

Directed update for the stochastic Green function algorithm

V. G. Rousseau

Instituut-Lorentz, LION, Universiteit Leiden, Postbus 9504, 2300 RA Leiden, The Netherlands

(Received 9 June 2008; revised manuscript received 12 August 2008; published 24 November 2008)

In a recent paper we presented a stochastic Green function (SGF) algorithm, which has the properties of being general and easy to apply to any lattice Hamiltonian of the form $\hat{\mathcal{H}} = \hat{\mathcal{V}} - \hat{\mathcal{T}}$, where $\hat{\mathcal{V}}$ is diagonal in the chosen occupation number basis and $\hat{\mathcal{T}}$ has only positive matrix elements. We propose here a modified version of the update scheme that keeps the simplicity and generality of the original SGF algorithm, and significantly enhances its efficiency.

DOI: [10.1103/PhysRevE.78.056707](https://doi.org/10.1103/PhysRevE.78.056707)

PACS number(s): 02.70.Uu, 05.30.Jp

I. INTRODUCTION

Monte Carlo methods [1] appeared about 60 years ago with the need to evaluate numerical values for various complex problems. These methods evolved and were applied early to quantum problems, thus putting within reach exact numerical solutions to nontrivial quantum problems [2–5]. Many improvements of these methods followed, avoiding critical slowing down near phase transitions and allowing one to work directly in the continuous imaginary time limit [6–13]. In recent years, interest in methods that work in the canonical ensemble with global updates and yet allow access to Green functions has intensified [14,15]. However, a method that works well for a given Hamiltonian often needs major modifications for another. For example, the addition of a four-site ring exchange term in the bosonic Hubbard model required special developments for a treatment by the stochastic series expansion algorithm [16], as well as by the world-line algorithm [17]. This can result in long delays. It is, therefore, advantageous to have at one's disposal an algorithm that can be applied to a very wide class of Hamiltonians without requiring any changes. In a recent paper [15], the stochastic Green function (SGF) algorithm was presented, which meets this goal. The algorithm can be applied to any lattice Hamiltonian of the form

$$\hat{\mathcal{H}} = \hat{\mathcal{V}} - \hat{\mathcal{T}}, \quad (1)$$

where $\hat{\mathcal{V}}$ is diagonal in the chosen occupation number basis and $\hat{\mathcal{T}}$ has only positive matrix elements. This includes all kinds of systems that can be treated by other methods presented in Refs. [5,11–14], for instance Bose-Hubbard models with or without a trap, Bose-Fermi mixtures in one dimension, Heisenberg models, and so on. In particular, Hamiltonians for which the nondiagonal part $\hat{\mathcal{T}}$ is nontrivial (the eigenbasis is unknown) are easily treated, such as the Bose-Hubbard model with ring exchange [16,17], or multispecies Hamiltonians in which a given species can be turned into another one [see Eq. (49) and Figs. 3 and 4 for a concrete example]. Systems for which it is not possible to find a basis in which $\hat{\mathcal{V}}$ is diagonal and $\hat{\mathcal{T}}$ has only positive matrix elements are said to have a “sign problem,” which usually arises with fermionic and frustrated systems. Like other quantum Monte Carlo (QMC) methods, the SGF algorithm does not solve this problem.

The algorithm allows one to measure several quantities of interest, such as the energy, the local density, local compressibility, density-density correlation functions, and so on. In particular, the winding is sampled and gives access to the superfluid density. Equal-time n -body Green functions are probably the most interesting quantities that can be measured by the algorithm, by giving access to momentum distribution functions which allow direct comparisons with experiments. All details on measurements are given in Ref. [15].

In addition the algorithm has the property of being easy to code, due in part to a simple update scheme in which all moves are accepted with a probability of 1. Despite such generality and simplicity, the algorithm might suffer from a reduced efficiency, compared to other algorithms in situations where they can be applied.

The purpose of this paper is to present a “directed” update scheme that (i) keeps the simplicity and generality of the original SGF algorithm, and (ii) enhances its efficiency by improving the sampling over the imaginary time axis. While the SGF algorithm is not intended to compete with the speed of other algorithms, the improvement resulting from the directed update scheme is remarkable (see Sec. V). But what makes the strength of the SGF method is that it allows one to simulate Hamiltonians that cannot be treated by other methods or that would require special developments [see Eq. (49) for a concrete example]. The paper is organized as follows. We introduce in Sec. II the notations and definitions used in Ref. [15]. In Sec. III, we propose a simplification of the update scheme used in the original SGF algorithm, and determine how to satisfy detailed balance. A generalization of the simplified update scheme is presented in Sec. IV, which constitutes the directed update scheme. Finally, Sec. V shows how to determine the introduced optimization parameters, and presents some tests of the algorithm and a comparison with the original version.

II. DEFINITIONS AND NOTATIONS

In this section, we recall the expression of the “Green operator” introduced in the SGF algorithm, and the extended partition function which is considered. Although not required for understanding this paper, we refer the reader to Ref. [15] for full details on the algorithm. Like many QMC algorithms, the SGF algorithm samples the partition function

$$\mathcal{Z}(\beta) = \text{Tr} e^{-\beta\hat{H}}. \quad (2)$$

The algorithm has the property of working in the canonical ensemble. In order to define the Green operator, we first define the “normalized” creation and annihilation operators,

$$\hat{A}^\dagger = a^\dagger \frac{1}{\sqrt{\hat{n}+1}}, \quad \hat{A} = \frac{1}{\sqrt{\hat{n}+1}} a, \quad (3)$$

where a^\dagger and a are the usual creation and annihilation operators of bosons, and $\hat{n}=a^\dagger a$ is the number operator. From Eq. (3) one can show the following relations for any state $|n\rangle$ in the occupation number representation:

$$\hat{A}^\dagger |n\rangle = |n+1\rangle, \quad \hat{A} |n\rangle = |n-1\rangle, \quad (4)$$

with the particular case $\hat{A}|0\rangle=0$. Apart from this exception, the operators \hat{A}^\dagger and \hat{A} change a state $|n\rangle$ by respectively creating and annihilating one particle, but they do not change the norm of the state.

Using the notation $\{i_p|j_q\}$ to denote two subsets of site indices i_1, i_2, \dots, i_p and j_1, j_2, \dots, j_q with the constraint that all indices in subset i are different from the indices in subset j (but several indices in one subset may be equal), we define the Green operator \hat{G} by

$$\hat{G} = \sum_{p=0}^{+\infty} \sum_{q=0}^{+\infty} g_{pq} \sum_{\{i_p|j_q\}} \prod_{k=1}^p \hat{A}_{i_k}^\dagger \prod_{l=1}^q \hat{A}_{j_l}, \quad (5)$$

where g_{pq} is a matrix that depends on the application of the algorithm [15]. In order to sample the partition function (2), an extended partition function $\mathcal{Z}(\beta, \tau)$ is considered by breaking up the propagator $e^{-\beta\hat{H}}$, and introducing the Green operator between the broken parts,

$$\mathcal{Z}(\beta, \tau) = \text{Tr} e^{-(\beta-\tau)\hat{H}} \hat{G} e^{-\tau\hat{H}}. \quad (6)$$

Defining the time-dependent operators $\hat{T}(\tau)$ and $\hat{G}(\tau)$,

$$\hat{T}(\tau) = e^{\tau\hat{V}} \hat{T} e^{-\tau\hat{V}}, \quad \hat{G}(\tau) = e^{\tau\hat{V}} \hat{G} e^{-\tau\hat{V}}, \quad (7)$$

and working in the occupation number basis in which \hat{V} is diagonal, the extended partition function takes the form

$$\begin{aligned} \mathcal{Z}(\beta, \tau) = & \sum_{n \geq 0} \int_{0 < \tau_1 < \dots < \tau_n < \beta} \langle \psi_0 | e^{-\beta\hat{V}} \hat{T}(\tau_n) | \psi_{n-1} \rangle \langle \psi_{n-1} | \hat{T}(\tau_{n-1}) \\ & \times | \psi_{n-2} \rangle \times \dots \times \langle \psi_{L+1} | \hat{T}(\tau_L) | \psi_L \rangle \langle \psi_L | \hat{G}(\tau) | \psi_R \rangle \\ & \times \langle \psi_R | \hat{T}(\tau_R) | \psi_{R-1} \rangle \times \dots \times \langle \psi_2 | \hat{T}(\tau_2) | \psi_1 \rangle \langle \psi_1 | \hat{T}(\tau_1) \\ & \times | \psi_0 \rangle d\tau_1 \dots d\tau_n, \end{aligned} \quad (8)$$

where the sum $\sum_{n \geq 0}$ implicitly runs over complete sets of states $\{|\psi_k\rangle\}$. We will systematically use the labels L and R to denote the states appearing on the left and the right of the Green operator, and use the notation V_k to denote the diagonal energy $\langle \psi_k | \hat{V} | \psi_k \rangle$. We will also denote by τ_L and τ_R the time indices of the \hat{T} operators appearing on the left and the right of \hat{G} .

