# Monte Carlo Eigenvalue Methods in Quantum Mechanics and Statistical Mechanics [*]

M. P. Nightingale

*Department of Physics, University of Rhode Island,*
*Kingston, RI 02881.*

C. J. Umrigar

*Cornell Theory Center and Laboratory of Atomic and Solid State Physics,*
*Cornell University, Ithaca, NY 14853.*

(February 1, 2008)

In this review we discuss, from a unified point of view, a variety of Monte Carlo methods used to solve eigenvalue problems in statistical mechanics and quantum mechanics. Although the applications of these methods differ widely, the underlying mathematics is quite similar in that they are stochastic implementations of the power method. In all cases, optimized trial states can be used to reduce the errors of Monte Carlo estimates.

## Contents

## I. INTRODUCTION

Many important problems in computational physics and chemistry can be reduced to the computation of dominant eigenvalues of matrices of high or infinite order. We shall focus on just a few of the numerous examples of such matrices, namely, quantum mechanical Hamiltonians, Markov matrices and transfer matrices. Quantum Hamiltonians, unlike the other two, probably can do without introduction. Markov matrices are used both in equilibrium and nonequilibrium statistical mechanics to describe dynamical phenomena. Transfer matrices were introduced by Kramers and Wannier in 1941 to study the two-dimensional Ising model,[1] and ever since, important work on lattice models in classical statistical mechanics has been done with transfer matrices, producing both exact and numerical results.[2]

The basic Monte Carlo methods reviewed in this chapter have been used in many different contexts and under many different names for many decades, but we emphasize the solution of eigenvalue problems by means of Monte Carlo methods and present the methods from a unified point of view. A vital ingredient in the methods discussed here is the use of optimized trial functions. Section IV deals with this topic briefly, but in general we suppose that optimized trial functions are given. We refer the reader to Ref. 3 for more details on their construction.

The analogy of the time-evolution operator in quantum mechanics on the one hand, and the transfer matrix and the Markov matrix in statistical mechanics on the other, allows the two fields to share numerous techniques. Specifically, a transfer matrix $G$ of a statistical mechanical lattice system in $d$ dimensions often can be interpreted as the evolution operator in discrete, imaginary time $t$ of a quantum mechanical analog in $d-1$ dimensions. That is, $G \approx \exp(-t\mathcal{H})$, where $\mathcal{H}$ is the Hamiltonian of a system in $d-1$ dimensions, the quantum mechanical analog of the statistical mechanical system. From this point of view, the computation of the partition function and of the ground-state energy are essentially the same problems: finding the largest eigenvalue of $G$ and of $\exp(-t\mathcal{H})$, respectively. As far as the Markov matrix is concerned, this simply is the time-evolution operator of a system evolving according to stochastic dynamics. The largest eigenvalue of such matrices equals unity, as follows from conservation of probability, and for systems in thermal equilibrium, the corresponding eigenstate is also known, namely the Boltzmann distribution. Clearly,

the dominant eigenstate in this case poses no problem. For nonequilibrium systems, the stationary state is unknown and one might use the methods described in this chapter in dealing with them. Another problem is the computation of the relaxation time of a system with stochastic dynamics. This problem is equivalent to the computation of the second largest eigenvalue of the Markov matrix, and again the current methods apply.

The emphasis of this chapter is on methods rather than applications, but the reader should have a general idea of the kind of problems for which these methods can be employed. Therefore, we start off by giving some specific examples of the physical systems one can deal with.

