, and thus a contribution to $\langle \hat{S}_i^+ \hat{S}_j^- \rangle$. Note that we do not actually need to perform such a partial loop flip as a Monte Carlo update, we can just virtually do it to perform the measurement as an improved estimator (section 2.11). When there is no external asymmetry A_{global}, the proper flip probability for use in the improved estimator is $\frac{1}{2}$ for the partial loop, and the improved estimator for $\hat{S}_i^+ \hat{S}_j^-$ is

$$4(\hat{S}_i^+ \hat{S}_j^-)_{impr} = \begin{cases} 1 \cdot \phi_{ij}, & \text{if the sites } i,j \text{ are on the same loop,} \\ 0, & \text{otherwise.} \end{cases} \tag{57}$$

(For the antiferromagnet a phase factor $\phi_{ij} = (-1)^{|\mathbf{r}_i - \mathbf{r}_j|}$ appears due to the sublattice rotation (footnote 4); $\phi_{ij} = 1$ otherwise.) This resembles the improved estimator eq. (49) for the diagonal correlation function. There the criterion for a finite contribution is that the two sites are located on the same *cluster*, whereas here it is the same *loop*. For the isotropic Heisenberg antiferromagnet at momentum π and for the ferromagnet at momentum 0, the improved estimators eq. (49) and eq. (57) are identical, directly reflecting spin rotation invariance. Note that by using the improved estimator, one can *simultaneously* measure $\hat{S}_i^+ \hat{S}_j^-$ in a given cluster configuration for all pairs of sites i,j. When there is an external weight A_{global}, the improved estimator eq. (57) has to be modified, as described in section 2.11.

Brower, Chandrasekharan, and Wiese [9] show that improved estimators exist for general Greens functions, by developing an *operator* description of loop clusters. Each specific plaquette breakup can be written as a local transfer-matrix operator $\mathcal{T}$ that evolves worldline spins across the plaquette in time direction. The operator is specified by its matrix elements, which are linear combinations of Kronecker-deltas for the spin values, reflecting the specific breakup. The important point here is the *existence* of such an operator. A given graph G then corresponds to an operator $\hat{M}_G$, with $tr_{spins}\hat{M}_G = 2^{(number\ of\ clusters\ in\ G)}$ for the fieldless case. The partition function is $Z = \sum_G w(G)\, tr\hat{M}_G$ (see eq. (43)), where $w(G)$ is the product over plaquettes of the breakup-weights w^{ij} occuring in G. In this operator language in graph space, we can now also write expectation values:

$$\langle \hat{O} \rangle = \frac{1}{Z} \sum_G w(G) tr\left(\hat{O}\hat{M}_G\right) = \langle \frac{tr\hat{O}\hat{M}_G}{tr\hat{M}_G} \rangle_{MonteCarlo}, \tag{58}$$

where the trace is over spin configurations, and the last average over the cluster configurations generated in a Monte Carlo simulation. For the two-point Greens function, $\hat{O}$ consists of two source terms, and we get the improved estimator eq. (57). Similar results obtain for general n-point Greens functions, where now contributions can come from more than one cluster. For further details we refer to ref. [9].

3.9 Fermionic Models

Fermionic models can be treated in the same way as hard core bosons [5], with the addition of a fermionic sign for each permutation of worldlines [29]. The Monte Carlo simulation is performed with the modulus $|W(\mathcal{S})|$ of the configuration weight, and observables are determined as

$$\langle \mathcal{O} \rangle_W = \frac{\langle \mathcal{O}\ sign \rangle_{|W|}}{\langle sign \rangle_{|W|}}, \tag{59}$$

where *sign* is the sign of $W(\mathcal{S})$. Since[19] $\langle sign \rangle \sim \exp\left(-const \cdot \beta \cdot Volume\right)$, simulations have so far been restricted mostly to one-dimensional unfrustrated models and to small or very low-doping higher-dimensional or frustrated systems. The identical sign problem also occurs for fermion simulations with the loop algorithm (first performed in ref. [60]). By clever use of improved estimators, a solution of the sign problem was recently found for a large class of models (see section 3.10). This class does not so far include the standard Hubbard model (but see section 3.10) and the tJ model. For these models, several generalizations of the loop algorithm have been developed:

The Hubbard Model can be viewed as consisting of two systems of tight binding fermions, each mapping to an XXZ-model (plus fermion sign), coupled by the Hubbard interaction $U \sum_i n_i^{\uparrow} n_i^{\downarrow}$. It can therefore immediately be simulated by employing a loop algorithm for each of the XXZ models, and taking the Hubbard interaction as well as the chemical potential $\mu \sum_i (n_i^{\uparrow} + n_i^{\downarrow})$ into the global weight A_{global}, eq. (18). However, for large values of $|U|$ this procedure is not very efficient, since the global weight will fluctuate too strongly, resulting in small acceptance rates, especially for the flips of large loops.

Kawashima, Gubernatis, and Evertz [19] therefore added an additional new type of loop-update, called loop-exchange, which flips between spin-up and spin-down, leaving unoccupied sites in the worldline lattice unchanged. These loops move upwards in time on spin-up sites, and downwards on spin-down sites. The breakup probabilites can be constructed with the formalism of section 2. The loops were chosen to change direction (i.e. use a horizontal breakup) when spin-up and spin-down worldlines meet.

[19] The exponential form results since negative contributions to $\langle sign \rangle$ originate in finite spacetime regions more or less independently, from negative matrix elements and/or winding of worldlines. More formally [12], $\langle sign \rangle = \exp\left(-\beta V \Delta f\right)$ corresponds physically to the difference Δf in free energy between the fermionic model and the "bosonic" model $|W|$, whose nature depends on the quantization and weight function W that was chosen.

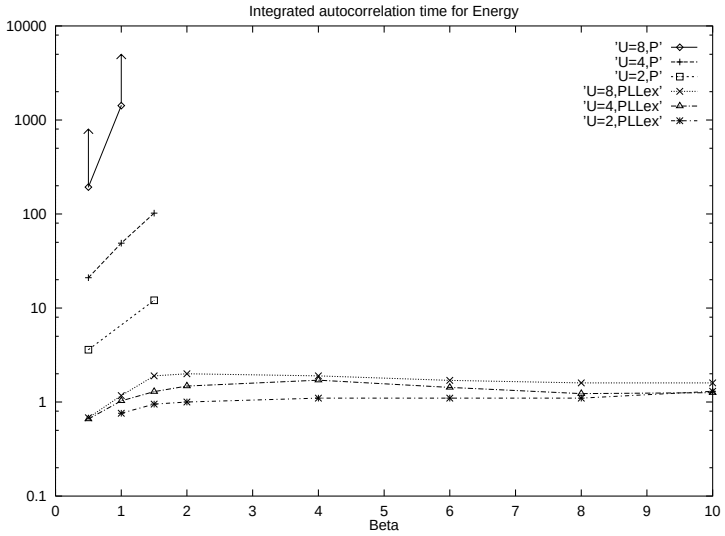

Figure 7: Comparison of the loop algorithm and local updates for the 1D Hubbard model (32 sites), adapted from ref. [19]. The figure shows the integrated autocorrelation time τ^E_{int} (which is proportional to the computer time required for a given accuracy) for the total energy E, on a logarithmic scale. The autocorrelation times for other quantities behave similarly. The upper three curves are for local updates, the others for loop updates including loop exchange. In both cases, $U = 2$, 4, and 8, with autocorrelation times increasing with growing U. With *local updates* the autocorrelation times are very large already at small β, and grow rapidly when β or U are increased (roughly consistent with $\tau \sim \beta^z U^x$ with $z \lesssim 2$ and $x \gtrsim 3$). For $U = 8$ these simulations did not converge, and only lower bounds for τ are shown. At $\beta \gtrsim 1.5$ none of the simulations with local updates converged. With *loop updates* the situation improves drastically. All autocorrelation times are less than about 2. Thus at large β and U several (likely many) orders of magnitude in computer time are saved.