As a result, the extended partition function is a sum over all possible configurations, each being determined by a set of time indices $\tau_1, \tau_2, \dots, \tau_R, \tau, \tau_L, \dots, \tau_n$ and a set of states $|\psi_0\rangle, |\psi_1\rangle, \dots, |\psi_R\rangle, |\psi_L\rangle, \dots, |\psi_{n-1}\rangle$. The algorithm consists in updating those configurations by making use of the Green operator. Assuming that the Green operator is acting at time τ , it can “create” a \hat{T} operator (that is to say a \hat{T} operator can be inserted in the operator string) at the same time, thus introducing a new intermediate state, then it can be shifted to a different time. While shifting, any \hat{T} operator encountered by the Green operator is “destroyed” (that is to say removed from the operator string). Assuming a left (or right) move, creating an operator will update the state $|\psi_R\rangle$ (or $|\psi_L\rangle$), while destroying will update the state $|\psi_L\rangle$ (or $|\psi_R\rangle$). When a diagonal configuration of the Green operator occurs, $|\psi_L\rangle = |\psi_R\rangle$, such a configuration associated to the extended partition function (8) is also a configuration associated to the partition function (2). Measurements can be done when this occurs (see Ref. [15] for details on measurements).

The next section presents a simple update scheme that meets the requirements of ergodicity and detailed balance.

III. SIMPLIFIED UPDATE SCHEME

Before introducing the directed update, we start by simplifying the update scheme used in the original SGF algorithm.

A. The update scheme

We will assume in the following that a left move of the Green operator is chosen. In the original version, the Green operator $\hat{G}(\tau)$ can choose to create or not on its right a \hat{T} operator at time τ . Then a time shift $\Delta\tau$ to the left is chosen for the Green operator with an exponential distribution in the range $[0; +\infty[$. If an operator is encountered while shifting the Green operator, then the operator is destroyed and the move stops there. As a result, four possible situations can occur during one move: (1) No creation, shift, no destruction; (2) Creation, shift, no destruction; (3) No creation, shift, destruction; (4) Creation, shift, destruction. It appears that the first possibility, “no creation, no destruction,” is actually useless, since no change is performed in the operator string. The idea is to get rid of this possibility by forcing the Green operator to destroy an operator if no creation is chosen. A further simplification can be done by noticing that the last possibility, “creation, destruction,” is not necessary for the ergodicity of the algorithm, and can be avoided by restricting the range of the time shift after having created an operator. Therefore we replace the original update scheme by the following. We assume that the Green operator is acting at time τ and that the operator on its left is acting at time τ_L . The Green operator $\hat{G}(\tau)$ chooses to create or not an operator on its right at time τ . If creation is chosen, then a time shift $\Delta\tau$ of the Green operator is chosen to the left in the range $[0; \tau_L - \tau[$, with the probability distribution defined below. If no creation is chosen, then the Green operator is directly shifted to the operator on its left at time τ_L , and the operator is destroyed. As a result only two possibilities have to be

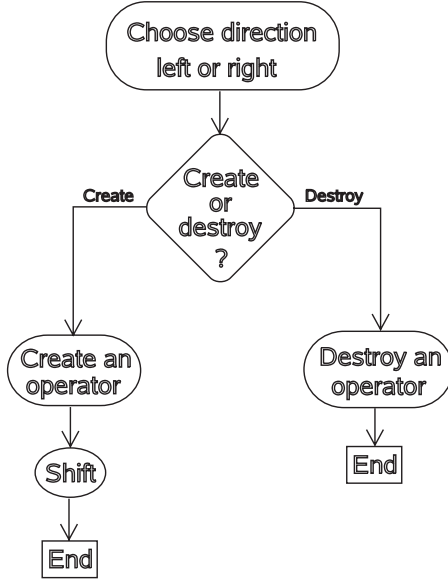


FIG. 1. Simplified update scheme. See text for details.

considered: (1) Creation, shift; (2) Shift, destruction.

Figure 1 shows the associated organigram. Section III B explains how detailed balance can be satisfied with this simplified update scheme.

B. Detailed balance

When updating the configurations according to the chosen update scheme, we need to generate different transitions from initial to final states with probabilities that satisfy detailed balance. In this section we propose a choice for these probabilities, and determine the corresponding acceptance factors. We denote the probability of the initial (final) configuration by P_i (P_f). We denote by $S_{i \rightarrow f}$ the probability of the transition from configuration i to configuration f , and by $S_{f \rightarrow i}$ the probability of the reverse transition. Finally we denote by $A_{i \rightarrow f}$ the acceptance rate of the transition from i to f , and by $A_{f \rightarrow i}$ the acceptance rate of the reverse transition. The detailed balance can be written as

$$P_i S_{i \rightarrow f} A_{i \rightarrow f} = P_f S_{f \rightarrow i} A_{f \rightarrow i}. \quad (9)$$

We will make use of the Metropolis solution [18]

$$A_{i \rightarrow f} = \min(1, q) \quad (10)$$

with

$$q = \frac{P_f S_{f \rightarrow i}}{P_i S_{i \rightarrow f}}. \quad (11)$$

We will use primed (nonprimed) labels for states and time indices to denote final (initial) configurations.

1. Creation, shift

We consider here the case where a left move is chosen, an operator is created on the right of the Green operator at time τ , and a new state is chosen. Then a time shift to the left is chosen for the Green operator in the range $[0, \tau'_L - \tau'_R]$. It is

important to note that τ'_L and τ'_R correspond to the time indices of the operators appearing on the left and the right of the Green operator after the new operator has been inserted, that is to say, at the moment where the time shift needs to be performed. Thus we have $\tau'_L = \tau_L$ and $\tau'_R = \tau$.

The probability of the initial configuration is the Boltzmann weight appearing in the extended partition function (8):

$$P_i \propto \langle \psi_L | \hat{G}(\tau) | \psi_R \rangle \propto e^{\tau V_L} \langle \psi_L | \hat{G} | \psi_R \rangle e^{-\tau V_R}. \quad (12)$$

The probability of the final configuration takes the form

$$P_f \propto \langle \psi'_L | \hat{G}(\tau') | \psi'_R \rangle \langle \psi'_R | \hat{T}(\tau'_R) | \psi'_{R-1} \rangle \\ \propto e^{\tau' V'_L} \langle \psi'_L | \hat{G} | \psi'_R \rangle e^{-(\tau' - \tau'_R) V'_R} \langle \psi'_R | \hat{T} | \psi'_{R-1} \rangle e^{-\tau'_R V'_{R-1}}. \quad (13)$$

It is important here to realize that the Green operator inserted only on its right the operator $|\psi'_R\rangle\langle\psi'_R|\hat{T}(\tau'_R)$, before being shifted from τ'_R to τ' . Therefore we have the equalities $\langle\psi'_L| = \langle\psi_L|$, $|\psi'_{R-1}\rangle = |\psi_R\rangle$, $V'_L = V_L$, and $V'_{R-1} = V_R$.

