Loop algorithms for quantum simulations of fermion models on lattices

N. Kawashima and J. E. Gubernatis
Center for Nonlinear Studies and Theoretical Division, Los Alamos National Laboratory, Los Alamos, New Mexico 87545

H. G. Evertz
Center for Simulational Physics, Department of Physics, University of Georgia, Athens, Georgia 30602
and Supercomputer Computations Research Institute, Florida State University, Tallahassee, Florida 32306
(Received 2 February 1994; revised manuscript received 7 March 1994)

Two cluster algorithms, based on constructing and flipping loops, are presented for world-line quantum Monte Carlo simulations of fermions and are tested on the one-dimensional repulsive Hubbard model. We call these algorithms the loop-flip and loop-exchange algorithms. For these two algorithms and the standard world-line algorithm, we calculated the autocorrelation times for various physical quantities and found that the ordinary world-line algorithm, which uses only local moves, suffers from very long correlation times that makes not only the estimate of the error difficult but also the estimate of the average values themselves difficult. These difficulties are especially severe in the low-temperature, large-U regime. In contrast, we find that new algorithms, when used alone or in combinations with themselves and the standard algorithm, can have significantly smaller autocorrelation times, in some cases being smaller by three orders of magnitude. The new algorithms, which use nonlocal moves, are discussed from the point of view of a general prescription for developing cluster algorithms. The loop-flip algorithm is also shown to be ergodic and to belong to the grand canonical ensemble. Extensions to other models and higher dimensions are briefly discussed.

I. INTRODUCTION

The world-line quantum Monte Carlo method is frequently used by condensed matter and field theorists to simulate lattice models of systems of interacting fermions, bosons, and quantum spins.[1] This method has become a textbook example[2] of a quantum Monte Carlo method as one of its virtues is its simplicity. Another virtue is its production of world-line patterns that often pictorially represent the imaginary-time quantum dynamics of the model. Several difficulties with the method are also well known.

The most notable difficulty,[3] which the world-line method shares with almost all other quantum Monte Carlo methods, is a sign problem which is manifested by Monte Carlo transition probabilities becoming negative. Typically, this difficulty renders the method ineffective. A difficulty[3] more unique to the world-line method is the lack of ergodicity as in practice the winding number of the world lines is conserved and thereby the sampling of phase space is restricted. The winding number conservation corresponds to the conservation of fermion number and thus places the method in the canonical ensemble.

A less appreciated and infrequently studied difficulty is the world-line method's very long autocorrelation time between measurements of physical quantities. These long times make error estimation for these quantities difficult and can cause long computer runs. The main purpose of this paper is to illustrate the extraordinary lengths these times can take and to present two new ways of implementing the world-line method that in many cases reduce these times by several orders of magnitude. The new methods are a significant improvement in efficiency. One method is also naturally ergodic and grand canonical as winding number is not conserved.

With improved efficiency, the new methods generate significant reductions in the variance of the calculated results for a fixed amount of computing time. Since the sign problem is usually accompanied by a dramatic increase in the variance of the measured quantities, the extensions potentially mitigate the sign problem. They do not, however, address it directly. Oddly, some combinations of the new methods permit a sign problem to occur in simulations for which the standard implementations have no sign problem. We found that this was a minor problem, occurring infrequently only at high temperature in very small systems and decreasing as the lattice size was increased. Constructing still other methods that conserve winding number is possible; however, allowing the winding number to change often appears to be a key ingredient for improved performance.

We will present our new algorithms by discussing their application to world-line simulations of the one-dimensional repulsive Hubbard model. The Hubbard model is one of the simplest models of interacting electrons, and for it we will study the autocorrelation time among measurements of the different physical quantities germane to its interesting physics. In Sec. II, we define the model and give a brief description of the computation of its thermodynamic properties from a path integral. Here, we will also define and discuss how we measured the autocorrelation times. To establish notation and make subsequent discussion reasonably self-contained, we also present the particular implementation of the world-line method used in our studies. This implementation focuses on Monte Carlo moves that change the state of

plaquettes defined on the space-time lattice on which the world lines exist.

In Sec. III, we present our two new world-line methods, which we call the loop-flip and loop-exchange methods. The loop-flip method is an extension of the work of Evertz *et al.*[4] who developed a loop-flip algorithm to reduce critical slowing down in six-vertex model simulations. In that work, they observed that world-line simulations of quantum spin systems have similarities to simulations of the six-vertex model and suggested the potential utility of their algorithm for simulating such systems. We will discuss the extensions of the procedure necessary for simulating fermion systems. We also present our second method, the loop-exchange algorithm. For fermion systems, we found it very useful to exchange portions of up- and down-spin world lines to accelerate the sampling of phase space. This exchange was designed to overcome the difficultly of moving up- and down-spin world lines across one another because of the Coulomb repulsion between the fermions that exists in the Hubbard model. Our two new algorithms, in sharp contrast with the standard implementations of the world-line method, use nonlocal (global) updating moves. These moves generalize the cluster algorithms recently developed to reduce long autocorrelation times accompanying simulations of critical phenomena in classical spin systems.[5] Here, we are not concerned with critical phenomena, but rather we are reducing the inherently long autocorrelation times that occur in the world-line method even when the physical system is far removed from any known phase transition.

We investigate in Sec. IV the effectiveness of various combinations of the algorithms for computing several different properties of the repulsive Hubbard model. For a standard version of the world-line method, which we call the plaquette-flip algorithm, the autocorrelation time was too long to measure in most cases, even with the use of relatively long computer runs. In these cases, reliable error estimation is very difficult. Our new algorithms did not suffer from this problem. Finally, in Sec. V, we summarize our findings and discuss important issues awaiting further investigation, particularly for doing models other than the Hubbard model and for studying models in higher dimensions.

II. BACKGROUND AND DEFINITIONS

A. Hubbard model

The one-dimensional Hubbard Hamiltonian is

$$H \equiv \sum_i H_{i,i+1} = \sum_i [T_{i,i+1} + \tfrac{1}{2}(V_i + V_{i+1})] , \tag{1}$$

where

$$T_{i,i+1} = -t \sum_\sigma (c^\dagger_{i,\sigma} c_{i+1,\sigma} + c^\dagger_{i+1,\sigma} c_{i,\sigma}) \tag{2}$$

and

$$V_i = \sum_\sigma [\tfrac{1}{2} U (n_{i,\sigma} - \tfrac{1}{2})(n_{i,-\sigma} - \tfrac{1}{2}) - \mu (n_{i,\sigma} - \tfrac{1}{2})] . \tag{3}$$

Here, $c^\dagger_{i,\sigma}$ and $c_{i,\sigma}$ are the creation and destruction operators for a fermion at site i with spin σ (up or down) and $n_{i,\sigma} = c^\dagger_{i,\sigma} c_{i,\sigma}$ is the fermion number operator at site i for spin σ. The first term in (1) is the kinetic energy of the electrons and describes their hopping, without spin flip, from site to site. The second term is the potential energy (Coulomb interaction) that exists only if two electrons occupy the same site and includes the chemical potential μ term for convenience. We took $U > 0$. The model is defined in such a way that when μ equals zero, the model exhibits particle-hole symmetry. With this symmetry, there are an equal number of up- and down-spin electrons and their sum equals the number of lattice sites N (the half-filled case). We assume periodic boundary conditions.

At half filling, the system is an insulator, with antiferromagnetic spin fluctuations dominating charge-density-wave fluctuations. At other fillings, it is metallic. Increasing the parameter U suppresses the charge fluctuations and enhances the spin fluctuations. The following correlation functions are useful descriptors of these two types of fluctuations:[6]

$$T\chi_\pm(q) = \frac{1}{\beta N} \int_0^\beta d\tau \sum_{j,k} e^{iq\cdot k} \langle (n_{j+k,\uparrow}(\tau) \pm n_{j+k,\downarrow}(\tau))(n_{j,\uparrow}(0) \pm n_{j,\downarrow}(0)) \rangle . \tag{4}$$

They measure static spin-density-wave (SDW) and charge-density-wave (CDW) correlations. Simpler measurable quantities include the energy $E = \langle H \rangle$ and the average electron occupancy per site,

$$n = \frac{1}{\beta N} \int_0^\beta d\tau \sum_j \langle [n_{j,\uparrow}(\tau) + n_{j,\downarrow}(\tau)] \rangle . \tag{5}$$

As we discuss below, we compute these quantities as functions of imaginary-time τ and their definitions reflect an average over this parameter.

In (4) and (5), the symbol $\langle \cdots \rangle$ denotes the finite-temperature expectation value of some physical observable and is defined by

$$\langle A \rangle = \mathrm{Tr}\, A e^{-\beta H} / Z , \tag{6}$$

where $\beta = 1/kT$ is the inverse temperature, $Z = \mathrm{Tr}\, e^{-\beta H}$ is the partition function of the system, and Tr denotes the trace operation. The basic idea of the world-line method is to express the trace operation as a path integral in imaginary time and then use Monte Carlo techniques to evaluate the resulting multidimensional functional integral numerically.

B. Path integrals

To develop the path integral, we rewrite the partition function of the system by dividing the imaginary-time in-

terval $0 \leq \tau \leq \beta$ into L slices of width $\tau = \beta/L$ and inserting complete sets of states $|s\rangle$ at each time slice:

$$Z = \sum_{\{s_i\}} \langle s_1 | e^{-\tau H} | s_L \rangle \cdots \langle s_3 | e^{-\tau H} | s_2 \rangle \langle s_2 | e^{-\tau H} | s_1 \rangle \ , \tag{7}$$

where the sum is over the set $\{s_i\} = \{s_1, s_2, \ldots, s_L\}$ of all states s_i. The subscript on the identifier s of the different members of the complete set marks the time slice at which the states were inserted. The properties of the trace require the propagation in imaginary time to be periodic.