## A. Quantum Systems

In the case of a quantum mechanical system, the problem in general is to compute expectation values, in particular the energy, of bosonic or fermionic ground or excited eigenstates. For systems with $n$ electrons, the spatial coordinates are denoted by a $3n$-dimensional vector $\mathbf{R}$. In terms of the vectors $\mathbf{r}_i$ specifying the coordinates of electron number $i$ this reads $\mathbf{R} = (\mathbf{r}_1, \ldots, \mathbf{r}_n)$. The dimensionless Hamiltonian is of the form

$$\langle \mathbf{R}|\mathcal{H}|\mathbf{R}'\rangle = [-\frac{1}{2\mu}\vec{\nabla}^2 + \mathcal{V}(\mathbf{R})]\delta(\mathbf{R} - \mathbf{R}'). \tag{1}$$

For atoms or molecules atomic units are used $\mu = 1$, and $\mathcal{V}$ is the usual Coulomb potential acting between the electrons and between the electrons and nuclei *i.e.,*

$$\mathcal{V}(\mathbf{R}) = \sum_{i<j} \frac{1}{r_{ij}} - \sum_{\alpha,i} \frac{Z_\alpha}{r_{\alpha i}} \tag{2}$$

where for arbitrary subscripts $a$ and $b$ we define $r_{ab} = |\mathbf{r}_a - \mathbf{r}_b|$; indices $i$ and $j$ label the electrons, and we assume that the nuclei are of infinite mass and that nucleus $\alpha$ has charge $Z_\alpha$ and is located at position $\mathbf{r}_\alpha$.

In the case of quantum mechanical van der Waals clusters,[4,5] $\mu$ is the reduced mass — $\mu = 2^{\frac{1}{3}}m\epsilon\sigma/\hbar^2$ in terms of the mass $m$, Planck's constant $\hbar$ and the conventional Lennard-Jones parameters $\epsilon$ and $\sigma$ — and the potential is given by

$$\mathcal{V}(\mathbf{R}) = \sum_{i<j} \frac{1}{r_{ij}^{12}} - \frac{2}{r_{ij}^6} \tag{3}$$

The quantum nature of the system increases with $1/\mu^2$, which is proportional to the conventional de Boer parameter.

The ground-state wavefunction of a bosonic system is positive everywhere, which is very convenient in a Monte Carlo context and allows one to obtain results with an accuracy that is limited only by practical considerations. For fermionic systems, the ground-state wave function has nodes, and this places more fundamental limits on the accuracy one can obtain with reasonable effort. In the methods discussed in this chapter, this bound on the accuracy takes the form of the so-called *fixed-node approximation.* Here one assumes that the nodal surface is given, and computes the ground-state wave function subject to this constraint.

The time-evolution operator $\exp(-\tau\mathcal{H})$ in the position representation is the Green function

$$G(\mathbf{R}', \mathbf{R}, \tau) = \langle \mathbf{R}' | e^{-\tau\mathcal{H}} | \mathbf{R} \rangle. \tag{4}$$

For both bosonic systems and fermionic systems in the fixed-node approximation, $G$ has only nonnegative elements. This is essential for the Monte Carlo methods discussed here. A problem specific to quantum mechanical systems is that $G$ is only known asymptotically for short times, so that the finite-time Green function has to be constructed by the application of the generalized Trotter formula[6,7], *i.e.*, $G(\tau) = \lim_{m\to\infty} G(\tau/m)^m$, where the position variables of $G$ have been suppressed.

## B. Transfer Matrices

Our next example is the transfer matrix of statistical mechanics. The largest eigenvalue yields the free energy, from which all thermodynamic properties follow. As a typical transfer matrix, one can think of the one-site, Kramers-Wannier transfer matrix for a two-dimensional model of Ising spins, $s_i = \pm 1$. Such a matrix takes a particularly simple form for a square lattice wrapped on a cylinder with helical boundary conditions with pitch one. This produces a mismatch of one lattice site for a path on the lattice around the cylinder. This geometry has the advantage that a two-dimensional lattice can be built one site at a time and that the process of adding each single site is identical each time. Suppose we choose a lattice of $M$ sites, wrapped on a cylinder with a circumference of $L$ lattice spaces. Imagine that we are adding sites so that the lattice grows toward the left. We can then define a conditional partition function $Z_M(\mathbf{S})$, which is a sum over those states (also referred to as configurations) for which the left-most edge of the lattice is in a given state $\mathbf{S}$. The physical interpretation of $Z_M(\mathbf{S})$ is the relative probability of finding the left-most edge in a state $\mathbf{S}$ with which the rest of the lattice to its right is in thermal equilibrium.