The probability $S_{i \rightarrow f}$ of the transition from the initial configuration to the final configuration is the probability $P(\leftarrow)$ of a left move, times the probability $P_{-}^{\dagger}(\tau)$ of a creation, times the probability $P_{-}(\psi'_R)$ to choose the new state ψ'_R , times the probability $P_{-}^{L'R'}(\tau' - \tau'_R)$ to shift the Green operator by $\tau' - \tau'_R$, knowing that the states on the left and the right of the Green operator at the moment of the shift are ψ'_L and ψ'_R :

$$S_{i \rightarrow f} = P(\leftarrow) P_{-}^{\dagger}(\tau) P_{-}(\psi'_R) P_{-}^{L'R'}(\tau' - \tau'_R). \quad (14)$$

The probability of the reverse transition is simply the probability $P(\rightarrow')$ of a right move, times the probability of no creation, $1 - P_{-}^{\dagger}(\tau')$:

$$S_{f \rightarrow i} = P(\rightarrow') [1 - P_{-}^{\dagger}(\tau')]. \quad (15)$$

From the original version of the SGF algorithm, we know that choosing the time shift with an exponential distribution is a good choice, because it cancels the exponentials appearing in the probabilities of the initial (12) and final (13) configurations, avoiding exponentially small acceptance factors. However a different normalization must be used here, since the time shift is chosen in the range $[0; \tau'_L - \tau'_R]$ instead of $[0; +\infty[$. The suitable solution is:

$$P_{-}^{L'R'}(\Delta\tau) = \frac{(V'_R - V'_L) e^{-\Delta\tau(V'_R - V'_L)}}{1 - e^{-(\tau'_L - \tau'_R)(V'_R - V'_L)}}. \quad (16)$$

It is straightforward to check that the above probability is correctly normalized and well defined for any real value of $V'_R - V'_L$, the particular case $V'_L = V'_R$ reducing to the uniform distribution $P(\Delta\tau) = 1/(\tau'_L - \tau'_R)$ (note that $\tau'_L - \tau'_R$ is always a positive number). For the probability $P_{-}(\psi'_R)$ to choose the new state ψ'_R , the convenient solution is the same as in the original version:

$$P_{\leftarrow}(\psi'_R) = \frac{\langle \psi_L | \hat{G} | \psi'_R \rangle \langle \psi'_R | \hat{T} | \psi_R \rangle}{\langle \psi_L | \hat{G} \hat{T} | \psi_R \rangle}. \quad (17)$$

Putting everything together, the acceptance factor (11) becomes

$$q_{\leftarrow}^c = \frac{\langle \psi_L | \hat{G} \hat{T} | \psi_R \rangle}{\langle \psi_L | \hat{G} | \psi_R \rangle P(\leftarrow) P_{\leftarrow}^{\dagger}(\tau)} \times \frac{P(\rightarrow')[1 - P_{\rightarrow}^{\dagger}(\tau')](1 - e^{-(\tau'_L - \tau'_R)(V'_R - V'_L)})}{V'_R - V'_L}, \quad (18)$$

where we have used the notation q_{\leftarrow}^c to emphasize that this acceptance factor corresponds to a creation. It is also important for the remainder of this paper to note that q_{\leftarrow}^c is written as a quantity that depends on the initial configuration, times a quantity that depends on the final configuration.

2. Shift, destruction

We consider here the case where a left move is chosen, and the operator on the left of the Green operator is destroyed. This move corresponds to the inverse of the above “creation, shift” move. Thus, the corresponding acceptance factor q_{\leftarrow}^d is obtained by inverting the acceptance factor q_{\leftarrow}^c , exchanging the initial time τ and final time τ' , and switching the direction. However $\tau_L - \tau_R$ represents an absolute time shift, so τ_L and τ_R do not have to be exchanged. We get

$$q_{\leftarrow}^d = \frac{V_L - V_R}{P(\leftarrow)[1 - P_{\leftarrow}^{\dagger}(\tau)](1 - e^{-(\tau_L - \tau_R)(V_L - V_R)})} \times \frac{\langle \psi'_L | \hat{G} | \psi'_R \rangle P(\rightarrow') P_{\rightarrow}^{\dagger}(\tau')}{\langle \psi'_L | \hat{T} \hat{G} | \psi'_R \rangle}, \quad (19)$$

which is written as a quantity that depends on the initial configuration, times a quantity that depends on the final configuration.

3. Simplification of the acceptance factors

We will use here the short notation $\langle \hat{G} \rangle$, $\langle \hat{G} \hat{T} \rangle$, and $\langle \hat{T} \hat{G} \rangle$ to denote, respectively, the quantities $\langle \psi_L | \hat{G} | \psi_R \rangle$, $\langle \psi_L | \hat{G} \hat{T} | \psi_R \rangle$, and $\langle \psi_L | \hat{T} \hat{G} | \psi_R \rangle$. As in Ref. [15], we have some freedom for the choice of the probabilities of choosing a left or right move, $P(\leftarrow)$ and $P(\rightarrow)$, and the probabilities of creation $P_{\leftarrow}^{\dagger}(\tau)$ and $P_{\rightarrow}^{\dagger}(\tau)$. A suitable choice for those probabilities can be done in order to accept all moves, resulting in an appreciable simplification of the algorithm. For this purpose, we impose the acceptance factor q_{\leftarrow}^c (or q_{\leftarrow}^d) to be equal to the acceptance factor q_{\leftarrow}^d (or q_{\leftarrow}^c). This allows us to determine the probabilities $P_{\leftarrow}^{\dagger}(\tau)$ and $P_{\rightarrow}^{\dagger}(\tau)$,

$$P_{\leftarrow}^{\dagger}(\tau) = \frac{\langle \hat{G} \hat{T} \rangle}{\langle \hat{G} \hat{T} \rangle + \langle \hat{G} \rangle} \frac{(V_L - V_R)}{1 - e^{-(\tau_L - \tau_R)(V_L - V_R)}} \quad (20)$$

$$P_{\rightarrow}^{\dagger}(\tau) = \frac{\langle \hat{T} \hat{G} \rangle}{\langle \hat{T} \hat{G} \rangle + \langle \hat{G} \rangle} \frac{(V_R - V_L)}{1 - e^{-(\tau_L - \tau_R)(V_R - V_L)}}, \quad (21)$$

and the acceptance factors $q_{\leftarrow}^c = q_{\leftarrow}^d$ and $q_{\rightarrow}^c = q_{\rightarrow}^d$ take the forms

$$q_{\leftarrow} = \frac{P(\rightarrow') r_{\leftarrow}(\tau)}{P(\leftarrow) r_{\rightarrow}(\tau')}, \quad q_{\rightarrow} = \frac{P(\leftarrow') r_{\rightarrow}(\tau)}{P(\rightarrow) r_{\leftarrow}(\tau')}, \quad (22)$$

with

$$r_{\leftarrow}(\tau) = \frac{\langle \hat{G} \hat{T} \rangle}{\langle \hat{G} \rangle} + \frac{V_L - V_R}{1 - e^{-(\tau_L - \tau_R)(V_L - V_R)}}, \quad (23)$$

$$r_{\rightarrow}(\tau) = \frac{\langle \hat{T} \hat{G} \rangle}{\langle \hat{G} \rangle} + \frac{V_R - V_L}{1 - e^{-(\tau_L - \tau_R)(V_R - V_L)}}. \quad (24)$$

Finally we can require the acceptance factors q_{\leftarrow} and q_{\rightarrow} to be equal. This implies

$$P(\leftarrow) = \frac{r_{\leftarrow}(\tau)}{r_{\leftarrow}(\tau) + r_{\rightarrow}(\tau)}, \quad P(\rightarrow) = \frac{r_{\rightarrow}(\tau)}{r_{\leftarrow}(\tau) + r_{\rightarrow}(\tau)}. \quad (25)$$

Defining $R(\tau) = r_{\leftarrow}(\tau) + r_{\rightarrow}(\tau)$, we are left with a single acceptance factor,

$$q = \frac{R(\tau)}{R(\tau')}, \quad (26)$$

which is independent of the chosen direction, and independent of the nature of the move (creation or destruction). Thus all moves can be accepted by making use of a proper reweighting, as explained in Ref. [15]. The Appendix shows how to generate random numbers with the appropriate exponential distribution (16).