In order to simplify the evaluation of the matrix elements in (7), we rewrite the full Hamiltonian as the sum of two easily diagonalizable, but not necessarily commuting, pieces, $H = H_1 + H_2$. Using one form of the Suzuki-Trotter approximation,[7] we may then approximate the imaginary-time evolution operator $e^{-\tau H}$ by

$$e^{-\tau H} \approx e^{-\tau H_2} e^{-\tau H_1} \ . \tag{8}$$

When this approximation is used in the expression for the partition function (7) and additional intermediate states are inserted, the resulting expression for the partition function is

$$Z = \sum_{\{s_i\}} \langle s_1 | e^{-\tau H_2} | s_{2L} \rangle \cdots \langle s_4 | e^{-\tau H_1} | s_3 \rangle \langle s_3 | e^{-\tau H_2} | s_2 \rangle \langle s_2 | e^{-\tau H_1} | s_1 \rangle \ . \tag{9}$$

In Sec. II C, we will give specific choices of H_i and $|s\rangle$ for the Hubbard model. Here, we are concerned only with the general framework of the method.

We now see that the partition function (9) may be re-expressed as a functional integral over all possible configurations $\mathcal{C} \equiv \{s_i\}$. To evaluate the path integral, Monte Carlo importance sampling is used. This method generates a sequence of configurations $\mathcal{C}_i$ (see below), which, in the infinite-sequence limit, conforms to the probability distribution

$$\begin{aligned} P(\mathcal{C}) &= \langle s_1 | e^{-\tau H_2} | s_{2L} \rangle \cdots \langle s_3 | e^{-\tau H_2} | s_2 \rangle \langle s_2 | e^{-\tau H_1} | s_1 \rangle / Z \\ &\equiv W(\mathcal{C})/Z \ . \end{aligned} \tag{10}$$

Such a sequence may be generated by beginning with an arbitrary (allowed) world-line configuration $\mathcal{C}_0$ and then considering some modified configuration $\mathcal{C}_0'$. One then computes the probabilities for these configurations, $P(\mathcal{C}_0)$ and $P(\mathcal{C}_0')$, and accepts the modified configuration with probability R for which we choose

$$R = \frac{P(\mathcal{C}_0')}{P(\mathcal{C}_0) + P(\mathcal{C}_0')} = \frac{W(\mathcal{C}_0')}{W(\mathcal{C}_0) + W(\mathcal{C}_0')} \ . \tag{11}$$

If the modification is accepted, the new configuration will be $\mathcal{C}_1 = \mathcal{C}_0'$; otherwise the old configurations will be retained: $\mathcal{C}_1 = \mathcal{C}_0$. The procedure is then iterated to produce $\mathcal{C}_2$, $\mathcal{C}_3$, etc.

C. World lines

For fermion lattice models, the standard choice for the complete set of states $|s\rangle$ is the occupation number basis formed from the state

$$|n_{1,\uparrow}, n_{2,\uparrow}, \ldots, n_{n,\uparrow}\rangle |n_{1,\downarrow}, n_{2,\downarrow}, \ldots, n_{N,\downarrow}\rangle$$

by allowing each lattice site to assume all possible occupancies of up and down electrons.[3] The splitting of the Hamiltonian $H = H_1 + H_2$ is usually done by having H_1 and H_2 refer to the odd and even lattice sites.[3] With these choices, the product of the matrix elements in the expression (9) for the partition function factorizes into products of matrix elements defined on shaded squares (plaquettes) of up- and down-spin checkerboards. The summation over all states becomes a summation over all combinations of occupancies $n_{i,l}^{\sigma}$ for each spin. Here, i labels the spatial (real-space) position and l, the temporal (imaginary-time) position. A shaded plaquette for a given spin is one with the four sites (i,l), $(i+1,l)$, $(i,l+1)$, and $(i+1,l+1)$ where i and l are either both even or both odd. A world line for a given spin is a continuous line constructed from straight line segments connecting occupied sites. A one-to-one correspondence between fermions and world lines can be made if two parallel vertical (temporal) lines are assigned to a shaded plaquette with four occupied sites. These concepts are illustrated in Fig. 1 which shows the checkerboard and a typical world-line configuration for one spin component of a system of four electrons on an eight-site lattice.

Because of the checkerboarding, we can write the partition function as

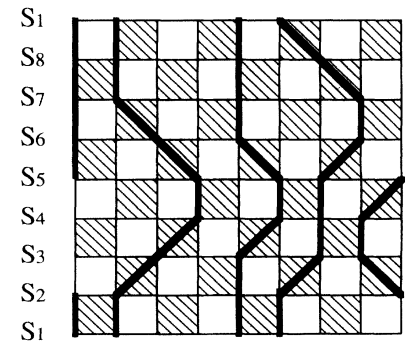

FIG. 1. An example of the space-time checkerboard for a single component of spin. Here, four electrons of the same spin are on eight lattice sites, which are along the horizontal direction. In the vertical direction are $2L = 8$ imaginary-time steps. The symbols S_i along the left edge designate the many-body state at that imaginary-time step. The ket $|S_2\rangle$ represents $|1,1,0,0,1,1,0,0\rangle$, for example.

$$Z=\sum_{\{n_{i,l}^{\sigma}\}}\operatorname{sgn}(\{n_{i,l}^{\sigma}\})W(\{n_{i,l}^{\sigma}\}) , \tag{12}$$

$$W(\{n_{i,l}^{\sigma}\})=\prod_{l=1}^{L}\prod_{i=1}^{N/2} w\begin{bmatrix} n^{\uparrow}_{2i-1,2l} & n^{\uparrow}_{2i,2l} & \vert & n^{\downarrow}_{2i-1,2l} & n^{\downarrow}_{2i,2l} \\ n^{\uparrow}_{2i-1,2l-1} & n^{\uparrow}_{2i,2l-1} & \vert & n^{\downarrow}_{2i-1,2l-1} & n^{\downarrow}_{2i,2l-1}\end{bmatrix} \times w\begin{bmatrix} n^{\uparrow}_{2i,2l+1} & n^{\uparrow}_{2i+1,2l+1} & \vert & n^{\downarrow}_{2i,2l+1} & n^{\downarrow}_{2i+1,2l+1} \\ n^{\uparrow}_{2i,2l} & n^{\uparrow}_{2i+1,2l} & \vert & n^{\downarrow}_{2i,2l} & n^{\downarrow}_{2i+1,2l}\end{bmatrix} . \tag{13}$$

The arguments of the functions on the right-hand side are written in a form that emphasizes the plaquette structure for both up and down spins. Clearly, the weight (13) reduces to a product of weights for only the shaded plaquettes. If we further choose

$$e^{-\tau H_{i,i+1}}\approx e^{-\tau(V_i+V_{i+1})/4}e^{-\tau T_{i,i+1}}e^{-\tau(V_i+V_{i+1})/4} \tag{14}$$

then we can express the local weights as

$$w\begin{bmatrix} n_1^{\uparrow}n_2^{\uparrow} & \vert & n_1^{\downarrow}n_2^{\downarrow} \\ n_3^{\uparrow}n_4^{\uparrow} & \vert & n_3^{\downarrow}n_4^{\downarrow}\end{bmatrix}\equiv u\begin{bmatrix} n_1^{\uparrow}n_2^{\uparrow} \\ n_3^{\uparrow}n_4^{\uparrow}\end{bmatrix}u\begin{bmatrix} n_1^{\downarrow}n_2^{\downarrow} \\ n_3^{\downarrow}n_4^{\downarrow}\end{bmatrix}v\begin{bmatrix} n_1^{\uparrow}n_2^{\uparrow} & \vert & n_1^{\downarrow}n_2^{\downarrow} \\ n_3^{\uparrow}n_4^{\uparrow} & \vert & n_3^{\downarrow}n_4^{\downarrow}\end{bmatrix} \tag{15}$$

with

$$u\begin{bmatrix} n_1n_2 \\ n_3n_4\end{bmatrix}\equiv\delta_{n_1,n_2}\delta_{n_3,n_4}\delta_{n_1,n_3}+\cosh(\tau t)\delta_{n_1,1-n_2}\delta_{n_3,1-n_4}\delta_{n_1,n_3}+\sinh(\tau t)\delta_{n_1,1-n_2}\delta_{n_3,1-n_4}\delta_{n_1,1-n_3} \tag{16}$$

and

$$v\begin{bmatrix} n_1^{\uparrow}n_2^{\uparrow} & \vert & n_1^{\downarrow}n_2^{\downarrow} \\ n_3^{\uparrow}n_4^{\uparrow} & \vert & n_3^{\downarrow}n_4^{\downarrow}\end{bmatrix}=\prod_{i=1}^{4}v_1(n_i^{\uparrow},n_i^{\downarrow}) , \tag{17}$$

where

$$v_1(n,n')\equiv e^{-(\tau/4)[U(n-1/2)(n'-1/2)-\mu(n+n'-1)]} . \tag{18}$$

In (16), we see the kinetic-energy contribution to the weight is nonvanishing only for certain sets of occupancies. In fact, out of the $2^4=16$ possible plaquettes for either an up or a down spin, only 6 plaquettes satisfy local fermion conservation and these are shown in Fig. 2.