Flips of these loops are not affected by the Hubbard interaction nor by the chemical potential, and can therefore always be accepted.

In the 1D Hubbard model this additional type of loop updates eliminated all remaining autocorrelations in the Hubbard simulations. An example is shown in fig. 7. For the loop updates, the autocorrelation times remain smaller than 2 at all temperatures. No slowing down is visible at all. For local updates, on the other hand, the autocorrelation times in fig. 7 show the expected rapid increase (see appendix B), consistent with $t \sim \beta^z$. They are orders of magnitude larger than the loop-autocorrelation times, already at small β and even for the energy as an observable, which as a locally defined quantity is expected to converge relatively fast in a local algorithm. Beyond $\beta \gtrsim 1.5$, the local Monte Carlo did not converge anymore. The autocorrelation times are expected to continue to grow. Note that the autocorrelation times for the local algorithm will additionally grow like $1/(\Delta\tau)^2$ for improved Trotter discretization, whereas the loop algorithm does not suffer from this effect, and can moreover be implemented directly in continuous time.

For the tJ model, several generalizations of the loop algorithm have recently been

developed [20, 21, 22, 23]. Here we have three kinds of site-occupation: spin-up, spin-down, and empty. The model can be simulated by a divide and conquer strategy [20, 21], using three different types of loop updates. In the first update, the empty sites in the worldline configuration are left untouched. The remaining sites with spin-up or spin-down can be updated with a loop algorithm very similar to that for the spin-$\frac{1}{2}$ Heisenberg antiferromagnet. In the second update, sites with spin-up are left untouched, and updates between spin-down and empty sites are done with a loop algorithm (which now looks similar to that for hard core particles). In the third update, loops on spin-up and empty sites are constructed. All three kinds of loops fall within the XXZ case discussed in section 2.7. The combined algorithm is indeed a working cluster algorithm for the tJ-model, ergodic, removing autocorrelations, and providing improved estimators which further reduce statistical errors. It was successfully used in ref. [21] on single tJ-chains without and with frustrated couplings, and on two and three coupled tJ chains.

Brunner and Muramatsu [22] use a different representation of the tJ model, in which only holes are fermionic. They appear in bilinear form and can only live on spin-down sites. In this representation, two kinds of loop updates therefore suffice. The authors study the stability of the Nagaoka state on two-dimensional lattices of size up to 10×10 at small J, with up to two holes, reaching temperatures down to $\beta t = 2500$ for a single hole and $\beta t = 150$ for 2 holes at $J = 0$, much lower than previously attainable with other methods.

Recently, Brunner, Assaad, and Muramatsu [23] developed a different very interesting method [23]. In the same representation, they simulate just the spin degrees of freedom (pure Heisenberg model), and can then calculate dynamical properties of a single hole in imaginary time by *exactly* integrating out the bilinear fermion degrees of freedom.

For relativistic gauge theories, Wiese et al. [61] have developed the so-called quantum-link formalism. They employ an additional artificial "time" direction, for which they can apply variants of the loop algorithm, e.g. for $U(1)$ gauge theory, aiming at efficient simulations of fermionic lattice QCD.

3.10 Fermion simulations without sign problem: The meron method

Chandrasekharan and Wiese [12] recently discovered an exciting possibility to overcome the fermion sign problem in a large class of models, by clever use of improved estimators. They treat the fermionic version of the spin $\frac{1}{2}$ antiferromagnetic Heisenberg model, i.e. spinless fermions with nearest neighbor repulsion $V = 2t$, with Hamiltonian

$$H = -t \sum_{\langle ij \rangle} (c_i^+ c_j + c_j^+ c_i) + V \sum_{\langle ij \rangle} (\hat{n}_i - \frac{1}{2})(\hat{n}_j - \frac{1}{2}), \tag{60}$$

and the usual relations $J_{x,y} = 2t$, $J_z = V$, $\hat{n}_i - \frac{1}{2} = \hat{S}_i^z$, and $n_i = 0, 1$ between particle language and spin language.

Then there are only vertical and horizontal breakups (section 2.8), no diagonal ones. In this case one can write down an improved estimator for the fermion sign in which the contribution from each cluster is *independent*. Namely, the fermion sign of the worldline configuration changes by a *factor* $(-1)^{1+n_w+n_h/2}$ when flipping a cluster, where n_w is the temporal winding number of the cluster, and n_h is the number of horizontal moves in the cluster. They denote by *"meron"* a cluster for which this factor is -1. Without chemical potential, each cluster can flip independently with propability $\frac{1}{2}$. Each meron then contributes a factor $((+1)+(-1))/2 = 0$ to the improved estimator for the sign, which is therefore zero whenever there is at least one meron. When there is no meron, the improved estimator is $+1$, since for the Heisenberg antiferromagnet *any* allowed loop configuration can be constructed from the reference staggered worldline configuration (see section 2.11), which has $sign = +1$, and since without merons, all configurations contributing to the improved estimator have the same sign. Thus,

$$(sign)_{impr} = \begin{cases} 1, & n_{meron} = 0 \\ 0, & n_{meron} > 0 \end{cases} \tag{61}$$

has no negative contributions. In a standard simulation, it is however dominated by contributions 0, and still has an exponentially bad variance.

Suppose we want to measure staggered two-point functions like $\mathcal{O} = (-1)^{i-j}(n_i - \frac{1}{2})(n_j - \frac{1}{2})$. To use eq. (59), we determine the improved estimator eq. (49) for $(\mathcal{O} \cdot sign)$. We can evaluate it by using the reference configuration. It is nonzero in two cases: (i) when there are no merons and i and j are on the same loop, it is $+\frac{1}{4}$; (ii)when there are two merons and i is located on one, j on the other meron, it is $+\frac{2}{4}$, since i can be on either cluster. For the susceptibility $\chi = \frac{\beta}{\mathrm{Vol}} \langle \hat{O}^2 \rangle$ of the staggered occupation $\hat{O} = \sum_{\mathbf{i}} (-1)^{\mathbf{i}} (\hat{n}_i - \frac{1}{2})$ we use eq. (55) for the improved estimator and get

$$\langle \chi \rangle = \frac{\langle (\chi \cdot sign)_{impr} \rangle}{\langle (sign)_{impr} \rangle} = \frac{\beta}{4M^2 \mathrm{Vol}} \frac{\langle \delta_{n_{meron},0} \sum_{(\text{clusters c})} |c|^2 + 2\delta_{n_{meron},2} |c_1| |c_2| \rangle}{\langle \delta_{n_{meron},0} \rangle} \tag{62}$$

which has contributions only from the zero and two meron sector.