C. Discussion

Although the above simplified update scheme works, it turns out to have a poor efficiency. This is because of a lack of directionality: The Green operator has, on average, a probability of 1/2 to choose a left move or a right move. Therefore the Green operator propagates along the operator string like a drunk man, with a diffusionlike law. The basic creation and destruction processes correspond to the steps of a random walk.

This suggests that the efficiency of the update scheme can be improved if one can force the Green operator to move in the same direction for several iterations. The next section presents a modified version of the simplified update scheme, which allows us to control the mean length of the steps of the random walk, that is to say, the mean number of creations and destructions in a given direction. The proposed directed update scheme can be considered analogous to the “directed loop update” used in the stochastic series expansion algorithm [11,19], which prevents a worm from going backward. However, the connection should not be pushed too far. Indeed, the picture of a worm whose head is evolving in both

space and imaginary time across vertices is obvious in a loop algorithm. In such an algorithm, a creation (or an annihilation) operator which is represented by the head of a worm is propagated in both space and imaginary time, while an annihilation (or a creation) operator represented by the tail of the worm remains at rest. The loop ends when the head of the worm bites the tail.

Such a worm picture is not obvious in the SGF algorithm: Instead of single creation or annihilation operators, it is the full Green operator over the whole space that is propagated only in imaginary time. This creates open worldlines, thus introducing discontinuities. These discontinuities increase or decrease while propagating in imaginary time. All open ends of the worldlines are localized at the same imaginary time index. Therefore it is actually not possible to draw step by step a worm whose head is evolving in space and imaginary time until it bites its tail.

IV. DIRECTED UPDATE SCHEME

We present in this section a directed update scheme which is obtained by modifying slightly the simplified update scheme, thus keeping the simplicity and generality of the algorithm.

A. The update scheme

Assuming that a left move is chosen, the Green operator chooses between starting the move by a creation or a destruction. After having created (or destroyed) an operator, the Green operator can choose to keep moving in the same direction and destroy (or create) with a probability P_{\leftarrow}^{kd} (or P_{\leftarrow}^{kc}), or to stop. If it keeps moving, then a destruction (or creation) occurs, and the Green operator can choose to keep moving and create (or destroy) with a probability P_{\leftarrow}^{kc} (or P_{\leftarrow}^{kd})... and so on, until it decides to stop. If the last action of the move is a creation, then a time shift is chosen. The organigram is represented in Fig. 2.

B. Detailed balance

In order to satisfy detailed balance, in addition to the acceptance factors q_{\leftarrow}^c and q_{\leftarrow}^d , we need to determine new acceptance factors of the form $q_{\leftarrow}^{cdcdcd\dots}$ and $q_{\leftarrow}^{dcdcdcd\dots}$. We first determine the new expressions of q_{\leftarrow}^c and q_{\leftarrow}^d resulting from the directed update scheme. For q_{\leftarrow}^c , the previous probability $S_{i \rightarrow f}$ has to be multiplied by the probability to stop the move after having created, $1 - P_{\leftarrow}^{kc}(\tau')$. The previous probability $S_{f \rightarrow i}$ has to be multiplied by the probability to stop the move after having destroyed, $1 - P_{\rightarrow}^{kd}(\tau)$. We get for q_{\leftarrow}^c and q_{\leftarrow}^d the new expressions

$$q_{\leftarrow}^c = \frac{\langle \psi_L | \hat{G} \hat{T} | \psi_R \rangle [1 - P_{\leftarrow}^{kc}(\tau)]}{\langle \psi_L | \hat{G} | \psi_R \rangle P(\leftarrow) P_{\leftarrow}^{\dagger}(\tau)} \times \frac{P(\rightarrow') [1 - P_{\leftarrow}^{\dagger}(\tau')] (1 - e^{-(\tau'_L - \tau'_R)(V'_R - V'_L)})}{[1 - P_{\leftarrow}^{kd}(\tau')] (V'_R - V'_L)}, \quad (27)$$

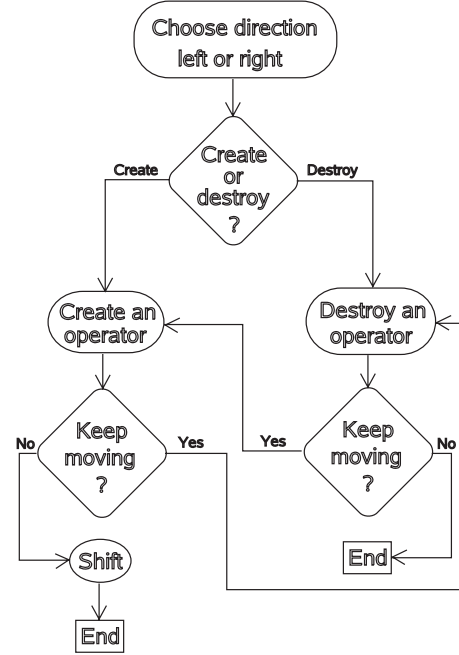


FIG. 2. Directed update scheme. See text for details.

$$q_{\leftarrow}^d = \frac{[1 - P_{\rightarrow}^{kd}(\tau)](V_L - V_R)}{P(\leftarrow) [1 - P_{\leftarrow}^{\dagger}(\tau)] (1 - e^{-(\tau_L - \tau_R)(V_L - V_R)})} \times \frac{\langle \psi'_L | \hat{G} | \psi'_R \rangle P(\rightarrow') P_{\leftarrow}^{\dagger}(\tau')}{\langle \psi'_L | \hat{T} \hat{G} | \psi'_R \rangle [1 - P_{\leftarrow}^{kc}(\tau')]}. \quad (28)$$

1. Creation, destruction

We consider here the case where a left move is chosen, an operator is created on the right of the Green operator, and a new state is chosen. Then the operator on the left of the Green operator is destroyed. Using the superscripts a, b, c, \dots to denote intermediate configurations between initial and final configurations, the sequence is the following:

- (1) $P_i \propto \langle \psi_{L+1} | \hat{T}(\tau_L) | \psi_L \rangle \langle \psi_L | \hat{G}(\tau) | \psi_R \rangle$,
- (2) $\langle \psi_{L+1}^a | \hat{T}(\tau_L^a) | \psi_L^a \rangle \langle \psi_L^a | \hat{G}(\tau^a) | \psi_R^a \rangle \langle \psi_R^a | \hat{T}(\tau_R^a) | \psi_{R-1}^a \rangle$,
- (3) $P_f \propto \langle \psi'_L | \hat{G}(\tau') | \psi'_R \rangle \langle \psi'_R | \hat{T}(\tau'_R) | \psi'_{R-1} \rangle$,

where we have $\langle \psi_{L+1} | = \langle \psi_{L+1}^a | = \langle \psi_L^a |$, $|\psi_R\rangle = |\psi_{R-1}^a\rangle = |\psi'_{R-1}\rangle$, $|\psi_L\rangle \langle \psi_L| = |\psi_L^a\rangle \langle \psi_L^a|$, and $|\psi_R^a\rangle \langle \psi_R^a| = |\psi'_R\rangle \langle \psi'_R|$. The probability of the transition from the initial configuration to the final configuration is the probability $P(\leftarrow)$ to choose a left move, times the probability $P_{\leftarrow}^{\dagger}(\tau)$ to create an operator at time τ , times the probability $P_{\leftarrow}(\psi_R^a)$ to choose the new state ψ_R^a , times the probability $P_{\leftarrow}^{kd}(a)$ to keep moving and destroy, times the probability $1 - P_{\leftarrow}^{kc}(\tau')$ to stop the move after having destroyed:

$$S_{i \rightarrow f} = P(\leftarrow) P_{\leftarrow}^{\dagger}(\tau) P_{\leftarrow}(\psi_R^a) P_{\leftarrow}^{kd}(a) [1 - P_{\leftarrow}^{kc}(\tau')]. \quad (29)$$

The probability of the reverse move is exactly symmetric:

$$S_{f \rightarrow i} = P(\rightarrow') P_{\leftarrow}^{\dagger}(\tau') P_{\leftarrow}(\psi_L^a) P_{\leftarrow}^{kd}(a) [1 - P_{\rightarrow}^{kc}(\tau)]. \quad (30)$$