The manifold on which the world lines are defined is a two-dimensional torus since we have a periodic boundary condition for both spatial and temporal directions. Accordingly, on this torus, we can define spatial and temporal winding numbers for any closed loop, for example, a world line. To be more specific, to define a spatial winding number for a given loop, let us first suppose we have a spatial cut, i.e., a vertical line in Fig. 1 which starts somewhere at the bottom and ends at the top of the checkerboard. (The location of the cut does not matter since it does not affect the definition.) Then, if we trace the entire path of the loop and it passes a spatial cut in one direction m times and in the opposite direction n times, then the spatial winding number of this loop is defined to be $m-n$. The total spatial winding number of world lines is defined as the sum of the spatial winding numbers of all the world lines. A temporal winding number of a given loop and the total temporal winding number of world lines are defined in the same fashion but with the temporal cut being a horizontal line in the checkerboard. In what follows, when we say *spatial or temporal winding number,* we will mean the total spatial or temporal winding number of the world lines.

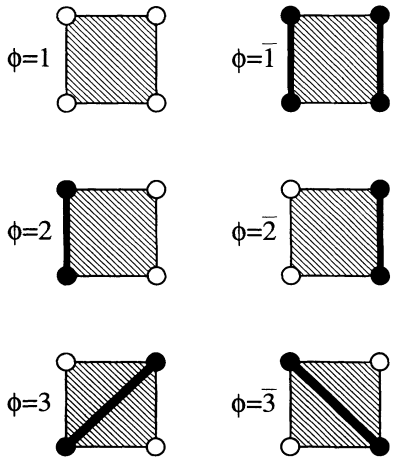

FIG. 2. The six allowed shaded plaquettes. The values of ϕ are our symbols for the depicted plaquette.

The sign in (12) is defined in terms of the world lines

$$\operatorname{sgn}(\{n_{i,l}^{\sigma}\})\equiv(-1)^{\sum_l(Z_l-1)} , \tag{19}$$

where Z_l is the temporal winding number of the lth world line and the sum is taken over all the world lines. Its origin is the fermion anticommutation. For the one-dimensional Hubbard model, the standard world-line algorithms have no sign problem as $Z_l=1$ and is conserved. For the new algorithms we will be proposing this will not be the case. Whenever a configuration with a negative sign occurred in our simulations, we treated it in the ordinary way;[8] that is, we computed thermodynamic averages by

$$\langle A\rangle=\sum_{\mathcal{C}}\operatorname{sgn}(\mathcal{C})A(\mathcal{C})/\sum_{\mathcal{C}}\operatorname{sgn}(\mathcal{C}) . \tag{20}$$

By using this formula, we do the Monte Carlo simulation with the following weight:

$$Z' = \sum_{\{n_{i,l}^{\sigma}\}} W(\{n_{i,l}^{\sigma}\}) , \tag{21}$$

that ignores the sign factor.

D. The plaquette-flip algorithm (algorithm *P*)

The world-line method can be implemented in several different ways. We will now highlight the main points of an implementation which focuses on Monte Carlo moves that change the local fermion occupancies of plaquettes defined on a checkerboarding of imaginary time and space. This implementation leads to efficient algorithms not only on serial computers but also on vector and parallel machines.[9-11]

To generate different sequences of world lines, one uses the observation[3] that only one basic movement of a world-line segment can occur: If an unshaded plaquette has a world-line segment on one vertical edge and none on the other, the allowed movement is vacating the one edge and occupying the other by moving the world-line segment across the unshaded plaquette. This movement is illustrated in Fig. 3. Algorithm *P* accomplishes this by visiting in sequence each unshaded plaquette for both the up- and down-spin checkerboards and attempting a local move that flips all four corners of the unshaded plaquette. *Flipping a corner* means replacing the value of the variable $n_{i,l}$ by $1-n_{i,l}$. The configuration weights $W(\mathcal{C})$ and $W(\mathcal{C}')$ are determined by computing the product of the weights (15) for the four neighboring shaded plaquettes for both the old and flipped configuration. These weights are then used in (11) to determine whether the flip is accepted or not. An important point is that we need to know only local states (the occupancies of the shaded plaquettes surrounding the unshaded one) to calculate this probability rather than the state of the whole system. This is the case because the attempted flip changes the configuration only locally since the Boltzmann weight (15) is factorized into a product of local weights.

In algorithm *P*, the winding number of any world line in any direction, temporal or spatial, is conserved. We found, however, that this broken ergodicity, as expected,[3] has only a small effect on the simulation, especially for larger lattices.

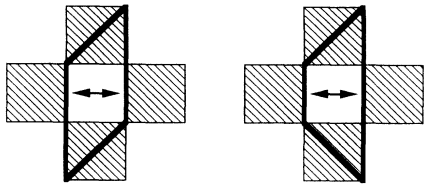

FIG. 3. The allowed movements of world-line segments back and forth across a unshaded plaquette.

E. Autocorrelation times

The basic property of the Monte Carlo method is to replace the thermodynamic average (6) by the average of $A(t)$ over M Monte Carlo steps:

$$\langle A \rangle \approx \overline{A} \equiv \frac{1}{M} \sum_{t=1}^{M} A(t) , \tag{22}$$

where $A(t)$ is the value of a physical quantity A (energy, SDW correlations, etc.) at Monte Carlo step t. If M is large enough and the $A(t)$'s are *statistically independent* estimates of A, then the error estimate for A would be $\sigma/\sqrt{M}$ where

$$\sigma^2 = \frac{1}{M-1} \sum_{t=1}^{M} [A(t) - \overline{A}]^2 . \tag{23}$$

To measure the degree of statistical correlations, we calculated two kinds of quantities. The first one is an integrated autocorrelation time defined by

$$\tau_{\text{int}}^{A} \equiv -\tfrac{1}{2} + \sum_{t=0}^{\infty} \Gamma_A(t) , \tag{24}$$

where

$$\Gamma_A(t) \equiv \frac{\langle A(t_0+t)A(t_0)\rangle - \langle A(t_0)\rangle\langle A(t_0)\rangle}{\langle A(t_0)A(t_0)\rangle - \langle A(t_0)\rangle\langle A(t_0)\rangle} . \tag{25}$$

In the actual simulation, $\Gamma_A(t)$ is approximated by

$$\Gamma_A(t) \approx \frac{\frac{1}{M-t}\sum_{i=1}^{M-t} A(i+t)A(i) - \frac{1}{(M-t)^2}\left[\sum_{i=1}^{M-t} A(i)\right]\left[\sum_{i=t+1}^{M} A(i)\right]}{\frac{1}{M}\sum_{i=1}^{M}[A(i)A(i) - \overline{A}\,\overline{A}]} . \tag{26}$$

The variance σ_{int}^2 of data correlated with the time τ_{int}^{A} is related to the variance σ^2 computed from (23) in the absence of correlations by[12]

$$\sigma_{\text{int}}^2 \approx 2\tau_{\text{int}}^{A}\sigma^2 . \tag{27}$$

Thus, in the presence of correlations, $2\tau_{\text{int}}^{A}$ more simulation steps are needed in order to achieve the same variance of a measured quantity. Accordingly, error estimates computed when correlations are unknowingly present are always smaller than the actual error.

Typically, the autocorrelation function is not a simple exponential function in time. This fact sometimes makes estimating τ_{int}^{A} by (24) difficult. Another estimate of τ_{int}^{A} is obtained by examining what it takes to produce statistically independent measurements. We grouped the M measurements into n bins of length $l = M/n$, and for a sequence of bin lengths (chosen to be 2,4,8, . . .), we computed the bin averages $\overline{A}_b(l)$ of the $A(t)$

$$\overline{A}_b(l)\equiv\frac{1}{l}\sum_{t=(b-1)l+1}^{bl}A(t)\ , \tag{28}$$

and the variance of these averages

$$\sigma(l)^2\equiv\frac{1}{n-1}\sum_{b=1}^{n}[\overline{A}_b(l)-\overline{A}]^2\ . \tag{29}$$

This variance should become inversely proportional to l as the bin size l becomes large enough so that the $\overline{A}_b(l)$ as a function of b becomes statistically independent.[13] When statistical independence is approached, the quantity

$$\tau_{\rm int}^{A}(l)=\frac{l\sigma(l)^2}{2\sigma^2}\ , \tag{30}$$

where σ^2 is computed from (23), approaches a constant as a function of l.[13] This constant is our estimate for $\tau_{\rm int}^{A}$. The error estimate becomes $\sigma(l^*)/\sqrt{n}$ where l^* is the value of l at which the constant is reached. We note that with the use of (30), we can rewrite this error estimate as $\sigma\sqrt{2\tau_{\rm int}/M}$ and obtain the expected result (27). We also note that we have defined $\tau_{\rm int}^{A}(l)$ so that $\tau_{\rm int}^{A}(1)=\frac{1}{2}$.

Generally, $\tau_{\rm int}^{A}(l)$ is an increasing function of l. This increase, however, ceases when l becomes much larger than the autocorrelation time $\tau_{\rm int}^{A}$. As we discuss below, we found cases where, in spite of a very long computer run, $\tau_{\rm int}^{A}(l)$ keeps increasing as function of l. In such cases, we estimated a lower bound for $\tau_{\rm int}^{A}$ by selecting the value $\tau_{\rm int}^{A}(l)$ for the largest available l.

A different measure of autocorrelation time we used is the exponential autocorrelation time $\tau_{\rm expt}$ which by definition is the largest autocorrelation time associated with a given algorithm. We determined $\tau_{\rm expt}$ from the $t\rightarrow\infty$ asymptotic behavior of $\Gamma_A(t)$,

$$\Gamma_A(t)\sim\gamma_A\exp(-t/\tau_{\rm exp})\ , \tag{31}$$

by fitting a straight line to $\ln\Gamma_A(t)$. We remark that, in practice, the most stable estimates of $\tau_{\rm int}^{A}$ were obtained by summing $\Gamma_A(t)$ in (24) only up to some finite $t=W$, with W of the order of $\tau_{\rm exp}$, and computing the contribution of $t>W$ from (31).