Chandrasekharan and Wiese point out that one can therefore restrict simulations to $n_{meron} \leq 2$, thereby drastically reducing the visited phase space. This is achieved by starting with the reference worldline configuration and with completely vertical breakups. The breakups are then updated with the usual probabilities *plaquette by plaquette*, subject to the additional constraint that the total number of merons in a cluster configuration can at most be 2. Each plaqutte update requires that one follows the partial loops attached to that plaquette to determine their meron nature. Since the typical loop size increases with growing spatial and temporal correlation length, this step increases the computational cost, but at most by a factor proportional to space-time volume. The resulting loop clusters are flipped (e.g.) after all plaquettes have been updated, to get to a new worldline configuration (the sign of which is irrelevant).

The denominator in eq. (62) can be viewed as the remaining effect of the fermionic sign. This denominator would up to now still be very small. This can be avoided by *reweighting* the simulation so that roughly half of the generated cluster configurations have zero merons. The corresponding reweighting factor has to be cancelled when computing expectation values [12]. All n-point functions can be computed similarly by including up to n merons. Overall, the total computational effort per sweep is proportional to between V_t and V_t^2, where V_t is the spacetime volume, with strictly positive estimators and no exponential sign problem remaining. The sign problem is thus solved !

For the meron approach to work we need to satisfy two stringent conditions. There cannot be any diagonal breakups, so we need $V \geq 2t$; and there must be a reference worldline configuration in the sense of section 2.11. We also need to ensure ergodicity. These restrictions exclude, for example, the important case of free fermions, which could be used as an ingredient for simulations of the standard Hubbard model. They also currently exclude frustrated spin models. Hopefully it will be possible in the future to lift or circumvent these restrictions. Note also that no chemical potential has been included in our description. This may be achieved along the lines of section 2.11, by fixing certain clusters and keeping $p_{flip} = \frac{1}{2}$ for the others. Recently, Wiese et al. applied the meron approach to a class of extended Hubbard models (including a chemical potential) which obey the restrictions [24], and found that these models show phase separation upon doping.

Note that the meron approach can also be applied to other sign problems. Chandrasekharan, Scarlet, and Wiese [11] recently used it with great success for the antiferromagnetic Heisenberg model in a magnetic field (see section 3.5).

4 Related Methods

4.1 Worm Algorithm

In the standard worldline formulation with local updates it is almost impossible to compute single particle Greens functions like e.g. $\langle a(x_0, t_0)\, a^\dagger(x_1, t_1)\rangle$ (where $a^\dagger, a$ are creation and annihilation operators). In order to do so, one would have to introduce sources at (x_0, t_0) and (x_1, t_1) explicitely, with a partial worldline between these two points, and to perform a separate simulation for each such pair of coordinates (see also section 3.8).

A very elegant solution to this problem was provided by Prokof'ev, Svistunov, and Tupitsyn [27]. Their method can be viewed from the perspective of single loop construction. *During* that construction, there is a partial loop with two open ends. Flipping this partial loop would result in a partial worldline, i.e. a propagator between two sources, just as desired. Thus every step in a single-loop construction can be taken to provide a configuration for the measurement of Greens functions (see also section 3.8).

Prokof'ev et al. turn this observation around and explicitely construct a single propagator with two ends ("Worm") in continuous time, in the same way that a single loop would be constructed. The Monte-Carlo-moves are thus *local* in space and in time (within a constant neighborhood, see section 3.7). Each local step provides a new configuration to the measurement of Greens functions. When the sources meet and annihilate (equivalent to the closing of a single loop), contact is made to the sourceless partition function, thus providing the correct normalization. (This contact can also be provided by matching the Greens function at distance 1 with measurements of the corresponding energy expectation value in the sourceless case). Prokof'ev et al. supplement these moves by additional moves corresponding to the flip of small closed loops, in order to make the simulation faster. The Worm-algorithm is available for any spin-magnitude. It is ergodic in the same way that the loop algorithm is.

An additional very important advantage of the *local* updates is that *all* interactions in the Hamiltonian, like e.g. magnetic fields, can be taken into account in each step, without encountering prohibitively small acceptance rates. This is in contrast to the loop algorithm itself, which has to put unsuitable interactions into the global weight A_{global}, and which for some models (like soft core bosons) can apparently not be formulated in an efficient way (see section 3.4). At large magnetic field ($\beta h \gtrsim 5$ in ref. [54]), the worm algorithm is therefore much faster than the loop method, which slows down exponentially. At small or zero fields, the loop method has a dynamical critical exponent z close to zero, whereas for the worm method in one dimension, $z \approx 1$ [54], as expected for a guided random walk. When spatial or temporal correlation lengths are large, the loop algorithm will therefore be much faster (starting with about a factor 30 on a 10-site system in ref. [54]).

Prokof'ev et al. also provide a general framework for their continuous time worldline method (CTWL) for Hamiltonians with a discrete representation. They have applied the method very successfully to the 1D Bose Hubbard model [27, 62] with soft core bosons (see also [5]), also with disorder, as well as to numerous other problems.

4.2 Stochastic Series Expansion

The stochastic series expansion (SSE) [63], developed by Anders Sandvik, is a powerful method with a very different approach, in which the power series for $tr \exp(-\beta H)$ is treated stochastically. There are some similarities to the worm method and to the loop algorithm. Like these methods, SSE has no time discretization error. It acts in a discrete index space instead of time continuum, which can be advantageous numerically. The trace in $tr \exp(-\beta H)$ is taken in occupation number space, resulting in a worldline-like representation. These worldlines can be updated either locally, or, as recently developed [64], in a similar manner as for the worm algorithm, in which it is possible to backtrack along a path. Dynamic scaling may thus be similar to the latter. Large diagonal magnetic fields are also treated successfully

in this way. In the special case of isotropic interaction and vanishing magnetic field, backtracking can be avoided, and, remarkably, the construction of ref. [64] becomes very similar to the single loop version of the loop algorithm. Like the worm method, SSE is easily applicable to a larger class of models (without serious sign problem) than the loop algorithm. When there is a transverse field, it is ergodic and allows the measurement of off-diagonal correlation functions [65] (similar to section 3.5). SSE has been applied with great success to a large number of problems [64].

Interestingly, since breakups can be formulated in operator language (section 3.8), it is also possible to perform a stochastic series expansion in this representation, directly resulting in a loop algorithm[20] in index space [41]. This has not yet been done.

5 Some Applications

We briefly point to some of the applications of the loop algorithm (necessarily incompletely) to show in which ways it has been used in practice. More applications have already been mentioned in previous sections.

Spin $\frac{1}{2}$ isotropic Heisenberg antiferromagnets have been investigated in many studies. For variations of this model, the loop algorithm has been particularly valuable. It has for example allowed high precision calculations of the critical exponents of a quantum critical point [28] in a 2D depleted system. System sizes up to one million sites (at $\beta J = 5.5$) [66] and temperatures down to $\beta J = 1000$ (for 2500 sites) [67] have been accessible. No sign of critical slowing down has been reported in the calculations for this model without magnetic field or site disorder.

The even/odd structure and correlation lengths of spin ladders [49, 68] and coupled ladders [69], including a quantum phase transition in intraladder coupling [70] have been investigated in detail. Very precise studies of the finite size scaling of the 2D system [66, 71, 72] have been able to extract the asymptotic infinite lattice low temperature behavior, with correlation lengths up to 350,000, and to test the predictions from chiral perturbation theory. Similar studies have been performed for layered 2D systems [73].