If one has helical boundary conditions and spins that interact only with their nearest neighbors, one can repeatedly add just a single site and the bonds connecting it to its neighbors above and to the right. Analogously, the transfer matrix $G$ can be used to compute recursively the conditional partition function of a lattice with one additional site

$$Z_{M+1}(\mathbf{S}') = \sum_{\mathbf{S}} G(\mathbf{S}'|\mathbf{S}) Z_M(\mathbf{S}), \tag{5}$$

with

$$G(\mathbf{S}'|\mathbf{S}) = \exp[K(s_1' s_1 + s_1' s_L)] \prod_{i=2}^{L} \delta_{s_i', s_{i-1}}, \tag{6}$$

with $\mathbf{S} = (s_1, s_2, \ldots, s_L)$ and $\mathbf{S}' = (s_1', s_2', \ldots, s_L')$, and the $s_i, s_i' = \pm 1$ are Ising spins. With this definition of the transfer matrix, the matrix multiplication in Eq. (5) accomplishes the following: (1) a new site, labeled 1, is appended to the lattice at the left edge; (2) the Boltzmann weight is updated to that of the lattice with increased size; (3) the old site $L$ is thermalized; and finally (4) old sites $1, \ldots, L-1$ are pushed down on the stack and are renamed to $2, \ldots, L$. Sites have to remain in the stack until all interacting sites have been
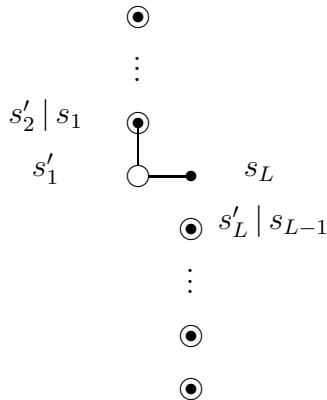
FIG. 1. Graphical representation of the transfer matrix. The primed variables are associated with the circles and combine into the left index of the matrix; the dots go with the right index and the unprimed variables. Coincidence of a circle and a dot produces a $\delta$-function. An edge indicates a contribution to the Boltzmann weight. Repeated application of this matrix constructs a lattice with nearest neighbor bonds and helical boundary conditions.

added, which determines the minimal depth of the stack. It is clear from Figure 1 that the transfer matrix is nonsymmetric and indeed symmetry is not required for the methods discussed in this chapter. It is of some interest that transfer matrices usually have the property that a right eigenvector can be transformed into a left eigenvector by a simple permutation and reweighting transformation. The details are not important here and let it suffice to mention that this follows from an obvious symmetry of the diagram shown in Figure 1: (1) rotate over $\pi$ about an axis perpendicular to the paper, which permutes the states; and (2) move the vertical bond back to its original position, which amounts to reweighting by the Boltzmann weight of a single bond.

Equation (5) implies that for large $M$ and generic boundary conditions at the right-hand edge of the lattice, the partition function approaches a dominant right eigenvector $\psi_0$ of the transfer matrix $G$

$$Z_M(\mathbf{S}) \propto \lambda_0^M \psi_0(\mathbf{S}), \tag{7}$$

where $\lambda_0$ is the dominant eigenvalue. Consequently, for $M \to \infty$ the free energy per site is given by

$$f = -kT \ln \lambda_0. \tag{8}$$

The problem relevant to this chapter is the computation of the eigenvalue $\lambda_0$ by Monte Carlo.[8]

## C. Markov Matrices

Discrete-time Markov processes are a third type of problem we shall discuss. One of the challenges in this case is to compute the correlation time of such a process in the vicinity of a critical point, where the correlation time goes to infinity, a phenomenon called "critical

slowing down". Computationally, the problem amounts to the evaluation of the second largest eigenvalue of the Markov matrix, or more precisely its difference from unity. The latter goes to zero as the correlation time approaches infinity.