The autocorrelation time $\tau_{\rm exp}$ corresponds to the second largest eigenvalue of the transition probability matrix $W(\mathcal{C}\rightarrow\mathcal{C}')$ for changing configuration $\mathcal{C}$ to $\mathcal{C}'$. (The largest eigenvalue is one, with the Boltzmann distribution as the eigenvector.[14]) This correlation time depends on the algorithm. The constant γ_A in (31) depends on the observable A (as well as on lattice size, temperature, etc.) and can be very small or zero for some observables. In general, $\tau_{\rm exp}$ is more difficult to estimate precisely than $\tau_{\rm int}^{A}$. In order to determine accurately $\tau_{\rm exp}$, we must therefore use a suitable range of observables to establish consistency. In some cases, we could not obtain reliable estimates of $\tau_{\rm exp}$ as we will see below. Another remark about $\tau_{\rm exp}$ is that the longest-lived physical mode associated with it can be a mode whose contributions to the quantities of interest are negligible. We will discuss this point more fully later.

III. NEW ALGORITHMS

A. The loop-flip algorithm (algorithm *L*)

We will discuss the loop-flip algorithm (algorithm L). In contrast to algorithm P, which makes only local changes in configuration space based on local decisions, the loop-flip algorithm makes nonlocal moves based on local decisions that are linked in a specific manner. The manner chosen is related to the loop-flip algorithm recently proposed by Evertz *et al.*[4] to reduce critical slowing down in simulations of the six-vertex model. The formal connection of the six-vertex model with the fermion (and quantum spin) problem is through the shaded plaquettes for each spin assuming only 6 out of the 16 possible configurations. The main conceptual connection is through the recognition that the world-line and the six-vertex configurations[15] can be parametrized by closed loops as a consequence of a zero-divergence property of the models. In the fermion case, this property is directly connected with the local conservation of fermions. From the loop point of view, the difference between two different world-line configurations must simply be one or more closed loops as the difference must also satisfy the zero-divergence condition. It is such differences that the loop-flip algorithm constructs.

The core of our loop-flip algorithm is the "massless" case of the six-vertex algorithm proposed by Evertz *et al.*[16] which uses only "break-up" operations. For each spin (up or down), the six allowed plaquettes are mapped onto three new plaquettes that have lines drawn on them. Since each lattice site belongs to two shaded plaquettes, these lines (loop segments) are drawn so that each site is touched by one line. When these lines are connected, loops form, some of which are very long and wind one or more times around the temporal and spatial directions. For each spin, flipping the loop corresponds to changing electrons on the loop into holes and vice versa. In the process, the number of electrons can change. This change occurs when the net temporal winding number associated with the system changes. If the temporal winding number changes, a "sign problem" may occur (19).

The existence of just six allowable plaquettes for each spin does not mean the Hubbard model is equivalent to a combination of six-vertex models. Several special considerations exist. In (13), the presence of the sign term is one. As we mentioned above, our use of the loop-flip algorithm can generate negative weights; fortunately, we found them for only uninteresting cases and then very rarely. A second consideration is the U term and μ term in the Hubbard model. We take these terms into account by observing that in their absence the weight (15) factors into an up and down term, each of which maps onto a six-vertex model. We apply our loop-flip algorithm to each of these pieces and then include the v contribution by modifying the acceptance probability so that the detailed balance condition is satisfied. This modification is straightforward; our main task is specifying the loop-flip algorithm.

To construct our algorithm, we first imagine that we stack the up- and down-spin checkerboards slightly above one another and focus our attention on blocks whose

upper and lower faces are the up- and down-spin shaded plaquettes. (We have 36 allowed local configurations for each block.) Next, we define the variable ϕ_b which species the state of a block b on which the eight-body local interaction in (15) is defined. For this variable, we take a pair of symbols, $\phi_b \equiv (\phi_b^{\uparrow}, \phi_b^{\downarrow})$, each of which has one of the six allowed values of the plaquette. We symbolize these six values by 1, $\bar{1}$, 2, $\bar{2}$, 3, and $\bar{3}$ and define them in Fig. 2. We then rewrite the Boltzmann weight (15) in terms of local variables defined on the shaded blocks as follows:

$$W(\mathcal{C}) \equiv \prod_b w(\phi_b) , \tag{32}$$

$$w(\phi_b) = u(\phi_b^{\downarrow}) u(\phi_b^{\uparrow}) v(\phi_b^{\downarrow}, \phi_b^{\uparrow}) . \tag{33}$$

Our task is to generate a Markov process, in which different configurations are visited with a frequency proportional to the weight (33). This can be done by cycling a Monte Carlo updating procedure that consists of two steps:[5] The first step is the stochastic assignment of a label to every shaded block, and the second step is a Monte Carlo move using modified weights associated with these labels.

In the first step, we assign a label ψ_b to each shaded block with a probability $p(\psi_b|\phi_b)$. Here, the label ψ_b is a pair of integers, $(\psi_b^{\downarrow}, \psi_b^{\uparrow})$, each of which may have a value 1, 2, or 3. These labels will be related to decompositions of a plaquette into line segments. (See Fig. 4.) The probabilities are to be determined so that the detailed balance condition holds for the overall updating procedure and the sum rule

$$\sum_{\psi_b} p(\psi_b|\phi_b) = 1 \tag{34}$$

is satisfied. Generally, this assignment is an overdetermined problem, but in the present problem, we can assign the probabilities uniquely, as we now discuss.

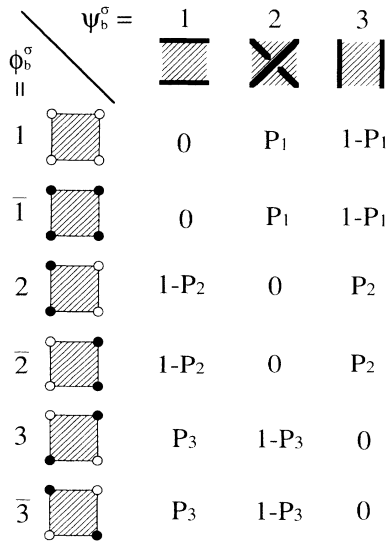

FIG. 4. The labeling probability for the algorithm L. In the top row, the correspondence between labels and loop decompositions is shown. In the leftmost column, the correspondence between symbols (1, $\bar{1}$, 2, $\bar{2}$, 3, and $\bar{3}$) and states are shown. Solid circles stand for particles while open circles for holes.

To define these probabilities, we first define a new local weight

$$\tilde{w}_{\psi_b}(\phi_b) \equiv c_{\psi_b} w(\phi_b) p(\psi_b|\phi_b) , \tag{35}$$

where c_{ψ_b} is a configuration independent renormalization constant which is to be determined so that (34) holds. In the second step of the algorithm, the stochastic updating of states, a Monte Carlo step is performed using (35) instead of (33). As long as the relationship (35) holds, this step together with the first step constitutes a single Monte Carlo move which satisfies the detailed balance condition.[17]

Following the viewpoint of Kawashima and Gubernatis,[17] the algorithm is characterized by the modified weights $\tilde{w}_{\psi_b}(\phi_b)$ since once we specify them, the probabilistic assignment of the labels, i.e., $p(\psi_b|\phi_b)$, is determined by (34) and (35). To show this, we define

$$\tilde{w}_{\psi_b}(\phi_b) = \tilde{u}_{\psi_b^{\uparrow}}(\phi_b^{\uparrow}) \tilde{u}_{\psi_b^{\downarrow}}(\phi_b^{\downarrow}) v(\phi_b^{\uparrow}, \phi_b^{\downarrow}) , \tag{36}$$

where $v(\phi_b^{\downarrow}, \phi_b^{\uparrow})$ is the same as in (33) and

$$\tilde{u}_x(y) = \begin{cases} 0 & \text{if } x = y \text{ or } x = \bar{y}, \\ 1 & \text{otherwise} . \end{cases} \tag{37}$$

The probability for the labeling procedure can then be written as

$$p(\psi_b|\phi_b) = p_P(\psi_b^{\downarrow}|\phi_b^{\downarrow}) p_P(\psi_b^{\uparrow}|\phi_b^{\uparrow}) . \tag{38}$$

For a given spin, the labeling probability p_P of one plaquette is easily found to be

$$\begin{aligned} &p_P(2|1) = 1 - p_P(3|1) = p_P(2|\bar{1}) = 1 - p_P(3|\bar{1}) = P_1 , \\ &p_P(3|2) = 1 - p_P(1|2) = p_P(3|\bar{2}) = 1 - p_P(1|\bar{2}) = P_2 , \\ &p_P(1|3) = 1 - p_P(2|3) = p_P(1|\bar{3}) = 1 - p_P(2|\bar{3}) = P_3 , \end{aligned} \tag{39}$$

where

$$P_1 = \tfrac{1}{2}(1 - e^{-\tau t}), \quad P_2 = \frac{1 + e^{\tau t}}{1 + e^{2\tau t}}, \quad P_3 = \frac{1 - e^{\tau t}}{1 - e^{-2\tau t}} \tag{40}$$

In order to make the simulation effective, it is desirable to have an updating rule which guarantees that any resulting configuration is an allowed state. This is easy to do in the present case, since the state which can be reached by flipping a loop configuration is an allowed one. Actually, the modified weights (37) are chosen so that this correspondence holds. In order to explain what loop configurations are, we define how we construct a loop: For each shaded plaquette, we connect each corner site to another corner by a segment as depicted in Fig. 4. The connected sites form loops. We note that any configuration reached by flipping a loop is allowed since a flip of any segment of a loop results in an allowed local configuration. *Flipping a loop* means replacing all the electrons on the loop by holes and vice versa.