It is important to notice that, when in the intermediate configuration a , the time τ_L^a of the operator to the left of the

Green operator is equal to τ_L , and the time τ_R^a of the operator to the right of the Green operator is equal to τ . Thus the acceptance factor takes the form

$$q_{\leftarrow}^{cd} = \frac{\langle \psi_L | \hat{G} \hat{T} | \psi_R \rangle [1 - P_{\leftarrow}^{kc}(\tau)] e^{-(\tau_L - \tau_R^a) V_R^a} P_{\leftarrow}^{kd}(a)}{\langle \psi_L | \hat{G} | \psi_R \rangle P(\leftarrow) P_{\leftarrow}^{\dagger}(\tau) e^{-(\tau_L - \tau_R^a) V_L} P_{\leftarrow}^{kd}(a)} \\ \times \frac{\langle \psi'_L | \hat{G} | \psi'_R \rangle P(\rightarrow') P_{\leftarrow}^{\dagger}(\tau')}{\langle \psi'_L | \hat{T} \hat{G} | \psi'_R \rangle [1 - P_{\leftarrow}^{kc}(\tau')]}, \quad (31)$$

and is written as a quantity that depends on the initial configuration, times a quantity that depends on the intermediate configuration a , times a quantity that depends on the final configuration. It is useful for the remainder of the paper to define the intermediate acceptance factor,

$$q_{\leftarrow}^{c-d}(a) = \frac{e^{-(\tau_L - \tau_R^a) V_R^a} P_{\leftarrow}^{kd}(a)}{e^{-(\tau_L - \tau_R^a) V_L} P_{\leftarrow}^{kd}(a)}. \quad (32)$$

2. Destruction, creation

We consider here the case where a left move is chosen, the operator on the left of the Green operator is destroyed, then an operator is created on its right, and a new state is chosen. Finally a time shift is chosen. The sequence of configurations is the following:

- (1) $P_i \propto \langle \psi_{L+1} | \hat{T}(\tau_L) | \psi_L \rangle \langle \psi_L | \hat{G}(\tau) | \psi_R \rangle$,
- (2) $\langle \psi'_L | \hat{G}(\tau') | \psi'_R \rangle$,
- (3) $P_f \propto \langle \psi'_L | \hat{G}(\tau') | \psi'_R \rangle \langle \psi'_R | \hat{T}(\tau'_R) | \psi'_{R-1} \rangle$,

where we have $\langle \psi_{L+1} | = \langle \psi'_L | = \langle \psi_L |$ and $|\psi_R\rangle = |\psi'_R\rangle = |\psi'_{R-1}\rangle$. The probability of the transition from the initial configuration to the final configuration is the probability $P(\leftarrow)$ to choose a left move, times the probability $1 - P_{\leftarrow}^{\dagger}(\tau)$ of no creation, times the probability $P_{\leftarrow}^{kc}(a)$ to keep moving and create, times the probability $P_{\leftarrow}(\psi'_R)$ to choose the new state ψ'_R , times the probability $1 - P_{\leftarrow}^{kd}(\tau')$ to stop the move after having destroyed, times the probability $P_{\leftarrow}^{L'R'}(\tau' - \tau'_R)$ to shift the Green operator by $\tau' - \tau'_R$:

$$S_{i \rightarrow f} = P(\leftarrow) [1 - P_{\leftarrow}^{\dagger}(\tau)] P_{\leftarrow}^{kc}(a) P_{\leftarrow}(\psi'_R) \\ \times [1 - P_{\leftarrow}^{kd}(\tau')] P_{\leftarrow}^{L'R'}(\tau' - \tau'_R). \quad (33)$$

The probability of the reverse move is exactly symmetric:

$$S_{f \rightarrow i} = P(\rightarrow') [1 - P_{\leftarrow}^{\dagger}(\tau')] P_{\leftarrow}^{kc}(a) P_{\leftarrow}(\psi_L) \\ \times [1 - P_{\leftarrow}^{kd}(\tau)] P_{\leftarrow}^{LR}(\tau_L - \tau). \quad (34)$$

The acceptance factor takes the form

$$q_{\leftarrow}^{dc} = \frac{[1 - P_{\leftarrow}^{kd}(\tau)] (V_L - V_R)}{P(\leftarrow) [1 - P_{\leftarrow}^{\dagger}(\tau)] (1 - e^{-(\tau_L - \tau_R)(V_L - V_R)}} \\ \times \frac{\langle \psi'_L | \hat{G} \hat{T} | \psi'_R \rangle P_{\leftarrow}^{kc}(a)}{\langle \psi'_L | \hat{T} \hat{G} | \psi'_R \rangle P_{\leftarrow}^{kc}(a)} \\ \times \frac{P(\rightarrow') [1 - P_{\leftarrow}^{\dagger}(\tau')] [1 - e^{-(\tau'_L - \tau'_R)(V'_R - V'_L)}]}{[1 - P_{\leftarrow}^{kd}(\tau')] (V'_R - V'_L)}, \quad (35)$$

and is written as a quantity that depends on the initial configuration, times a quantity that depends on the intermediate configuration a , times a quantity that depends on the final configuration. It is useful for the remaining of the paper to define the intermediate acceptance factor,

$$q_{\leftarrow}^{d-c}(a) = \frac{\langle \psi'_L | \hat{G} \hat{T} | \psi'_R \rangle P_{\leftarrow}^{kc}(a)}{\langle \psi'_L | \hat{T} \hat{G} | \psi'_R \rangle P_{\leftarrow}^{kc}(a)}. \quad (36)$$

3. Creation, destruction, creation

We consider here the case where a left move is chosen, an operator is created on the right of the Green operator, then the operator on its left is destroyed, then a second operator is created on its right. Finally, a time shift of the Green operator is performed. The sequence of configurations is the following:

- (1) $P_i \propto \langle \psi_{L+1} | \hat{T}(\tau_L) | \psi_L \rangle \langle \psi_L | \hat{G}(\tau) | \psi_R \rangle$,
- (2) $\langle \psi'_{L+1} | \hat{T}(\tau'_L) | \psi'_L \rangle \langle \psi'_L | \hat{G}(\tau') | \psi'_R \rangle \langle \psi'_R | \hat{T}(\tau'_R) | \psi'_{R-1} \rangle$,
- (3) $\langle \psi'_L | \hat{G}(\tau') | \psi'_R \rangle \langle \psi'_R | \hat{T}(\tau'_R) | \psi'_{R-1} \rangle$,
- (4) $P_f \propto \langle \psi'_L | \hat{G}(\tau') | \psi'_R \rangle \langle \psi'_R | \hat{T}(\tau'_R) | \psi'_{R-1} \rangle \langle \psi'_{R-1} | \hat{T}(\tau'_{R-1}) | \psi'_{R-2} \rangle$.

Considering the intermediate configurations a and b between the initial and final configurations, it is easy to show that the corresponding acceptance factor can be written

$$q_{\leftarrow}^{cdc} = q_{\leftarrow}^c \times q_{\leftarrow}^{c-d}(a) \times q_{\leftarrow}^{d-c}(b). \quad (37)$$

4. Destruction, creation, destruction

We consider here the case where a left move is chosen, the operator on the left of the Green operator is destroyed, then an operator is created on its right. Finally a second operator on the left of Green operator is destroyed. The sequence of configurations is the following

- (1) $P_i \propto \langle \psi_{L+2} | \hat{T}(\tau_{L+1}) | \psi_{L+1} \rangle \langle \psi_{L+1} | \hat{T}(\tau_L) | \psi_L \rangle \langle \psi_L | \hat{G}(\tau) | \psi_R \rangle$,
- (2) $\langle \psi'_{L+1} | \hat{T}(\tau'_L) | \psi'_L \rangle \langle \psi'_L | \hat{G}(\tau') | \psi'_R \rangle$,
- (3) $\langle \psi'_{L+1} | \hat{T}(\tau'_L) | \psi'_L \rangle \langle \psi'_L | \hat{G}(\tau') | \psi'_R \rangle \langle \psi'_R | \hat{T}(\tau'_R) | \psi'_{R-1} \rangle$,
- (4) $P_f \propto \langle \psi'_L | \hat{G}(\tau') | \psi'_R \rangle \langle \psi'_R | \hat{T}(\tau'_R) | \psi'_{R-1} \rangle$.