Random systems have become accessible at sufficiently low temperatures and large system sizes, including the necessary averaging over disorder realizations. Bond disorder in chains [74, 16] as well as random nonmagnetic impurities in ladders [75], coupled ladders [76], and two-dimensional systems [67] have been investigated. The increased precision has also allowed simulations of frustrated models with a sign problem [77, 21].

For the quantum XY-model, accessibility of winding number fluctuations has allowed high precision studies of the Kosterlitz-Thouless transition, including the model

[20] Similar to the construction described in section 3.5, occurence of the operator $(2\hat{S}_i^z\hat{S}_j^z + \frac{1}{2})$, with matrix elements 0,1, would then correspond to a freezing operation. In a worm-like loop construction [64], backtracking can replace this freezing.

with site disorder [55], via the jump in helicity [50], as well as universality of the critical temperature in a bilayer system with small magnetic field [78].

With Ising anisotropy, magnetic field driven transitions could be studied in two and three dimensions for moderately large fields [51]. Griffiths-McCoy singularities have become accessible for the random transverse Ising model (section 3.5) in two dimensions [10, 79].

For highter spin representations, correlations in one [18, 17] and two [15, 72] dimensions have been analyzed with high precision, the effects of bond disorder [16, 80] and spin-$\frac{1}{2}$ impurities [81] in chains have been investigated, as well as a quantum phase transition from dilution on the square lattice [67].

Other models studied include spin-orbit coupling [82], a spin-Peierls system [83], relativistic gauge theory [61] (see section 3.9), hardcore bosons with disorder [55], and the fermionic models in sections 3.9 and 3.10, including two-hole tJ-model simulations at previously unaccessibly low temperatures.

6 Conclusions

The loop algorithm and its generalizations have opened up exciting new opportunities. Many of them remain to be investigated, e.g. for newly accessible fermionic models for which the sign problem has been overcome. A summary of advantages and limitations of the loop approach has been given in the introduction. Further improvements of the loop algorithm appear possible. For models in which it can be applied without sign problem and without overly big global weights, it offers large benefits. An example was given in fig. 7 in section 3.9. Attractive alternatives for less suitable models are the worm algorithm (section 4.1) and the stochastic series expansion [63]. Last, but not least, the mapping to a combined spin and loop model that is the basis of the loop algorithm promises further advances on the theoretical side.

Acknowledgements

I am indebted to Mihai Marcu for his long standing friendship and collaboration. Without him the loop algorithm would not exist. I am grateful to W. Hanke, M. Imada, D.J. Scalapino, and W. von der Linden for their support and ongoing interest. I would like to thank N. Kawashima, M. Troyer, and U.-J. Wiese, who have played an especially important and decisive role in the application and generalization of the loop algorithm for numerous discussions, and them as well as B. Beard, W. Koller, S. Todo, and D. Veberic for helpful comments on the first edition of this review.

Appendices

For correct Monte Carlo simulations it is essential, yet often neglected, that convergence and statistical errors are properly determined. To facilitate this task we provide a prescription. The requirements of detailed balance and ergodicity are briefly summarized in Appendix A. Appendix B discusses autocorrelations and their increase in physically interesting situation, which can drastically increase the necessary simulation times. The loop algorithm was designed to overcome this problem. Autocorrelations, especially very large ones, can easily can be overlooked and can be a serious problem in practice, causing (even drastically) incorrect results. Appendix C then describes how to properly ensure convergence and how to calculate correct error estimates. We also point to the recently developed so-called *Exact Monte Carlo* as an attractive potential alternative.

A Detailed Balance and Ergodicity

There are several excellent reviews of the Monte Carlo approach, e.g. in refs. [35, 84]. Here we briefly summarize some properties which we need elsewhere. The Monte Carlo procedure in classical statistical physics allows stochastic evaluation of expectation values

$$\langle \mathcal{O} \rangle = \frac{1}{Z} \sum_{\mathcal{S} \in \{\mathcal{S}\}} \mathcal{O}(\mathcal{S})\, W(\mathcal{S}) \tag{63}$$

with respect to the partition function $Z = \sum_{\mathcal{S} \in \{\mathcal{S}\}} W(\mathcal{S})$ and the phase space $\{\mathcal{S}\}$, by generating a Markov chain of configurations $\mathcal{S}_{(1)}$, $\mathcal{S}_{(2)}$, $\mathcal{S}_{(3)}$, ..., which is distributed like $W(\mathcal{S})$. Therefore one can compute $\langle \mathcal{O} \rangle$ from a sample of configurations

$$\langle \mathcal{O} \rangle = \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} \mathcal{O}(\mathcal{S}_{(k+i)}) \; . \tag{64}$$

(In practice, the first $k > \tau_{exp}$ configurations should be discarded to allow "thermalization" into the Boltzmann distribution. See appendix C.) A set of sufficient conditions to achieve this distribution is

(1) *Detailed Balance:* The transition probability $0 \leq p(\mathcal{S}_{(i)} \to \mathcal{S}_{(i+1)}) \leq 1$ of the Markov chain satisfies $W(\mathcal{S})\, p(\mathcal{S} \to \mathcal{S}') = W(\mathcal{S}')\, p(\mathcal{S}' \to \mathcal{S})$.

(2) *Ergodicity:* Every configuration $\mathcal{S} \in \{\mathcal{S}\}$ can be reached from every other configuration with finite probability in a finite number of steps.

Solutions for detailed balance are for example the Metropolis probability

$$p(\mathcal{S} \to \mathcal{S}') = \max(1, \frac{W(\mathcal{S}')}{W(\mathcal{S})}) \tag{65}$$

and the heat bath like probability

$$p(\mathcal{S} \to \mathcal{S}') = \frac{W(\mathcal{S}')}{W(\mathcal{S}) + W(\mathcal{S}') + const} \; . \tag{66}$$

It is often advantageous, as it is for the loop algorithm, to split the weight $W(\mathcal{S})$ into two parts:

$$W(\mathcal{S}) = W_1(\mathcal{S}) \cdot W_2(\mathcal{S}) \; . \tag{67}$$

Let $p_1(\mathcal{S} \to \mathcal{S}')$ be a transition probability that satisfies detailed balance with respect to W_1. We get a Monte Carlo procedure for W by using W_2 as a "filter" to accept or reject $\mathcal{S}'$. More precisely: First apply p_1 to propose a Markov step $\mathcal{S} \to \mathcal{S}'$. Then decide with a probability $p_{accept}(\mathcal{S} \to \mathcal{S}')$ whether to take $\mathcal{S}'$ as the next configuration in the Markov chain. Otherwise keep $\mathcal{S}$. Here p_{accept} only needs to satisfy detailed balance between $\mathcal{S}$ and $\mathcal{S}'$ with respect to W_2,

$$W_2(\mathcal{S}) \; p_{accept}(\mathcal{S} \to \mathcal{S}') = W_2(\mathcal{S}') \; p_{accept}(\mathcal{S}' \to \mathcal{S}) \; . \tag{68}$$

One can easily see that the overall update satisfies detailed balance with respect to W. For p_{accept} we can for example choose the heatbath probability $p_{accept}(\mathcal{S} \to \mathcal{S}') = W_2(\mathcal{S}')/(W_2(\mathcal{S}) + W_2(\mathcal{S}'))$. Ergodicity has to be shown separately for the overall procedure.