If the v term in (35) were not present, we would simply flip each loop independently with probability $\frac{1}{2}$. In our

case, with the v term present, each loop is flipped with a probability

$$R \equiv 1/(1+e^{-\tau\Delta}) , \tag{41}$$

where

$$\Delta \equiv \frac{U}{2}(2\mathcal{M}-\mathcal{L}) - \frac{\mu}{2}(2\mathcal{N}-\mathcal{L}) , \tag{42}$$

where $\mathcal{N}$ is the total number of particles on the loop, $\mathcal{M}$ is the total number of sites on which there are zero or two electrons, and $\mathcal{L}$ is the length of the loop. We remark that the flip of one loop can change the value of Δ for the other loops. We, therefore, flip loops sequentially, one after the other. For $N=2$ and $L=4$, an example of a loop flip is shown in Fig. 5.

We remark that any change of the world lines by algorithm P can be realized by algorithm L with a finite probability. Additionally, any winding number can occur, and thus the algorithm simulates the grand-canonical ensemble of the model. We can prove ergodicity and even a stronger statement that any state which does not violate local particle-number conservation can be reached with a finite probability from any other such state in one sweep Monte Carlo updating. These statements are a consequence of a loop flip corresponding to changing the occupancy on all sites connected by the loop. Conversely, the difference between a pair of allowed world-line configurations corresponds exactly to a loop configuration. Thus, any two allowed world-line configurations can be transformed in to each other by a single flip of a loop configuration.

B. The loop-exchange algorithm (algorithm L_{ex})

The loop-flip algorithm described above was an extension of the algorithm that Evertz *et al.* applied to the six-vertex model; however, for the Hubbard model it has shortcomings which do not exist for the six-vertex model or a quantum spin system. In particular, when U is large compared to T, long loops have little chance of being flipped.[11] We can understand this behavior by observing that long loops decrease the acceptance probability (42) when $U>0$. Since we are often interested in the model with large positive U and small T, this can be problematic as it can lead to long autocorrelation times. In addition, we are also interested in non-half-filled cases. For them, we also face a similar difficulty of long correlation times because flipping a loop may change the total number of particles, and the acceptance ratio (41) may therefore be strongly suppressed by the μ term in (42).

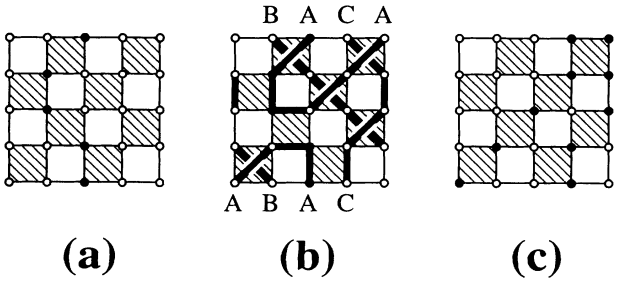

FIG. 5. An example of a loop flip: (a) a configuration with a single fermion, (b) a possible decomposition resulting in three loops, and (c) a possible world-line configuration resulting from the loop flips. Depicted in (c) is a two-fermion configuration with a temporal winding number of two and a spatial winding number of one. Between (a) and (c), a sign change occurred.

In order to resolve these difficulties, we developed a new loop algorithm, the loop-exchange algorithm (algorithm L_{ex}). In this algorithm, two loops which have the same shape, but different spins, are flipped simultaneously. In other words, these flips do not change the U term nor the μ term in (42). Inherently, this algorithm is nonergodic so it must be used with some ergodic algorithm to construct a correct Markov process. In the present paper, we use algorithm L for this purpose.

We will discuss algorithm L_{ex} as a modification of algorithm L. In algorithm L_{ex}, all 36 configurations of a block, *except* two, map uniquely onto a smaller set of labels. The rule for label assignment is straightforward: First, we draw world lines by following the ordinary rule described in Sec. II C. Then, we overlap two plaquettes and erase all doubly drawn lines, i.e., world-line segments drawn at the sample place on both the plaquettes. If the resulting picture has lines that deadend at opposite corners of the same plaquette, these corners are connected by a straight line. After doing this for all blocks, we identify the resulting sets of overlapping world-line segments with labels and assign these labels with a probability of one. This labeling is illustrated in Fig. 6.

The two exceptional cases are those for which the rule of the label assignment is not unique. These cases are configurations $(2,\bar{2})$ and $(\bar{2},2)$. Each can have one of two labels, which we denote by 1 and 2. Their modified weights are defined as

$$\tilde{w}_1[(2,\bar{2})]=\tilde{w}_1[(\bar{2},2)]=\tilde{w}_2[(2,\bar{2})]=\tilde{w}_2[(\bar{2},2)]=1 \tag{43}$$

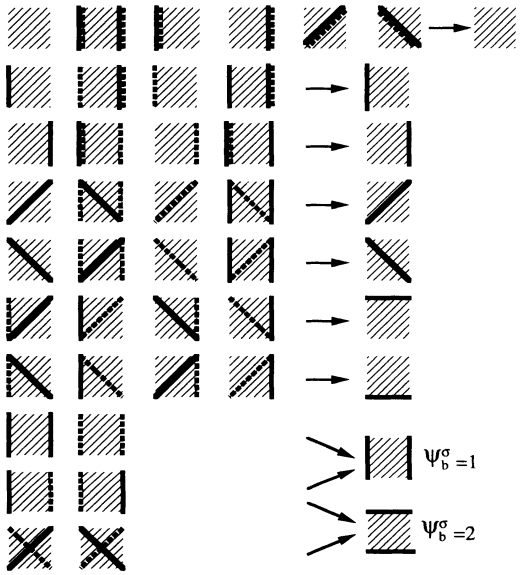

FIG. 6. Correspondence between local states and loop segments in the algorithm L_{ex}. The states are depicted in terms of world-line segments. Solid lines are for down-spin particles and dashed lines are for up-spin particles.

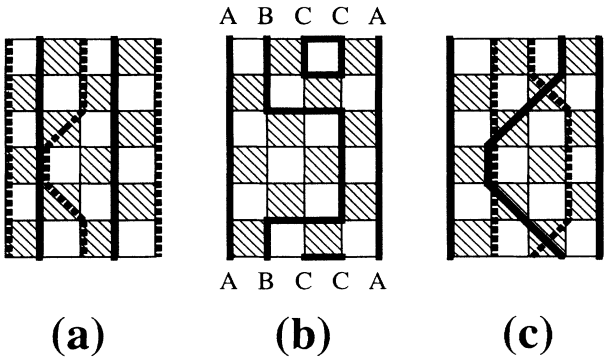

FIG. 7. An example of a loop exchange.

with all other modified weights, i.e., those $\tilde{w}_i$ with the two exceptional labels but with i not equal to 1 or 2, vanishing.

With the specification of all other modified weights as unity, the definition of the loop-exchange algorithm is now complete. We also note that once the labels are chosen, the modified weights for all allowed loop configurations are equal and the transition probability of a Monte Carlo attempt is one-half for any loop, no matter how large U and μ may be. In terms of labeling probabilities, the algorithm $L_{\rm ex}$ is characterized by

$$p[1|(2,\bar{2})]=p[1|(\bar{2},2)]=1-\tanh^2\tau t\ , \tag{44}$$

$$p[2|(2,\bar{2})]=p[2|(\bar{2},2)]=\tanh^2\tau t\ . \tag{45}$$

For the other configurations, the labeling probability is 1 or 0 as described above.

For algorithm $L_{\rm ex}$, only an allowed state can be reached from a flipped loop. Both the up and down planes are treated simultaneously, whereas in algorithm L, they are treated independently as far as the loop construction is concerned. In algorithm $L_{\rm ex}$, therefore, only one set of loops exists for the two planes. The construction rule for the loops is simple: All we do is just connect the segments defined above, which have a one-to-one correspondence to the labels. This rule gives connected loops; no open curves appear. A useful way to view this is as follows: The plaquettes in the right column of Fig. 6 are the result of an exclusive-or (XOR) operation on the overlaid plaquettes along the rows to the left. This XOR operation replaces the overlaid world lines by a lattice with 0's of 1's at the lattice sites. The $L_{\rm ex}$ loops connect the sites with the 1's, of which there are 0, 2, or 4 per shaded plaquette. An example of a loop exchange is shown in Fig. 7 when $N=4$ and $L=3$.

IV. RESULTS

In this section, we present our numerical results for the integrated autocorrelation times (30) associated with the average energy, electron occupancy (5), and the $q=\pi$, static charge-density T_{χ_+} and spin-density T_{χ_-} correlations (4). We will denote these times by $\tau_{\rm int}^E$, $\tau_{\rm int}^N$, $\tau_{\rm int}^C$, and $\tau_{\rm int}^S$. We will also present results for $\tau_{\rm exp}$.

In what follows, we will be concerned with various combinations of the three algorithms described in the previous sections. One case is the plaquette algorithm P by itself. Another is the combination of P and L, which we will refer to as PL. Other symbols, such as L, $LL_{\rm ex}$, and $PLL_{\rm ex}$, should be understood in a similar fashion. The application of any combination which includes L results naturally in a grand-canonical and ergodic simulation, while the application P and $L_{\rm ex}$ alone or together results in a canonical and nonergodic simulation. Although we can force algorithm L to simulate the canonical ensemble, we did not do this unless otherwise stated.