Considering the intermediate configurations a and b between the initial and final configurations, it is easy to show that the corresponding acceptance factor can be written

$$q_{\leftarrow}^{dcd} = q_{\leftarrow}^d \times q_{\leftarrow}^{d-c}(a) \times q_{\leftarrow}^{c-d}(b). \quad (38)$$

5. Generalization

It is straightforward to show that the acceptance factors of the form q_{\leftarrow}^{cdcdcd} , $q_{\leftarrow}^{cdcdcdcd}$, $q_{\leftarrow}^{cdcdcdcdcd}$, ... (or $q_{\leftarrow}^{dcdcdcd}$, $q_{\leftarrow}^{dcdcdcdcd}$, $q_{\leftarrow}^{dcdcdcdcdcd}$, ...) can be expressed as products of the acceptance factor q_{\leftarrow}^c (or q_{\leftarrow}^d) and the intermediate factors q_{\leftarrow}^{c-d} and q_{\leftarrow}^{d-c} .

In the same manner, the acceptance factors of the form q_{\leftarrow}^{cdcd} , q_{\leftarrow}^{cdcdcd} , $q_{\leftarrow}^{cdcdcdcd}$, ... (or q_{\leftarrow}^{dcdcd} , $q_{\leftarrow}^{dcdcdcd}$, $q_{\leftarrow}^{dcdcdcdcd}$, ...) can be expressed as products of the acceptance factor q_{\leftarrow}^{cd} (or q_{\leftarrow}^{dc}) and the intermediate factors q_{\leftarrow}^{c-d} and q_{\leftarrow}^{d-c} .

6. Simplification of the acceptance factors

Here again it is possible to take advantage of the freedom that we have for the choice of the probabilities $P(\leftarrow)$, P_{\leftarrow}^{\dagger} ,

P_{\leftarrow}^{kc} , and P_{\leftarrow}^{kd} [or $P(\leftarrow)$, $P_{\rightarrow}^{\dagger}$, P_{\leftarrow}^{kc} , and P_{\leftarrow}^{kd}]. A proper choice of these probabilities can be done in order to allow us to accept all moves, simplicity and generality being the leitmotif of the SGF algorithm.

For this purpose, we impose to all acceptance factors corresponding to left (or right) moves to be equal. This requires the intermediate acceptance factors q_{\leftarrow}^{c-d} and q_{\leftarrow}^{d-c} (or q_{\rightarrow}^{c-d} and q_{\rightarrow}^{d-c}) to be equal to 1. This is realized if

$$P_{\leftarrow}^{kc} = \alpha_c \min\left(1, \frac{\langle \hat{\mathcal{G}}\hat{\mathcal{T}} \rangle}{\langle \hat{\mathcal{T}}\hat{\mathcal{G}} \rangle}\right), \quad (39)$$

$$P_{\rightarrow}^{kc} = \alpha_c \min\left(1, \frac{\langle \hat{\mathcal{T}}\hat{\mathcal{G}} \rangle}{\langle \hat{\mathcal{G}}\hat{\mathcal{T}} \rangle}\right), \quad (40)$$

$$P_{\leftarrow}^{kd} = \alpha_d \min(1, e^{-(\tau_L^k - \tau_R^k)(V_R^k - V_L^k)}), \quad (41)$$

$$P_{\rightarrow}^{kd} = \alpha_d \min(1, e^{-(\tau_L^k - \tau_R^k)(V_L^k - V_R^k)}), \quad (42)$$

where α_c and α_d are optimization parameters belonging to $[0;1]$. By tuning these parameters, the mean length of the steps of the Green operator can be controlled. Note that we have explicitly excluded 1 from the allowed values for these optimization parameters. This is necessary for the Green operator to have a chance to end in a diagonal configuration, $|\psi_L\rangle = |\psi_R\rangle$. Indeed, the choice $\alpha_c = \alpha_d = 1$ would systematically lead to values of 1 for the probabilities P^{kc} and P^{kd} for diagonal configurations. Therefore the Green operator would never stop in a diagonal configuration, and no measurement could be done. It is important here to note that the quantities $\langle \hat{\mathcal{G}} \rangle$, $\langle \hat{\mathcal{G}}\hat{\mathcal{T}} \rangle$, and $\langle \hat{\mathcal{T}}\hat{\mathcal{G}} \rangle$ are evaluated between the states on the left and the right of the Green operator that are present at the moment where those quantities are needed, as well as for the times indices τ_L^k and τ_R^k and the potentials V_L^k and V_R^k .

All acceptance factors corresponding to a given direction of propagation become equal if we choose for the creation probabilities

$$P_{\leftarrow}^{\dagger}(\tau) = \frac{\langle \hat{\mathcal{G}}\hat{\mathcal{T}} \rangle}{\langle \hat{\mathcal{G}}\hat{\mathcal{T}} \rangle + \langle \hat{\mathcal{G}} \rangle \frac{(1 - P_{\leftarrow}^{kd})(V_L - V_R)}{(1 - P_{\leftarrow}^{kc})(1 - e^{-(\tau_L - \tau_R)(V_L - V_R)})}}, \quad (43)$$

$$P_{\rightarrow}^{\dagger}(\tau) = \frac{\langle \hat{\mathcal{T}}\hat{\mathcal{G}} \rangle}{\langle \hat{\mathcal{T}}\hat{\mathcal{G}} \rangle + \langle \hat{\mathcal{G}} \rangle \frac{(1 - P_{\leftarrow}^{kd})(V_R - V_L)}{(1 - P_{\leftarrow}^{kc})(1 - e^{-(\tau_L - \tau_R)(V_R - V_L)})}}. \quad (44)$$

Finally, all acceptance factors become independent of the direction of propagation if we choose $P(\leftarrow) = r_{\leftarrow}(\tau) / [r_{\leftarrow}(\tau) + r_{\rightarrow}(\tau)]$ and $P(\rightarrow) = r_{\rightarrow}(\tau) / [r_{\leftarrow}(\tau) + r_{\rightarrow}(\tau)]$ with

$$r_{\leftarrow}(\tau) = (1 - P_{\leftarrow}^{kc}) \frac{\langle \hat{\mathcal{G}}\hat{\mathcal{T}} \rangle}{\langle \hat{\mathcal{G}} \rangle} + \frac{(1 - P_{\leftarrow}^{kd})(V_L - V_R)}{(1 - e^{-(\tau_L - \tau_R)(V_L - V_R)})}, \quad (45)$$

$$r_{\rightarrow}(\tau) = (1 - P_{\leftarrow}^{kc}) \frac{\langle \hat{\mathcal{T}}\hat{\mathcal{G}} \rangle}{\langle \hat{\mathcal{G}} \rangle} + \frac{(1 - P_{\leftarrow}^{kd})(V_R - V_L)}{(1 - e^{-(\tau_L - \tau_R)(V_R - V_L)})}. \quad (46)$$

As a result all moves can be accepted again, ensuring the maximum of simplicity of the algorithm. We still have some freedom in the choice of the optimization parameters α_c and α_d . This is discussed in the next section.

V. TEST AND OPTIMIZATION OF THE ALGORITHM

From the central limit theorem, we know that the error bar associated with any measured quantity must decrease as the square root of the number of measurements, or equivalently, as the square root of the time of the simulation. Therefore it makes sense to define the efficiency \mathcal{E} of a QMC algorithm by

$$\mathcal{E}(\Omega, \mathcal{O}) = \frac{1}{T(\Omega)[\Delta\mathcal{O}(\Omega)]^2}, \quad (47)$$

where Ω represents the set of all optimization parameters of the algorithm, \mathcal{O} is the measured quantity of interest, $T(\Omega)$ is the time of the simulation, and $\Delta\mathcal{O}(\Omega)$ is the error bar associated with the measured quantity \mathcal{O} . This definition ensures that \mathcal{E} is independent of the time of the simulation. As a result, the larger \mathcal{E} , the more efficient the algorithm.