B Autocorrelations and Critical Slowing Down

Successive configurations $\mathcal{S}_{(1)}, \mathcal{S}_{(2)}, \mathcal{S}_{(3)}, ...$ in the Markov chain of a Monte Carlo configuration are correlated. Here we discuss the corresponding autocorrelation times, which can be extremely large. For other treatments of this topic, see e.g. references [85, 35, 84]. We follow refs. [35] and [19] in slightly simplified form. Define a Monte Carlo average from n measurements

$$\overline{\mathcal{O}} := \frac{1}{n} \sum_{i=1}^{n} \mathcal{O}_{(i)} \tag{69}$$

and define the autocorrelation function for the observable $\mathcal{O}$

$$\begin{aligned} C_{\mathcal{O}\mathcal{O}}(t) &:= \langle \mathcal{O}_{(i)} \mathcal{O}_{(i+t)} \rangle - \langle \mathcal{O}_{(i)} \rangle \langle \mathcal{O}_{(i+t)} \rangle \\ \approx \; \overline{C}_{\mathcal{O}\mathcal{O}}(t) &:= \tfrac{1}{n} \textstyle\sum_{i=1}^{n} \left\{ \left(\mathcal{O}_{(i)} - \tfrac{1}{n} \sum_{i=1}^{n} \mathcal{O}_{(i)} \right) \right) \left(\mathcal{O}_{(i+t)} - \tfrac{1}{n} \sum_{i=1}^{n} \mathcal{O}_{(i+t)} \right) \Big\} \; , \end{aligned} \tag{70}$$

with the normalized version

$$\Gamma_{\mathcal{O}\mathcal{O}}(t) := \frac{C_{\mathcal{O}\mathcal{O}}(t)}{C_{\mathcal{O}\mathcal{O}}(0)} \; . \tag{71}$$

Typically, $\Gamma_{\mathcal{O}\mathcal{O}}(t)$ is convex and will decay exponentially at large t like $e^{-|t|/\tau}$. Define the *exponential autocorrelation time* for the observable $\mathcal{O}$ by this asymptotic decay

$$\tau_{exp}^{\mathcal{O}} := \limsup_{t\to\infty} \frac{t}{-\log|\Gamma_{\mathcal{O}\mathcal{O}}(t)|} \ . \tag{72}$$

This is the relaxation time of the slowest mode in the Monte Carlo updates which couples to $\mathcal{O}$. The slowest overall mode is

$$\tau_{exp} := \sup_{\mathcal{O}} \left(\tau_{exp}^{\mathcal{O}}\right) \tag{73}$$

and corresponds to the second largest eigenvalue of the Markov transition matrix. (The largest eigenvalue is 1 and has the Boltzmann distribution as eigenvector).

If the $\mathcal{O}_{(i)}$ were statistically independent, then the error estimate of $\overline{\mathcal{O}}$ would be $\sigma/\sqrt{n}$ with

$$\sigma^2 = \frac{n}{n-1}\overline{C}_{\mathcal{O}\mathcal{O}}(0) \ . \tag{74}$$

Instead, the statistical error of $\overline{\mathcal{O}}$ is controlled by the *integrated autocorrelation time*

$$\tau_{int}^{\mathcal{O}} := \frac{1}{2} + \sum_{t=1}^{\infty} \Gamma_{\mathcal{O}\mathcal{O}}(t) \tag{75}$$

and becomes $\sigma_{int}/\sqrt{n}$ with

$$\sigma_{int}^2 \ \simeq \ 2\tau_{int}^{\mathcal{O}}\, \overline{C}_{\mathcal{O}\mathcal{O}}(0) \quad \text{for } n \gg \tau_{int}^{\mathcal{O}} \ . \tag{76}$$

Therefore a Monte Carlo run of n measurements effectively contains only $n/(2\tau_{int}^{\mathcal{O}})$ independent samples for measuring $\langle\mathcal{O}\rangle$. If $\Gamma_{\mathcal{O}\mathcal{O}}(t)$ is a single exponential $e^{-t/\tau_{exp}^{\mathcal{O}}}$, i.e. if only a single mode of the Markov transition matrix couples to $\mathcal{O}$, then $\tau_{int}^{\mathcal{O}} = \tau_{exp}^{\mathcal{O}}$, otherwise $\tau_{int}^{\mathcal{O}} < \tau_{exp}^{\mathcal{O}}$.

In simulations of classical statistical systems, autocorrelation times typically grow like

$$\tau_{int,exp}^{\mathcal{O}} \sim \min(L,\xi)^{z_{int,exp}^{MC}(\mathcal{O})} \ , \tag{77}$$

where L is the linear size of the system, ξ is the physical correlation length in the infinite volume limit at the same couplings, and z^{MC} is called the (Monte-Carlo) dynamical critical exponent. In general, z^{MC} depends on the observable $\mathcal{O}$, and $z_{int}^{MC}(\mathcal{O}) \neq z_{exp}^{MC}(\mathcal{O})$. (Note that $\tau_{exp}^{\mathcal{O}}$ is a correlation time, whereas $\tau_{int}^{\mathcal{O}}$ resembles the corresponding "susceptibility"; they will in general have different critical behavior.) For local unguided updates, including the case of local updates in the determinental formalism, one has so far always found

$$z^{MC,local} \gtrsim 2 \ . \tag{78}$$

The intuitive reason is that changes in a configuration have to spread over a distance $\min(L,\xi)$ in order to provide a statistically independent configuration. With local

updates, this spread resembles a random walk with step size one [86], which needs r^2 steps to travel a distance r. For local overrelaxed updates (see ref. [35] and section 4.1), z^{MC} can be as small as 1.

In nonrelativistic quantum simulations, space and imaginary time are asymmetric. With local Monte Carlo updates one can expect

$$\tau \sim \left\{ \max\left(\min(L,\xi)\,,\ \frac{1}{\Delta\tau}\min(\beta,\frac{1}{\Delta}) \right) \right\}^{z^{\mathrm{MC}}} , \tag{79}$$

where now L and ξ are spatial lengths, $\beta/\Delta\tau = L_t$ is the temporal extent of the lattice, Δ is the energy gap, and $1/(\Delta\Delta\tau)$ is the temporal correlation length. Again we need to distinguish $\tau^{\mathcal{O}}_{int}$ from $\tau^{\mathcal{O}}_{exp}$, and again $z^{MC} \gtrsim 2$ for local updates.

Close to phase transitions ($\xi \to \infty$) or at low temperatures and small gaps ($\Delta \to 0$), the autocorrelation times of local algorithms, with $z^{MC} \gtrsim 2$, will grow very fast. In addition, one needs to take the limit $\Delta\tau \to 0$. For local algorithms, this results in another large factor $1/(\Delta\tau)^{(z+1)}$ in required computer time. (We get "$(z+1)$" since each MC sweep has to update $\beta/\Delta\tau$ timeslices). With the loop algorithm, on the other hand, critical slowing down often disappears: $z^{MC} \approx 0$. In continuous time, the factor $1/\Delta\tau$ also disappears entirely, so that the autocorrelation times remain small even for large values of L, ξ, β, or $\frac{1}{\Delta}$.