As we mentioned above, in some cases, especially when algorithm P was used alone, we were unable to estimate $\tau_{\rm int}^A$ because the $\tau_{\rm int}^A(l)$ versus l curve did not reach a plateau during the course of our rather lengthy Monte Carlo runs. In such cases, we took the largest available value (with still a reasonably large number of bins remaining) as a practical lower bound on $\tau_{\rm int}^A$. This choice of a lower bound is justified since all the $\tau_{\rm int}^A(l)$ versus l curves we computed, whether they reached the plateau or not, were nondecreasing functions. In Fig. 8, we show $\tau_{\rm int}^A(l)$ versus l curves for a typical example where using algorithm P, we were forced to take a lower bound, and where using algorithm $PLL_{\rm ex}$, we reached a plateau. This figure also illustrates that in some cases we can reduce the autocorrelation time dramatically by combining P with L and $L_{\rm ex}$.

For most simulations, the length of the Monte Carlo calculation was 0.25 million Monte Carlo steps (MCS). In some exceptional cases, where the autocorrelation times are very long, we performed longer runs.[18] In algorithm P, a Monte Carlo step consists of a sweep across all unshaded plaquettes on which an attempt to flip its corners is made. In algorithms L and $L_{\rm ex}$, a Monte Carlo step means decomposing the whole system into loops and attempting to flip every loop. When several algorithms are combined, a MCS consists of one sweep through each algorithm. For example, in PL, one Monte Carlo step is one sweep through P and one sweep through L.

In Fig. 9, various autocorrelation times for P and $PLL_{\rm ex}$ are plotted for fixed value of $y\equiv U\beta/L=0.5$, where L is the lattice extension in imaginary time direction. From Figs. 9(a)–9(c), we see that for our new algorithm $PLL_{\rm ex}$, the autocorrelation time is very small, of order one, in almost all cases. After some initial increase, it is flat as a function of $1/T$ and also as a function of U. Thus, algorithm $PLL_{\rm ex}$ performs very well.

The behavior of the algorithm P, the conventional algorithm, is not as clear. At small β, where we have performed very long simulations, we see that $\tau_{\rm int}$ increases rapidly as a function of β. We also noticed that as a function of U, $\tau_{\rm int}$ increases rapidly. At larger β, most of our (fairly short) runs with algorithm P did not converge, and we show only lower bounds for $\tau_{\rm int}$. It is possible that the initial increase of $\tau_{\rm int}$ with β actually continues towards larger β. Therefore, some of the lower bounds on $\tau_{\rm int}$ could possibly be an order of magnitude below the actual value.

In short, comparing P and $PLL_{\rm ex}$, we see that at large

U and small β, the difference in performance between these algorithms is enormous, being more than 3 orders of magnitude at $U=8$. As argued above, there could be an even bigger difference in performance when β becomes large.

Figure 9 also shows that our PLL_{ex} algorithm is very effective in reducing the integrated correlation time for the SDW correlation function. The reason for this is that we can exchange world lines for an up-spin and a down-spin electron in either algorithm L and L_{ex} without visiting intermediate states with small weights; however, algorithm L_{ex} is most effective in this regard as it was designed to do precisely this. At large U and at low temperature, the typical spatial world-line configuration is the one where up-spin and down-spin world lines appear alternatively with small overlap. The overlap is strongly discouraged by a large value of U. In algorithm P, in order to exchange the positions of two world-lines, we need intermediate states in which the two world lines overlap at least partially. We avoid these intermediate states with the algorithms L and L_{ex}. We have observed strong SDW ordering of the system by looking at the average

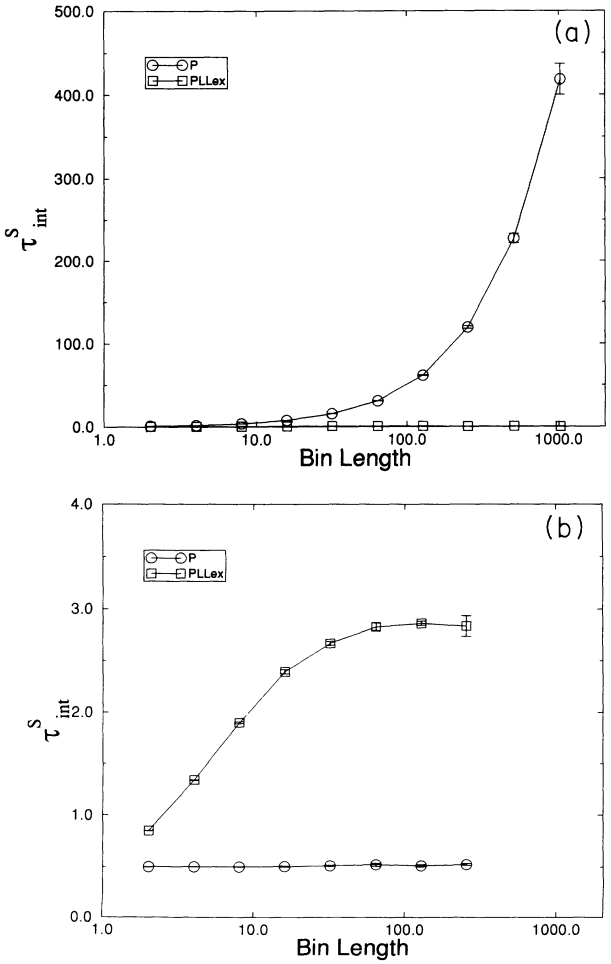

FIG. 8. The bin-length length dependence of $\tau^A_{int}(l)$ for the SDW fluctuation for both the P and PLL_{ex} algorithms. (a) A case where $\tau^A_{int}(l)$ does not converge ($N=32$, $L=160$, $U=8$, $\beta=8$, and $\mu=0.0$). (b) A case where $\tau^A_{int}(l)$ converges.

values of T_{χ_-} at $q=\pi$. To show an example, $T_{\chi_-}(\pi)=0.884(1)$ at $U=8$ and $\beta=0.5$ for PLL_{ex} whereas it becomes $T_{\chi_-}(\pi)=2.000(8)$ at $\beta=8$.

Figure 9(c) for the CDW function sharply contrasts that of Fig. 9(b) for the SDW function in that a significant increase in the correlation time does not occur as the temperature is lowered. (We again emphasize that the τ_{int} displayed for algorithm P are in general lower bound estimates and are likely to be too small.) In fact, the correlation time is almost constant for $\beta>6$ in all six curves. The difference between P and PLL_{ex} is far less noticeable: the correlation time for P is only a few times larger than its counterpart for PLL_{ex}. Consistent with these observations is the fact that CDW order is much weaker than the SDW order. For example, $T_{\chi_+}(\pi)=0.1281(5)$ at $U=8$ and $\beta=0.5$ for PLL_{ex} whereas it becomes $T_{\chi_+}(\pi)=0.003\,80(2)$ at $\beta=8$.

The longest integrated correlation times for PLL_{ex} are seen in Fig. 9(d) for the average electron number. Since there are no fluctuations in the particle number in algorithm P, there is no corresponding curves for P for this quantity. However, even for PLL_{ex}, the fluctuation in the particle number is very small. The variance in the particle number is sometimes so small that we could not obtain a reliable estimate for its correlation time. This undetermination is why some data points are missing in the figure, especially in the region of large correlation times at lower temperatures.

The comparison between τ_{int} and τ_{exp} in Fig. 9(d) shows that in most cases the integrated correlation time τ_{int} for the particle number is nearly as large as the longest mode correlation time τ_{exp}. This fact suggests that the longest mode in PLL_{ex} is the mode in which creation and annihilation of particles are involved. Empirically, the small variance in the particle number and the long correlation time associated with charge creation and annihilation appear closely related. For example, in the case where $N=32$, $U=8$, $\mu=0$, $1/T=4$, and $N=32$, the square root of the particle number variance (i.e., one standard deviation) measured by the algorithm PLL_{ex} is 0.02. For lower temperatures, with the rest of parameters the same, we were unable to estimate this quantity reliably. However, the influence of this long mode for relevant physical quantities may be negligible because the change in this quantity may be so small that we can effectively consider it as a constant. This is the case with the long mode associated to the average particle number, since the average values of other physical quantities, such as energy and static SDW susceptibility, are not as sensitive to particle number fluctuations so this exponentially small fluctuation cannot appreciably affect the average value of those quantities. In this sense, the longest mode correlation time is not necessarily the relevant measure for the computational time, except if we are interested in the longest mode itself.

On the other hand, although the creation and annihilation of particles are rather rare and cause long τ_{exp}'s in some cases, we found that it is still important in other cases to allow them to happen in order to reduce the in-

tegrated correlation times for quantities of physical interest. To see this, we have performed some simulations in which all the Monte Carlo attempts for loops whose winding number is nonvanishing (and therefore may change the particle number) are rejected. In Table I, we show the correlation times for these special simulations in comparison with the grand-canonical simulations. The result clearly shows the importance of creation and annihilation processes.

As already mentioned, the present algorithm allows negative sign configurations to occur. We found, however, the negative sign ratio defined by[19]

$$\Gamma \equiv (Z_+ - Z_-)/(Z_+ + Z_-) , \tag{46}$$

where Z_+ (Z_-) is the Boltzmann weight for the subset of configurations that are positive (negative), is close to unity for most cases, especially for larger system sizes. This observation means that the "sign problem" is not a problem for the new algorithms. In fact, the only cases for which we found any configurations of negative sign are the cases where $N=32$, $\mu=0.0$, $U=2$, $y=0.5$, and $\beta=4$, 6, 8, and 10. The values of Γ for these cases are 0.9978(6), 0.979(4), 0.964(5), and 0.88(1). At larger values of U or higher temperatures, we did not observe any negative sign configurations. For smaller lattices, such as $N=16$, we observed a larger number of negative sign configurations, although they are again just a small fraction of the total. We expect that the negative sign ratio approaches to unity as the system size becomes large.