In the present case we have $\Omega = \{\alpha_c, \alpha_d\}$, while $\Omega = \emptyset$ for the original SGF algorithm. It is useful here to realize that, by symmetry, the mean values of P_{\leftarrow}^{kc} and P_{\leftarrow}^{kd} (and P_{\leftarrow}^{kc} and P_{\leftarrow}^{kd}) must be equal. Therefore we define $P^{kc} = \langle P_{\leftarrow}^{kc} \rangle = \langle P_{\leftarrow}^{kd} \rangle$ and $P^{kd} = \langle P_{\leftarrow}^{kd} \rangle = \langle P_{\leftarrow}^{kc} \rangle$. It seems reasonable to impose a condition of uniform sampling, $P^{kc} = P^{kd}$. This condition can be satisfied by adjusting dynamically the values of α_c and α_d during the thermalization process. For this purpose we introduce a new optimization parameter $\alpha \in [0;1]$ and apply the following algorithm from time to time while thermalizing (we start with $\alpha_c = \alpha_d = \alpha$):

Evaluate P^{kc} and P^{kd} over few iterations

If $P^{kc} < P^{kd}$

then $\alpha_d \rightarrow \alpha_d \frac{P^{kc}}{P^{kd}}$

else $\alpha_c \rightarrow \alpha_c \frac{P^{kd}}{P^{kc}}$.

If $\alpha_c < \alpha_d$

then $\alpha_c = \frac{\alpha}{\alpha_d}, \alpha_d = \alpha$

else $\alpha_d = \frac{\alpha}{\alpha_c}, \alpha_c = \alpha$.

TABLE I. Relative efficiency of the algorithm applied to $\hat{\mathcal{H}}_1$ at half filling for the energy, the superfluid density, and the number of particles in the zero-momentum state.

α	$\langle S(\alpha) \rangle$	$\mathcal{E}(\alpha, E)$	$\mathcal{E}(\alpha, \rho_s)$	$\mathcal{E}(\alpha, n(0))$
0	1.00	0.307400	0.487457	0.503105
0.1	1.10	0.774161	0.513633	0.805048
0.5	1.91	0.430843	3.771422	1.289757
0.9	7.00	0.977413	5.400997	6.629893
0.95	10.49	2.427874	10.688100	7.994883
0.99	17.49	1.286403	27.281408	1.327064
0.9999	20.93	0.818048	17.510068	1.059823
0.99999	21.00	0.710448	13.353809	0.779245

Thus we are left with the optimization parameter α . In order to determine the optimal value, we have considered two different Hamiltonians $\hat{\mathcal{H}}_1$ and $\hat{\mathcal{H}}_2$, and evaluated the efficiency of the algorithm while scanning α . The first Hamiltonian we have considered describes free hardcore bosons and is exactly solvable,

$$\hat{\mathcal{H}}_1 = -t \sum_{\langle i,j \rangle} (a_i^\dagger a_j + a_j^\dagger a_i), \quad (48)$$

where the sum runs over pairs of first neighboring sites and t is the hopping parameter. The second Hamiltonian is highly nontrivial and describes a mixture of atoms and diatomic molecules, with a special term allowing conversions between the two species [20],

$$\begin{aligned} \hat{\mathcal{H}}_2 = & -t_a \sum_{\langle i,j \rangle} (a_i^\dagger a_j + a_j^\dagger a_i) - t_m \sum_{\langle i,j \rangle} (m_i^\dagger m_j + m_j^\dagger m_i) \\ & + U_{aa} \sum_i \hat{n}_i^a (\hat{n}_i^a - 1) + U_{mm} \sum_i \hat{n}_i^m (\hat{n}_i^m - 1) + U_{am} \sum_i \hat{n}_i^a \hat{n}_i^m \\ & + D \sum_i \hat{n}_i^m + g \sum_i (m_i^\dagger a_i a_i + a_i^\dagger a_i m_i), \end{aligned} \quad (49)$$

where a_i^\dagger and a_i (m_i^\dagger and m_i) are the creation and annihilation operators of atoms (molecules), t_a , t_m , U_{aa} , U_{mm} , and U_{am} are, respectively, the hopping parameter of atoms, the hopping parameter of molecules, the atomic on-site interaction parameter, the molecular on-site interaction parameter, and the interspecies interaction parameter. The conversion term is

TABLE II. Relative efficiency of the algorithm applied to $\hat{\mathcal{H}}_2$ for the energy and the density of atoms and molecules.

α	$\langle S(\alpha) \rangle$	$\mathcal{E}(\alpha, E)$	$\mathcal{E}(\alpha, \rho_a)$	$\mathcal{E}(\alpha, \rho_m)$
0	1.00	1.086334	0.455569	1.670239
0.1	1.10	1.424308	0.506873	1.858339
0.5	1.88	2.813905	1.265620	4.640123
0.9	6.35	2.562529	5.999027	21.993900
0.95	8.99	2.335315	3.917233	14.361774
0.99	12.79	2.592328	1.721519	6.311612

TABLE III. Relative efficiency of the algorithm applied to $\hat{\mathcal{H}}_2$ for the occupation of the zero-momentum state and the visibility of atoms and molecules.

α	$\mathcal{E}(\alpha, n_a(0))$	$\mathcal{E}(\alpha, n_m(0))$	$\mathcal{E}(\alpha, \mathcal{V}_a)$	$\mathcal{E}(\alpha, \mathcal{V}_m)$
0	0.433382	0.234412	1.323720	0.239113
0.1	0.269700	0.181019	0.585183	0.248060
0.5	1.752466	2.806166	2.667114	1.357462
0.9	7.080124	5.638859	16.454676	4.482435
0.95	4.893878	3.757436	5.088775	2.248427
0.99	3.871723	2.341222	7.783268	1.279447

tunable via the parameter g and does not conserve the number N_a of atoms or the number N_m of molecules. However, the total number of particles $N = N_a + 2N_m$ is conserved and is the canonical constraint. The parameter D allows to control the ratio between the number of atoms and molecules. The application of the SGF algorithm to the Hamiltonian (49) is described in detail in Ref. [15]. The changes coming with the directed update scheme are completely independent of the chosen Hamiltonian.

Table I shows the mean number of creations and destructions in one step, $\langle S(\alpha) \rangle$, and the relative efficiency $\mathcal{E}(\alpha, \mathcal{O})/\mathcal{E}(\emptyset, \mathcal{O})$ of the algorithm applied to $\hat{\mathcal{H}}_1$ at half filling, for which we have measured the energy E , the superfluid density ρ_s , and the number of particles in the zero momentum state $n(k=0)$.

For $\hat{\mathcal{H}}_2$, we have used the parameters $t_a=1$, $t_m=1/2$, $U_{aa}=5$, $U_{mm}=5$, $U_{am}=5$, $g=5$, $D=3$, and a density of particles $\rho=2$. Tables II and III show $\langle S(\alpha) \rangle$, and the relative efficiency of the algorithm for the energy E , the density of atoms and molecules ρ_a and ρ_m , the occupation of the zero momentum state for atoms and molecules $n_a(0)$ and $n_m(0)$, and the atomic and molecular visibilities \mathcal{V}_a and \mathcal{V}_m .

While the best value of α depends on the Hamiltonian that is considered and the measured quantity, it appears that a

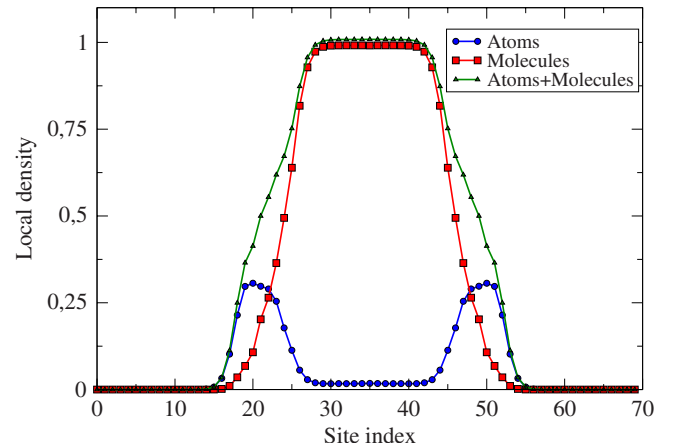


FIG. 3. (Color online) An example of density profiles obtained when the trapping potential (50) is added to the Hamiltonian (49). The error bars are smaller than the symbol sizes, and are the biggest in the neighborhood of site indices 23 and 47, where they equal the size of the symbols.