C Convergence and Error Calculation

Since MC measurements are correlated (see appendix B), it is not at all trivial to calculate correct statistical errors, or even to ensure convergence of a MC simulation (see also fig. 7). For error calculations, there are two good strategies in practice, *binning* and *Jackknife*, which can also be combined. Let us emphasize that to ensure convergence, it is *indispensable* to begin simulations on extremely small systems and at unproblematic parameters (e.g. high temperature), and to slowly increase system size, while monitoring autocorrelations through binning and/or a thorough analysis of the time series and its autocorrelation function Γ for all observables to be measured (preferrably for all suspect observables). Otherwise, if one starts on too big a system, the simulation can all too easily be locked in some region of phase space, *without* any detectable signal in measured autocorrelations, but with possibly entirely wrong results.

Remarkably, it has recently been discovered that it is possible to perform Monte Carlo simulations with *exact* convergence, generating with moderate effort only completely independent configurations from the exact desired Boltzmann distribution. This so-called *Exact Monte Carlo* was developed as "Coupling From The Past" (CFTP) by Propp and Wilson [87]. Unfortunately, such a method is not available for Swendsen-Wang-like cluster algorithms. It has however already been used in the original CFTP paper for the pure cluster representation of the Ising model. This strategy should also be applicable to the loop algorithm.

Binning: (We follow ref. [19]). Group the n measurements $\mathcal{O}(i) \equiv \mathcal{O}_{(i)}$ into k bins of length $l = n/k$ with (e.g.) $l = 2, 4, 8, ...$. Compute k bin averages

$$\overline{\mathcal{O}}_b(l) := \frac{1}{l} \sum_{i=(b-1)l+1}^{bl} \mathcal{O}(i) \ \ , b = 1, .., k \tag{80}$$

and the variance of these averages

$$\sigma^2(l) := \frac{1}{k-1} \sum_{b=1}^{k} \left(\overline{\mathcal{O}}_b(l) - \overline{\mathcal{O}} \right)^2 \ . \tag{81}$$

This variance should become inversely proportional to l as the bin size l becomes large enough, whence the $\overline{\mathcal{O}}_b(l)$ as a function of b become statistically independent [85]. The expectation value of the quantity

$$\tau_{int}^{\mathcal{O}}(l) := \frac{l\sigma^2(l)}{2\sigma^2} \ , \tag{82}$$

where σ is given by eq. (74), grows monotonically in l. When statistical independence is approached, $\tau_{int}^{\mathcal{O}}(l)$ approaches the integrated autocorrelation time $\tau_{int}^{\mathcal{O}}$ from below. The converged asymptotic value of $\tau_{int}^{\mathcal{O}}(l)$ (or rather its expectation value) can therefore be used in eq. (76) to compute the actual statistical error of $\overline{\mathcal{O}}$. Note that $\tau_{int}^{\mathcal{O}}(l)$ will start to fluctuate at large l, since for finite number of measurements the number of bins k becomes small.

Convergence: If $\tau_{int}^{\mathcal{O}}(l)$ does not converge, then its expectation value at the largest l is a *lower bound* for $\tau_{int}^{\mathcal{O}}$, giving a *lower* bound for the error of $\overline{\mathcal{O}}$. In that case the MC run has not converged, and the data cannot be used to deduce physical results for $\langle \mathcal{O} \rangle$. Convergence of $\tau_{int}^{\mathcal{O}}(l)$ is a *prerequisite* for using the MC results. Since $\tau_{int}^{\mathcal{O}}$ varies for different observables $\mathcal{O}$, $\tau_{int}^{\mathcal{O}}(l)$ may have converged for some $\mathcal{O}$, and not for others. This is a dangerous situation, since the very slow modes visible in the nonconverged observables may be relevant for the apparently converged observables, too. Moreover, before starting measurements, the Monte Carlo configuration must be allowed to *thermalize*, i.e. to approach the Boltzmann distribution. It can be shown that the thermalization (from an arbitrary starting configuration) is governed by the overall exponential autocorrelation time τ_{exp}, i.e. the very largest time scale in the simulation. The thermalization time needs to be a reasonably large multiple of τ_{exp}. Therefore it is necessary to have at least an upper bound on τ_{exp} available. If an insufficient time is spent on thermalization, then the MC averages $\overline{\mathcal{O}}$ contain a systematic bias, and will converge more slowly.

An unfortunate problem in practice is that simulations may be started on an overly big system, for which – unbeknownst to the simulator – there are huge autocorrelation times. Then it may happen that within any feasible MC run, these large time scales remain invisible, so that the MC run appears to have converged, whereas in reality it has barely moved in phase space and the results may be completely wrong. (Take

for example the simulation of a simple Ising model with a local algorithm at low temperature. The total magnetization takes an exponentially large time to change sign. It may never do so during the simulation, and may appear converged at a large finite value, whereas the true average magnetization is zero.) Unless an "Exact Monte Carlo" method [87] can be used, the only way to avoid this problem is apparently to begin simulations on extremely small systems and away from problematic parameter regions, so that convergence is guaranteed by brute force. Slowly increasing system size, while measuring autocorrelation times, one can ensure that autocorrelations do not get out of hand. Note that this approach does not require much additional computer time, since simulations on small systems will be fast.

One rather sensitive and simple instrument to detect some autocorrelations long before they are visible in a binning analysis is to simply plot the MC evolution $\mathcal{O}(i)$, $i = 1, ..., n$ graphically and to look for long correlations by eye.

Autocorrelation function: A quantitative analysis of autocorrelations beyond $\tau_{int}^{\mathcal{O}}$, e.g. in order to calculate $\tau_{exp}^{\mathcal{O}}$, requires calculation of the autocorrelation function $\Gamma_{\mathcal{O}\mathcal{O}}(t)$. This is feasible only when $\overline{\mathcal{O}}$ has converged. Contrary to claims in the literature, one *cannot* reliably extract the integrated autocorrelation time $\tau_{int}^{\mathcal{O}}$ from $\Gamma_{\mathcal{O}\mathcal{O}}(t)$ by neglecting it beyond a "window" $t < W$ selfconsistently determined from the slope of $\Gamma_{\mathcal{O}\mathcal{O}}(W)$. Typically (but simplified [35]), we have

$$\Gamma_{\mathcal{O}\mathcal{O}}(t) = \sum_j c_j^{\mathcal{O}} \, e^{-t/\tau_j} \ , \tag{83}$$

with contributions from all eigenmodes of the Markov transition matrix, unless they are orthogonal to $\mathcal{O}$ (whence $c_j^{\mathcal{O}} = 0$). Therefore $\tau_{int}^{\mathcal{O}} \approx \sum_j c_j^{\mathcal{O}} \tau_j$ can get sizeable contributions from very large time scales τ_j, even when they couple only with small matrix elements $c_j^{\mathcal{O}}$ and are therefore not visible at small times t. This does indeed commonly happen in practice. A more reliable procedure is the following: Ensure convergence of $\overline{\mathcal{O}}$. Calculate $\Gamma_{\mathcal{O}\mathcal{O}}(t)$ for $t < t_{max}$, where t_{max} is chosen as large as possible while $\Gamma_{\mathcal{O}\mathcal{O}}(t)$ remains well above zero within error bars for all $t < t_{max}$. Compute estimates for $\tau_{exp}^{\mathcal{O}}$ and its matrix element $c_{exp}^{\mathcal{O}}$ from the (hopefully) asymptotic decay of $\Gamma_{\mathcal{O}\mathcal{O}}(t)$. Calculate $\tau_{int}^{\mathcal{O}}$ from eq. (75) by summing t up to the order of $\tau_{exp}^{\mathcal{O}}$ and computing the remainder of $\tau_{int}^{\mathcal{O}}$ from the asymptotic form of eq. (83), $\Gamma_{\mathcal{O}\mathcal{O}}(t) \sim c_{exp}^{\mathcal{O}} e^{-t/\tau_{exp}^{\mathcal{O}}}$. Of course, even this procedure will fail if the MC run is too short to show the largest autocorrelation times.