The reduction in autocorrelation times brought about by algorithm L_{ex} is most noticeable for the integrated correlation time for the SDW correlation function. In Table II, we list the correlation times for various combinations of the three algorithms, in the case where $U=4$, $\beta=1.5$, $N=32$, and $\mu=0.0$, to examine the efficiency of several combinations. We can see that the integrated correlation time $\tau_{\rm int}^S$ for the SDW correlation function for PLL_{ex} is about $\frac{1}{8}$ of that for PL. Since we have not carried out this type of investigation for a variety of physical parameters systematically, cases may exist where the efficiency of L_{ex} is much more noticeable than presented here. The reason why the efficiency of L_{ex} is most noticeable in the correlation time for SDW is conjectured as follows: As already argued, the integrated

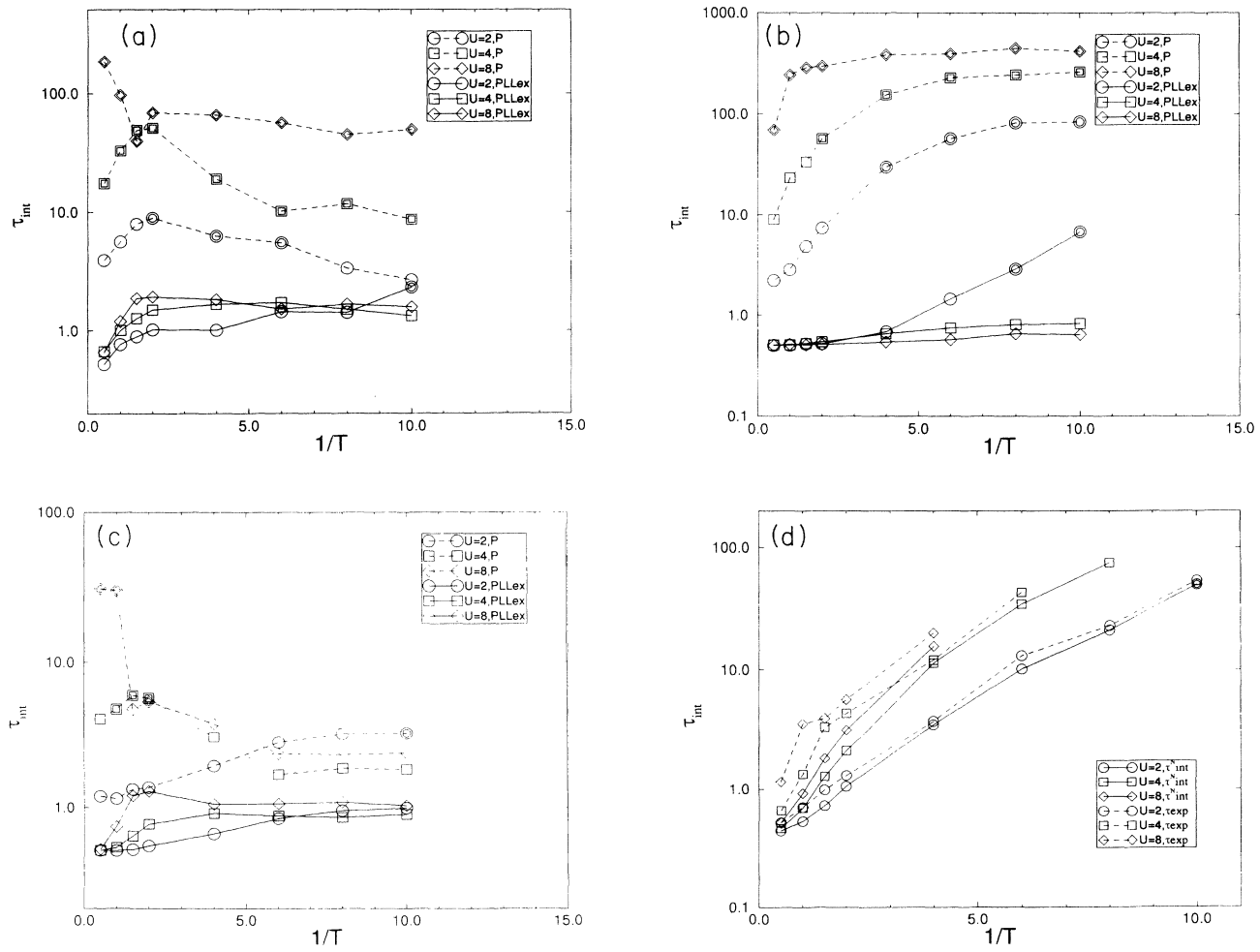

FIG. 9. The integrated autocorrelation times for (a) energy, (b) SDW fluctuation, (c) CDW fluctuation, and (d) the number of particles, in the case of $N=32$, $\mu=0.0$, and $U\beta/L=0.5$. The double symbols and dashed curves show (most likely poor) lower bounds for the correlation times, not the correlation times themselves. The curves are computed for several values of U.

TABLE I. Autocorrelation times for $U=8$, $L=8$, $1/T=1$, $N=16$, and $\mu=0$.

Algorithm	τ_{int}^{E}	τ_{int}^{N}	τ_{int}^{C}	τ_{int}^{S}	τ_{exp}
PL (canonical)	>46	∞	13(2)	60(5)	[a]
PL (grand canonical)	1.7(2)	0.89(3)	0.81(1)	10.8(9)	19(7)
PLL_{ex} (canonical)	>28	∞	10.7(1)	1.80(5)	[a]
PLL_{ex} (grand canonical)	0.99(4)	0.92(2)	0.718(7)	0.509(8)	1.75(5)

[a]Unable to measure.

correlation time for SDW fluctuations is related to the exchange of two world lines of opposite spins. Although we can achieve this exchange through either algorithm *P* or *L*, we inevitably encounter unpleasant intermediate states in the case of *P*. Algorithm *L* may suffer from similar difficulty because in forming loops in either the up-spin plane or the down-spin plane it does not use the information about the configuration in the other plane. Therefore, in some cases, the attempted movement of the world line results in larger overlaps between down-spin world lines and up-spin ones, and the move is improbable. On the other hand, the algorithm L_{ex} is reasonably free from this kind of difficulty.

The acceptance ratios of the various algorithms also shows a wide range of behavior. In Table III, we list the acceptance ratios for some typical cases. The acceptance ratio *r* for one MCS is defined as

$$r \equiv \frac{[\text{the number of flipped sites}]}{n \times [\text{the number of sites in the system}]} , \quad (47)$$

where $n=2$ for *P* and $n=1$ for *L* and L_{ex}. The integer *n* is the number of times a site is flipped in a MCS. For the algorithm *P*, this definition is equivalent to the ordinary acceptance ratio defined as the number of accepted attempts divided by the total number of attempts. The numbers in Table III are averaged over all the Monte Carlo steps. We see that the acceptance ratio for *P* and *L* are strongly dependent on β and *U* and become very small as the value of *U* increases. In contrast, the acceptance ratio for L_{ex} is much less parameter dependent and only moderately increases to $\frac{1}{2}$ as *U* increases.

Some results on the quarter-filled case are presented in Table IV. In this case, in order to compare the results to those of the algorithm *P* on a roughly equal basis, we adjusted the chemical potential μ for the algorithm PLL_{ex} so that the resulting average particle number becomes close to 0.5. The difference between the algorithms is less striking than that in the half-filled case although we can still see considerable reduction in the correlation time. We remark that for PLL_{ex} the autocorrelations time for quarter filling are comparable to those at half filling. Since the overhead of PLL_{ex}, relative to *P*, is approximately a factor of 10 larger, the quarter-filled case is near a marginal filling where the overhead and the reduction of correlation time balances.

V. CONCLUSION

We presented two new algorithms for world-line Monte Carlo simulations of electron systems and demonstrated their efficiency in the case of the one-dimensional, repulsive Hubbard model. Their extension to higher dimensions is straightforward. The first algorithm, which we call the loop-flip algorithm, is an extension to fermion systems of a loop-flip algorithm for a six-vertex model recently proposed by Evertz *et al.*[4] Our new algorithm enables us to simulate the grand-canonical ensemble while the traditional plaquette flip algorithm simulates in the canonical ensemble. We also found that switching to grand-canonical simulation often reduces the autocorrelation time of the simulation dramatically. The second algorithm is the loop-exchange algorithm in which up-spin and down-spin world-line segments exchange their locations. Since all world-line movement in the loop-exchange algorithm is unaccompanied by changes in the chemical potential term or the *U* term in the Hamiltonian, we expected that this algorithm would be effective when the long correlation time is due to difficulties in exchanging world lines. Consistent with this expectation, we found that the loop-exchange algorithm is especially effective in reducing the autocorrelation time of the SDW correlation function when the value of *U* is large. These algorithms generalize the cluster algorithms[5] recently developed mainly to reduce long autocorrelation times accompanying simulations of critical phenomena in classical spin systems. In contrast to the usual use of cluster

TABLE II. Autocorrelation times for $U=4$, $L=12$, $1/T=1.5$, $N=32$, and $\mu=0$.

Algorithm	τ_{int}^{E}	τ_{int}^{N}	τ_{int}^{C}	τ_{int}^{S}	τ_{exp}
P	102(7)	∞	17(1)	45(2)	[a]
L	4.2(3)	1.22(6)	0.73(3)	5.6()5)	10(1)
PL	2.0(2)	1.20(3)	0.63(2)	3.8(1)	5.1(5)
PL_{ex}	>33	∞	>5.2	>1.7	[a]
LL_{ex}	3.1(2)	1.28(4)	0.711(4)	0.60(3)	5.9(8)
PLL_{ex}	1.26(2)	1.28(8)	0.63(3)	0.525(4)	3.4(6)

[a]Unable to measure.