The Markov matrix defines the stochastic evolution of the system in discrete time. That is, suppose that at time $t$ the probability of finding the system in state $\mathbf{S}$ is given by $\rho_t(\mathbf{S})$. If the probability of making a transition from state $\mathbf{S}$ to state $\mathbf{S}'$ is $\hat{P}(\mathbf{S}'|\mathbf{S})$ (sorry about the hat, we shall take it off soon!), then

$$\rho_{t+1}(\mathbf{S}') = \sum_{\mathbf{S}} \hat{P}(\mathbf{S}'|\mathbf{S})\rho_t(\mathbf{S}) \tag{9}$$

In the case of interest here, the Markov matrix $\hat{P}$ is constructed so that its stationary state is the Boltzmann distribution $\psi_{\mathrm{B}}^2 = \exp(-\beta\mathcal{H})$. Sufficient conditions are that (a) each state can be reached from every state in a finite number of transitions and that (b) $\hat{P}$ satisfies detailed balance

$$\hat{P}(\mathbf{S}'|\mathbf{S})\psi_{\mathrm{B}}(\mathbf{S})^2 = \hat{P}(\mathbf{S}|\mathbf{S}')\psi_{\mathrm{B}}(\mathbf{S}')^2. \tag{10}$$

It immediately follows from detailed balance that the matrix $\hat{P}$ defined by

$$P(\mathbf{S}'|\mathbf{S}) = \frac{1}{\psi_{\mathrm{B}}(\mathbf{S}')}\hat{P}(\mathbf{S}'|\mathbf{S})\psi_{\mathrm{B}}(\mathbf{S}). \tag{11}$$

is symmetric and equivalent by similarity transformation to the original Markov matrix. Because of this symmetry, expressions tend to take a simpler form when $P$ is used, as we shall do, but one should keep in mind that $P$ itself is not a Markov matrix, since the sum on its left index does not yield unity identically.

Again, to provide a specific example, we mention that the methods discussed below have been applied[9,10] to an Ising model on a square lattice with the heat bath or Yang[11] transition probabilities and random site selection. In that case, time evolution takes place by single spin-flip transitions which occur between a given state $\mathbf{S}$ to one of the states $\mathbf{S}'$ that differ only at one randomly selected site. For any such pair of states, the transition probability is given by

$$\hat{P}(\mathbf{S}'|\mathbf{S}) = \begin{cases} \frac{1}{2N}\frac{e^{-\frac{1}{2}\beta\Delta\mathcal{H}}}{\cosh\frac{1}{2}\beta\Delta\mathcal{H}} & \text{for } \mathbf{S}' \neq \mathbf{S} \\ 1 - \sum_{\mathbf{S}''\neq\mathbf{S}}\hat{P}(\mathbf{S}''|\mathbf{S}) & \text{for } \mathbf{S}' = \mathbf{S}, \end{cases} \tag{12}$$

for a system of $N$ sites with Hamiltonian

$$-\beta\mathcal{H} = K\sum_{(i,j)} s_i s_j + K'\sum_{(i,j)'} s_i s_j , \tag{13}$$

where $(i,j)$ denotes nearest-neighbor pairs, and $(i,j)'$ denotes next-nearest-neighbor pairs and $\Delta\mathcal{H} \equiv \mathcal{H}(\mathbf{S}') - \mathcal{H}(\mathbf{S})$.

We note that whereas the transfer matrix is used to deal with systems that are infinite in one direction, the systems used in the dynamical computations are of finite spatial dimensions only.

## II. THE POWER METHOD

Before discussing technical details of Monte Carlo methods to compute eigenvalues and expectation values, we introduce the mathematical ideas and the types of expressions for which statistical estimates are sought. We formulate the problem in terms of an operator $G$ of which one wants to compute the dominant eigenstate and eigenvalue, $|\psi_0\rangle$ and $\lambda_0$. Mathematically, but not necessarily in a Monte Carlo setting, dominant may mean dominant relative to eigenstates of a given symmetry only.