Jackknife: A binning-type analysis is a prerequisite for checking convergence. It also produces values for the autocorrelation times $\tau_{int}^{\mathcal{O}}$. However, standard error propagation becomes rather cumbersome for nonlinear quantities, like correlation functions in simulations with a sign-problem. It is much easier to compute errors with the jackknife procedure [88]. We give a brief recipe. Split the measured values $\mathcal{O}(i)$ into k groups of length $l = n/k$. To obtain the *asymptotic* error, l must be significantly larger than the relevant autocorrelation time $\tau_{int}^{\mathcal{O}}$. Compute the bin averages $\overline{\mathcal{O}}_b$, $b = 1, .., k$, eq. (80). Do this for all required observables $\mathcal{O}$.

Now perform the complete, possibly highly nonlinear, analysis of the MC-data

$k+1$ times: first with all data, leading to a result "$R^{(0)}$", then, for $j = 1, .., k$, with all data except those in bin j (i.e. pretend that bin j was never measured), leading to values "$R^{(j)}$". Then the overall result R is

$$\begin{array}{lcl} R & = & R^{(0)} - Bias, \quad \text{where} \\ Bias & = & (k-1)\,(R^{av} - R^{(0)}), \\ R^{av} & = & \frac{1}{k}\,\sum_{j=1}^{k} R^{(j)}\ , \end{array} \tag{84}$$

with statistical error

$$\delta(R) = (k-1)^{1/2} \left(\frac{1}{k} \sum_{j=1}^{k} (R^{(j)})^2 - (R^{av})^2 \right)^{1/2} . \tag{85}$$

In this procedure, error propagation is *automatic*. In each of the $k+1$ analyses, almost the full set of data is used, avoiding problems in the usual analysis like instabilities of fits. It is also possible to combine Jackknife and binning by repeating the Jackknife procedure for different bin lengths, to check for convergence and to compute integrated autocorrelation times and asymptotic errors according to eq. (82).

References

[1] H.G. Evertz, G. Lana, and M. Marcu, Phys. Rev. Lett. 70, 875 (1993).
[2] H.G. Evertz and M. Marcu, in *Lattice 92*, Amsterdam 1992, ed. J. Smit et al., Nucl. Phys. B S30, 277 (1993).
[3] H.G. Evertz and M. Marcu, in "Quantum Monte Carlo Methods in Condensed Matter Physics", ed. M. Suzuki, World Scientific, 1994, p. 65.
[4] The loop algorithm was first presented in a talk by the present author at *Lattice 91*, Tsukuba, Japan.
[5] R.T. Scalettar and G.G. Batrouni, this volume.
[6] N. Kawashima and J.E. Gubernatis, J. Stat. Phys. 80, 169 (1995).
[7] R. H. Swendsen and J. S. Wang, Phys. Rev. Lett. 58, 86 (1987).
[8] B.B. Beard and U.J. Wiese, Phys. Rev. Lett. 77, 5130 (1996).
[9] R. Brower, S. Chandrasekharan, and U.-J. Wiese, Physica A 261, 520 (1998).
[10] H. Rieger and N. Kawashima, Europ. Phys. J. B 9, 233 (1999).
[11] S. Chandrasekharan, B. Scarlet, and U.-J. Wiese, cond-mat/9909451.
[12] S. Chandrasekharan and U.-J. Wiese, Phys. Rev. Lett. 83, 3116 (1999).
[13] N. Kawashima and J.E. Gubernatis, Phys. Rev. Lett. 73, 1295 (1994).
[14] N. Kawashima, J. Stat. Phys. 82, 131 (1996).
[15] K. Harada, M. Troyer, and N. Kawashima J. Phys. Soc. Jpn. 67 (1998) 1130.
[16] S. Todo, K. Kato, and H. Takayama, "Computer Simulation Studies in Condensed Matter Physics XI," ed. D.P. Landau et. al., Springer Proc. in Physics, 1998.
[17] S. Todo and K. Kato, cond-mat/9911047.

[18] Y.J. Kim, M. Greven, U.-J. Wiese, and R.J. Birgeneau, Eur. Phys. J. B 4, 291 (1998).
[19] N. Kawashima, J.E. Gubernatis, and H.G. Evertz, Phys. Rev. B 50, 136 (1994).
[20] N. Kawashima, in "Computer Simulations in Condensed Matter Physics IX", ed. D.P. Landau et. al., Springer Proceedings in Physics, 1996.
[21] B. Ammon, H.G. Evertz, N. Kawashima, M. Troyer, and B. Frischmuth, Phys. Rev. B 58, 4304 (1998).
[22] M. Brunner and A. Muramatsu, Phys. Rev. B 58, R10100 (1998).
[23] M. Brunner, F.F. Assaad, and A. Muramatsu, cond-mat/9904150.
[24] J. Cox, C. Gattringer, K. Holland, B. Scarlet, U.-J. Wiese, hep-lat/9909119.
[25] U.J. Wiese and H.-P. Ying, Z. Phys. B 93, 147 (1994).
[26] M. Aizenmann and B. Nachtergaele, Comm. Math. Phys. 164, 17 (1994).
[27] N.V. Prokof'ev, B.V. Svistunov, I.S. Tupitsyn, Sov. Phys. - JETP 87, 310 (1998).
[28] M. Troyer, H. Kontani, and K. Ueda, Phys. Rev. Lett. 76, 3822 (1996); M. Troyer, M. Imada, and K. Ueda, J. Phys. Soc. Jpn. 66, 2957 (1997); M. Troyer and M. Imada, in "Computer Simulations in Condensed Matter Physics X", ed. D.P. Landau et. al., Springer Proceedings in Physics, 1997.
[29] J.E. Hirsch, R.L. Sugar, D.J. Scalapino, and R. Blankenbecler, Phys. Rev. B 26, 5033 (1982).
[30] H.F. Trotter, Proc. Am. Math. Soc. 10, 545 (1959); M. Suzuki, Prog. Theor. Phys. 56, 1454 (1976).
[31] R. J. Baxter, *Exactly Solved Models in Statistical Mechanics*, (Academic, New York 1989).
[32] N. Kawashima and J.E. Gubernatis, Phys. Rev. E 51, 1547 (1995).
[33] J. M. Kosterlitz and D. J. Thouless, J. Phys. C 6, 1181 (1973).
[34] E. H. Lieb, Phys. Rev. Lett. 18, 1967 (1046); E. H. Lieb and F. Y. Wu, in *Phase Transitions and Critical Phenomena* Vol. 1, C. Domb and M. S. Green, ed., (Academic, 1972) p. 331.
[35] A.D. Sokal, *Bosonic Algorithms*, in "Quantum Fields on the Computer", ed. M. Creutz, 1992. (Available electronically via ref. [87]).
[36] D. Kandel and E. Domany, Phys. Rev. B 43, 8539 (1991).
[37] P.W. Kasteleyn and C.M. Fortuin, J. Phys. Soc. Jpn. 26, (Suppl.) 11 (1969); and C.M. Fortuin and P.W. Kasteleyn, Physica 57, 536 (1972).
[38] H. G. Evertz, M. Hasenbusch, M. Marcu, K. Pinn and S. Solomon, Phys. Lett. 254B, 185 (1991), and Int. J. Mod. Phys. C3 , 235 (1992).
[39] M. Hasenbusch, G. Lana, M. Marcu and K. Pinn, Phys. Rev. B 46 (1992) 10472.
[40] J. Kondev and C. Henley, Nucl. Phys. B 464, 540 (1996).
[41] This discussion has apparently not appeared before.
[42] J. Kondev, Int. J. Mod. Phys. B11, 153 (1997).
[43] M. Sweeny, Phys. Rev. B 27, 4445 (1983).
[44] U. Wolff, Phys. Rev. Lett. 62, 361 (1989).
[45] U. Wolff, Nucl. Phys. B 334, 581 (1990).
[46] H. Mino, Computer Phys. Comm. 66, 25 (1991).