TABLE III. Acceptance ratios for $N=32$, $U/\beta L=1$, and $\mu=0$.

β	Algorithm	$U=2$	$U=4$	$U=8$
2	*P*	0.146	0.071	0.020
	L	0.208	0.074	0.012
	L_{ex}	0.349	0.413	0.478
6	*P*	0.157	0.081	0.028
	L	0.105	0.050	0.014
	L_{ex}	0.355	0.415	0.471

TABLE IV. Autocorrelation times for $N=16$, $L=8$, and $U=8$, at quarter filling or nearly quarter filling.

Algorithm	$1/T$	$\langle n \rangle$	τ_{int}^{E}	τ_{int}^{N}	τ_{int}^{C}	τ_{int}^{S}	τ_{exp}
P	1.0	0.5	12.4(9)	∞	6.8(4)	9.9(5)	[a]
P	2.0	0.5	22(5)	∞	8.1(3)	10.6(8)	[a]
P	4.0	0.5	28(2)	∞	10.0(7)	11(1)	[a]
PLL_{ex}	1.0	0.4975(2)	1.12(5)	1.24(6)	0.55(1)	0.514(5)	1.21(3)
PLL_{ex}	2.0	0.4940(6)	1.51(5)	3.3(2)	0.786(9)	0.60(2)	4.0(1)
PLL_{ex}	4.0	0.4824(3)	2.0(2)	34(4)	1.49(3)	1.1(1)	35(2)

[a]Unable to measure.

algorithms, we were concerned with reducing the inherently long autocorrelation times that occur in the world-line method even when the physical system is far removed from any known finite-temperature phase transition.

At the lowest simulated temperatures, we observed reductions of the autocorrelation time, measured in the units of Monte Carlo steps, as large as 3 orders of magnitudes. One MCS in the algorithm PLL_{ex}, however, takes from 8 to 10 times longer in CPU time than one step in algorithm P so the above reduction of autocorrelation time gives a reduction in CPU time by a factor as large as 2 orders of magnitude.

The estimates of improved efficiency are cautious ones. In most cases we were able only to estimate a lower bound for the autocorrelation times associated with the plaquette algorithm. It is not unreasonable in some cases to expect the actual autocorrelation times to be an order of magnitude larger. Additionally, it should be possible to improve the efficiency of our implementation of the loop algorithms. We used a highly optimized code for the plaquette algorithm but used unoptimized codes for the loop algorithms. Further, we used ordinary workstations for all the calculations. Parallelizing these computations is possible and will most likely be needed when the present algorithms are applied to more challenging problems, such as those in higher dimensions.

It is natural to ask about the extensions of the present methods to higher dimensions and other models. When one changes the problem, it is perhaps best to ask whether one should also change the methods. The algorithms presented were designed for the one-dimensional repulsive Hubbard model but can be viewed from the perspective of a general approach to developing cluster algorithms.[17] Tailoring the algorithms, when extended to other problems, might be possible. Still, the direct use of our new methods to many other problems is possible. How efficient this use will be is what needs investigation. For many cases, their use should be more efficient than the standard algorithm.

The most obvious extension would be to the two-dimensional repulsive Hubbard model. The standard world-line algorithm suffers such severe sign problems in two dimensions that is often almost pointless to use it. We have no reason to believe our algorithms reduce the sign problem. Our best expectation would be the achievement of a significant enough reduction in the variances of the measured physical quantities so simulations could be performed over a limited range of parameters. Wiese[20] has reported the use an extension of the massless version of the Evertz *et al.* loop-flip algorithm on the two-dimensional, free-fermion problem, and the few high-temperature results he reported show seemingly accurate values for the average electron occupancy even in the presence of a severe sign problem. When $U=\mu=0$ our loop-flip algorithm for fermions reduces to his if we only attempt to flip one loop at each Monte Carlo step. The potential effectiveness of these methods for fermions problems in two or more dimensions, while appearing not especially promising, needs more study.

Of our two algorithms, the loop-exchange method is the one most specific to the repulsive Hubbard and related models. If $U<0$, an algorithm, which breaks up world lines wanting to be on top of one another, would be more appropriate and should be possible to construct. For the extended Hubbard model, which has a Coulomb term between electrons on neighboring sites, an additional specific construct might also be necessary. The loop-exchange algorithm, because it addresses the hopping part of the problem, will change little from problem to problem.

Loop algorithm L, in fact, can be almost directly applied to the spin-$\frac{1}{2}$ quantum spin chains. This applicability follows from the similarity the path-integral representation of the spin-$\frac{1}{2}$ quantum spin system has with spinless fermion model.[11] Use of these algorithms in two dimensions also appears quite direct and desirable. Wiese and Ying,[21] using what turns out to be the massless version of the Evertz *et al.* method, studied the spin-$\frac{1}{2}$ Heisenberg antiferromagnet with success, although they do not report efficiency figures of merit. Recently, we developed cluster algorithms for spin of arbitrary magnitude[22] and are in the process of testing these methods. We will report our results elsewhere.

In closing, we comment that the total temporal and spatial winding numbers are both physically important. The total temporal winding number is the total number of particles and the average of the square of its fluctuations is related to the charge compressibility of the system. The average of the square of the fluctuations of the total spatial winding number is directly related to the dc conductivity.[23] In the conventional world-line Monte Carlo, i.e., algorithm P, the total winding number of ei-

ther type is conserved. To estimate the charge compressibility and dc conductivity, we have to do some additional manipulations. In the new algorithms, these manipulations seem unnecessary because the total winding numbers of both kinds can change simultaneously, individually, or not at all. We can see the changes by noting that in algorithm L flipping a single loop of spatial winding number w changes the total spatial winding number by w, and that a loop with any spatial winding number can form with a finite probability as long as it is allowed in the size of the system under consideration. A similar remark can be made with respect to the temporal winding number. Therefore, we can directly estimate the above-mentioned quantities. We have yet to investigate the efficiency of this possibility because in this paper we were mainly concerned with overall algorithmic issues and development. An additional important point is that similar possibilities will also exist for the relevant extension of the loop algorithms to bosons and quantum spin systems.

ACKNOWLEDGMENTS

The work of J.E.G. was supported by the Department of Energy's High Performance Computing and Communication program at the Los Alamos National Laboratory. H.G.E. would like to express his gratitude to the Department of Energy for providing computer time on the Cray Y-MP computer at Florida State University. We thank A. Sandvik for several helpful comments regarding our original manuscript.

[1]For examples of applications to quantum spin systems, see *Quantum Monte Carlo in Equilibrium and Non-Equilibrium Systems,* edited by M. Suzuki (Springer-Verlag, Heidelberg, 1987).

[2]J. W. Negle and H. Orland, *Quantum Many-Particle Systems* (Addison-Wesley, New York, 1988), Chap. 8.

[3]J. E. Hirsch, R. L. Sugar, D. J. Scalapino, and R. Blankenbecler, Phys. Rev. Lett. **47**, 1628 (1981); Phys. Rev. B **26**, 5033 (1982).

[4]H. Evertz, G. Lana, and M. Marcu, Phys. Rev. Lett. **70**, 875 (1993).

[5]For a review of various methods, see D. Kandel and E. Domany, Phys. Rev. B **43**, 8539 (1991).

[6]J. E. Hirsch and D. J. Scalapino, Phys. Rev. Lett. **50**, 1168 (1983); Phys. Rev. B **29**, 5554 (1984).

[7]M. Suzuki, Prog. Theor. Phys. **56**, 1454 (1976).

[8]R. Blankenbecler, R. L. Sugar, and D. J. Scalapino, Phys. Rev. D **24**, 2278 (1981).

[9]Y. Okabe and M. Kikuchi, Phys. Rev. B **34**, 7896 (1986).

[10]M. Marcu, J. Müller, and F.-K. Schimatzer, Comput. Phys. Commun. **44**, 63 (1987).

[11]W. R. Somsky and J. E. Gubernatis, Comput. Phys. **6**, 178 (1992).

[12]K. Binder and D. W. Heermann, *Monte Carlo Simulation in Statistical Physics* (Springer-Verlag, Heidelberg, 1988), Chap. 2.

[13]M. P. Allen and D. J. Tildesley, *Computer Simulations of Liquids* (Oxford University Press, Oxford, 1987), Chap. 6.

[14]J. M. Hammersley and D. C. Handcomb, *Monte Carlo Methods* (Methuen, London, 1964), Chap. 5.

[15]R. J. Baxter, *Exactly Solved Models in Statistical Mechanics* (Academic, London, 1982), Chap. 8.

[16]H. G. Evertz and M. Marcu, in *Computer Simulations in Condensed Matter Physics VI,* edited by D. P. Landau, K. K. Mon, and H.-B. Schütterr (Springer-Verlag, Berlin, 1993), p. 109.

[17]N. Kawashima and J. E. Gubernatis (unpublished).

[18]These exceptional cases are 1M MCS for four simulations with algorithm P at quarter filling when $N=32$ and $U=4$; 8M MCS for a simulation also with algorithm P now at half filling but again when $N=32$ and $U=4$; and 10M MCS for simulation with algorithm a canonical version of PL at half filling when $N=32$ and $U=8$.

[19]E. Y. Loh, Jr., J. E. Gubernatis, R. T. Scaletarr, S. R. White, D. J. Scalapino, and R. L. Sugar, Phys. Rev. B **41**, 9301 (1990).

[20]U.-J. Wiese, Phys. Lett. B **311**, 235 (1993).

[21]U.-J. Wiese and H.-P. Ying (unpublished).

[22]N. Kawashima and J. E. Gubernatis (unpublished).

[23]D. J. Scalapino, S. R. White, and S. C. Zhang, Phys. Rev. B **47**, 7995 (1993), and references therein.