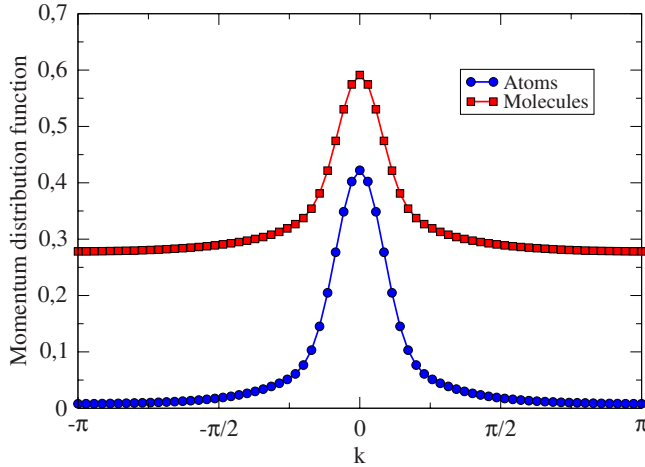


FIG. 4. (Color online) An example of momentum distribution functions obtained when the trapping potential (50) is added to the Hamiltonian (49). The error bars are smaller than the symbol sizes, and are the biggest for $k=0$, where they equal the size of the symbols.

good compromise is to choose α between 0.90 and 0.99. The improvement of the efficiency is remarkable. In the following, we illustrate the applicability of the algorithm to problems with nonuniform potentials, by adding a parabolic trap to the Hamiltonian (49):

$$\hat{\mathcal{H}}_T = W_a \sum_i (i - L/2)^2 \hat{n}_i^a + W_m \sum_i (i - L/2)^2 \hat{n}_i^m. \quad (50)$$

The parameters W_a and W_m allow us to control the curvature of the trap associated with atoms and molecules, respectively, and L is the number of lattice sites. The inclusion of this term in the algorithm is trivial since only the values of the diagonal energies V_L and V_R are changed. Figures 3 and 4 show the density profiles and momentum distribution functions obtained for a system with $L=70$ lattice sites initially loaded with 50 atoms and no molecules, and the parameters $t_a=1$, $t_m=0.5$, $U_{aa}=4$, $U_{am}=12$, $U_{mm}=\infty$, $g=0.5$, $D=0$, $W_a=0.008$, $W_m=0.008$, and $\beta=20$. The presented results were obtained by performing 10^4 updates for thermalization, and 2×10^4 updates with measurements (an update is to be understood as the occurrence of a diagonal configuration). The time of the simulation is about 8 h on a cheap 32-bit laptop with 1 GHz processor, with an implementation of the algorithm involving dynamical structures with pointers (see Ref. [15]).

VI. CONCLUSION

We have presented a directed update scheme for the SGF algorithm, which has the properties of keeping the simplicity

and generality of the original algorithm, and significantly improving its efficiency.

ACKNOWLEDGMENTS

I would like to express special thanks to Peter Denteneer for useful suggestions. This work is part of the research program of the ‘‘Stichting voor Fundamenteel Onderzoek der Materie (FOM),’’ which is financially supported by the ‘‘Nederlandse Organisatie voor Wetenschappelijk Onderzoek (NWO).’’

APPENDIX: EXPONENTIAL RANDOM NUMBER GENERATOR

We describe here how to generate numbers with the appropriate exponential distribution (16). Assuming that we have at our disposal a uniform random number generator that generates a random variable U with the distribution $\rho_U(u) = 1$ for $u \in [0; 1[$, we would like to find a function f such that the random variable $T=f(U)$ is generated with the distribution

$$\rho_T^{\Delta\tau, \Delta V}(\tau) = \frac{\Delta V e^{-\tau\Delta V}}{1 - e^{-\Delta\tau\Delta V}}, \quad \tau \in [0; \Delta\tau[, \quad (A1)$$

where $\Delta\tau$ and ΔV are the parameters of the exponential distribution. Because of the relation $T=f(U)$, the probability to find T in the range $[\tau; \tau+d\tau[$ must be equal to the probability to find U in the range $[u; u+du[$. This implies the condition

$$\rho_U(u)|du| = \rho_T^{\Delta\tau, \Delta V}(\tau)|d\tau|, \quad (A2)$$

with $|d\tau/du| = \pm df/du$. Thus we have

$$\frac{\Delta V e^{-f(u)\Delta V} df}{1 - e^{-\Delta\tau\Delta V} du} = \pm 1. \quad (A3)$$

Taking the antiderivative with respect to u on both sides of the equation, we get

$$\frac{-e^{-f(u)\Delta V}}{1 - e^{-\Delta\tau\Delta V}} = \pm (u + C), \quad (A4)$$

where C is a constant. This constant and the correct sign are determined by imposing the conditions $f(0)=0$ and $f(1)=\Delta\tau$. As a result, if u is a realization of U , then a realization of T is given by

$$\tau = -\frac{1}{\Delta V} \ln[1 - u(1 - e^{-\Delta\tau\Delta V})]. \quad (A5)$$

- [1] N. Metropolis and S. Ulam, *J. Am. Stat. Assoc.* **44**, 335 (1949).
- [2] D. C. Handscomb, *Proc. Cambridge Philos. Soc.* **58**, 594 (1962).
- [3] M. H. Kalos, *Phys. Rev.* **128**, 1791 (1962).
- [4] R. Blankenbecler, D. J. Scalapino, and R. L. Sugar, *Phys. Rev. D* **24**, 2278 (1981).
- [5] G. G. Batrouni and R. T. Scalettar, *Phys. Rev. B* **46**, 9051 (1992).
- [6] W. von der Linden, *Phys. Rep.* **220**, 53 (1992).
- [7] H. G. Evertz, G. Lana, and M. Marcu, *Phys. Rev. Lett.* **70**, 875 (1993).
- [8] D. M. Ceperley, *Rev. Mod. Phys.* **67**, 279 (1995).
- [9] B. B. Beard and U.-J. Wiese, *Phys. Rev. Lett.* **77**, 5130 (1996).
- [10] *Quantum Monte Carlo Methods in Physics and Chemistry*, edited by M. P. Nightingale and C. J. Umrigar, NATO Science Series C Vol. 525 (Kluwer Academic, Dordrecht, 1999).
- [11] A. W. Sandvik, *J. Phys. A* **25**, 3667 (1992); *Phys. Rev. B* **59**, R14157 (1999).
- [12] N. V. Prokof'ev, B. V. Svistunov, and I. S. Tupitsyn, *JETP Lett.* **87**, 310 (1998).
- [13] M. Rigol, A. Muramatsu, G. G. Batrouni, and R. T. Scalettar, *Phys. Rev. Lett.* **91**, 130403 (2003).
- [14] K. Van Houcke, S. M. A. Rombouts, and L. Pollet, *Phys. Rev. E* **73**, 056703 (2006).
- [15] V. G. Rousseau, *Phys. Rev. E* **77**, 056705 (2008).
- [16] A. W. Sandvik, S. Daul, R. R. P. Singh, and D. J. Scalapino, *Phys. Rev. Lett.* **89**, 247201 (2002).
- [17] V. G. Rousseau, R. T. Scalettar, and G. G. Batrouni, *Phys. Rev. B* **72**, 054524 (2005).
- [18] N. Metropolis, A. W. Rosenbluth, M. N. Metropolis, A. H. Teller, and E. Teller, *J. Chem. Phys.* **21**, 1087 (1953).
- [19] Olav F. Syljuasen and Anders W. Sandvik, *Phys. Rev. E* **66**, 046701 (2002).
- [20] V. G. Rousseau and P. J. H. Denteneer, *Phys. Rev. A* **77**, 013609 (2008).