[47] H.G. Evertz, J. Stat. Phys. 70, 1075 (1993).
[48] M. A. Novotny and H. G. Evertz, in *Quantum Monte Carlo Methods in Condensed Matter Physics*, ed. M. Suzuki (World Scientific 1993).
[49] M. Greven, R.J. Birgeneau, and U.J. Wiese, Phys. Rev. Lett. 77, 1865 (1996).
[50] K. Harada and N. Kawashima, Phys. Rev. B 55, R11949 (1997); J. Phys. Soc. Jpn. 67, 2768 (1998).
[51] M. Kohno and M. Takahashi, Phys. Rev. B 56, 3212 (1997).
[52] M.S. Makivic and H.-Q. Ding, Phys. Rev. B 43, 3562 (1991).
[53] H.G. Evertz and M. Marcu, Int. J. Mod. Phys. C4, 1147 (1993).
[54] V. A. Kashurnikov, N. V. Prokof'ev, B. V. Svistunov, and M. Troyer, Phys. Rev. B 59, 1162 (1999).
[55] S. Zhang, N. Kawashima, J. Carlson, and J.E. Gubernatis, Phys. Rev. Lett. 74, 1500 (1995).
[56] H. Onishi, M. Nishino, N. Kawashima, and S. Miyashita, J. Phys. Soc. Jpn. 68, 2547 (1999).
[57] O.F. Syljuasen, cond-mat/9907142.
[58] E. Farhi and S. Gutmann, Ann. Phys. (N.Y.) 213, 182 (1992).
[59] M. Aizenman and E.H. Lieb, Phys. Rev. Lett. 65, 1470 (1990).
[60] U.-J. Wiese, Phys. Lett. B311 (1993) 235.
[61] U.-J. Wiese, Nucl. Phys. Proc. Suppl. 73 (1999) 146, and references therein.
[62] N. Prokof'ev and B. Svistunov, Phys. Rev. Lett. 80, 4355 (1998).
[63] A. Sandvik, this volume.
[64] A.W. Sandvik, Phys. Rev. B 59, R14157 (1999).
[65] P. Henelius, A.W. Sandvik, C. Timm, and S. M. Girvin, cond-mat/9903148.
[66] J.K. Kim and M. Troyer, Phys. Rev. Lett. 80, 2705 (1998).
[67] K. Kato, S. Todo, K. Harada, N. Kawashima, S. Miyashita, and H. Takayama, cond-mat/9905379.
[68] B. Frischmuth, B. Ammon, and M. Troyer, Phys. Rev. B 54, R3714 (1996); B. Frischmuth, S. Haas, G. Sierra, and T. M. Rice, Phys. Rev. B 55, R3340 (1997); O.F. Syljuasen, S. Chakravarty, and M. Greven, Phys. Rev. Lett. 78, 4115 (1997).
[69] Y. J. Kim et. al., Phys. Rev. B 60, 3294 (1999).
[70] M. Troyer, M.E. Zhitomirsky, and K. Ueda, Phys. Rev. B 55, R6117 (1997).
[71] B. B. Beard, R. J. Birgeneau, M. Greven, and U.-J. Wiese, Phys. Rev. Lett. 80, 1742 (1998); J.K. Kim and M. Troyer, Phys. Rev. Lett. 80, 2705 (1998).
[72] B.B. Beard, V. Chudnovsky, and P. Keller-Marxer, cond-mat/9910291.
[73] L. Yin, M. Troyer, and S. Chakravarty, Europhys. Lett. 52, 559 (1998); M. A. Korotin, I. S. Elfimov, V. I. Anisimov, M. Troyer, and D. I. Khomskii, Phys. Rev. Lett. 83, 1387 (1999).
[74] B. Frischmuth and M. Sigrist, Phys. Rev. Lett. 79, 147 (1997); B. Frischmuth, M. Sigrist, B. Ammon, and M. Troyer, Phys. Rev. B 60, 3388 (1999); B. Ammon and M. Sigrist, J. Phys. Soc. Jpn. 68, Vol. 3. (1999); M. Nishino, H. Onishi, P. Roos, K. Yamaguchi, and S. Miyashita, cond-mat/9906426.

[75] Y. Iino and M. Imada, J. Phys. Soc. Jpn. 65, 3728 (1996); T. Miyazaki, M. Troyer, M. Ogata, K. Ueda, and D. Yoshioka, J. Phys. Soc. Jpn. 66 (1997) 2580; M. Greven and R. J. Birgeneau, Phys. Rev. Lett. 81, 1945 (1998).
[76] M. Imada and Y. Iino, J. Phys. Soc. Jpn. 66 (1997) 568.
[77] S. Miyahara, M. Troyer, D.C. Johnston, and K. Ueda, J. Phys. Soc. Jpn. 67, 3918 (1998).
[78] M. Troyer and S. Sachdev, Phys. Rev. Lett. 81, 5418 (1998).
[79] T. Ikegami, S. Miyashita, and H. Rieger, J. Phys. Soc. Jpn. 67, 2671 (1998); C. Pich, A. P. Young, H. Rieger, and N. Kawashima, Phys. Rev. Lett. 81, 5916 (1998).
[80] M. Kohno, M. Takahashi, and M. Hagiwara, Phys. Rev. B 57, 1046 (1998).
[81] P. Roos and S. Miyashita, Phys. Rev. B 59, 13782 (1999).
[82] B. Frischmuth, F. Mila, and M. Troyer, Phys. Rev. Lett. 82, 835 (1999); F. Mila, B. Frischmuth, A. Deppeler, and M. Troyer, Phys. Rev. Lett. 82, 3697 (1999).
[83] H. Onishi and S. Miyashita, cond-mat/9912210.
[84] See e.g. *Monte Carlo Methods in Statistical Physics*, 2nd ed., ed. by K. Binder (Springer Berlin, New York 1986).
[85] M.P. Allen and D.J. Tildesley, *Computer Simulations of Liquids* (Oxford University Press, Oxford, 1987), Chap. 6.
[86] P. C. Hohenberg and B. I. Halperin, Rev. Mod. Phys. 49, 435 (1977).
[87] J.G. Propp and D.B. Wilson, Random Structures and Algorithms 9(1&2), 223 (1996). Paper and current annotated bibliography are available at dbwilson.com/exact.
[88] See e.g. C.K. Yang and D.H. Robinson, *Understanding and Learning Statistics by Computer*, World Scientific Series in Computer Sciences, Vol. 4, 1986.