The methods to be discussed are variations of the power method, which relies on the fact that for a generic initial state $|u^{(0)}\rangle$ of the appropriate symmetry, the states $|u^{(t)}\rangle$ defined by

$$|u^{(t+1)}\rangle = \frac{1}{c_{t+1}} G |u^{(t)}\rangle \tag{14}$$

converge to the dominant eigenstate $|\psi_0\rangle$ of $G$, if the constants $c_t$ are chosen so that $|u^{(t)}\rangle$ assumes a standard form, in which case the constants $c_t$ converge to the dominant eigenvalue. This follows immediately by expanding the initial state $|u^{(0)}\rangle$ in eigenstates of $G$. One possible standard form is that, in some convenient representation, the component of $|u^{(t)}\rangle$ largest in magnitude equals unity.

For quantum mechanical systems, $G$ usually is the imaginary-time evolution operator, $\exp(-\tau \mathcal{H})$. As mentioned above, a technical problem in that case is that an explicit expression is known only asymptotically for short times $\tau$. In practice, this asymptotic expression is used for a small but finite $\tau$ and this leads to systematic, time-step errors. We shall deal with this problem below at length, but ignore it for the time being.

The exponential operator $\exp(-\tau \mathcal{H})$ is one of various alternatives that can be employed to compute the ground-state properties of the Hamiltonian. If the latter is bounded from above, one may be able to use $\mathbb{1} - \tau \mathcal{H}$, where $\tau$ should be small enough that $\lambda_0 \equiv 1 - \tau E_0$ is the dominant eigenvalue of $\mathbb{1} - \tau \mathcal{H}$. In this case, there is no time-step error and the same holds for yet another method of inverting the spectrum of the Hamiltonian, $viz.$ the Green function Monte Carlo method. There one uses $(\mathcal{H} - E)^{-1}$, where $E$ is a constant chosen so that the ground state becomes the dominant eigenstate of this operator. In a Monte Carlo context, matrix elements of the respective operators are proportional to transition probabilities and therefore have to be non-negative, which, if one uses either of the last two methods, may impose further restrictions on the values of $\tau$ and $E$.

For the statistical mechanical applications, the operators $G$ are indeed evolution operators by construction. The transfer matrix evolves the physical system in a spatial rather than time direction, but this spatial direction corresponds to time from the point of view of a Monte Carlo time series. With this in mind, we shall refer to the operator $G$ as the evolution operator, or the *Monte Carlo* evolution operator, if it is necessary to distinguish it from the usual time-evolution operator $\exp(-\tau \mathcal{H})$.

Suppose that $X$ is an operator of which one wants to compute an expectation value. Particularly simple to deal with are the cases in which the operators $X$ and $G$ are the same or commute. We introduce the following notation. Suppose that $|u_\alpha\rangle$ and $|u_\beta\rangle$ are two states, then $X_{\alpha\beta}^{(p',p)}$ denotes the matrix element

$$X_{\alpha\beta}^{(p',p)} = \frac{\langle u_\alpha | G^{p'} X G^p | u_\beta \rangle}{\langle u_\alpha | G^{p'+p} | u_\beta \rangle}. \tag{15}$$

This definition is chosen to simplify the discussion, and generalization to physically relevant expressions, such as Eq. 46, is straightforward.

Various Monte Carlo methods are designed to estimate particular instances of $X_{\alpha\beta}^{(p',p)}$, and often the ultimate goal is to compute the expectation value in the dominant eigenstate

$$X_0 = \frac{\langle\psi_0|X|\psi_0\rangle}{\langle\psi_0|\psi_0\rangle}, \tag{16}$$

which reduces to an expression for the dominant eigenvalue of interest if one chooses for $X$, in the applications discussed in the introduction, the Hamiltonian, transfer or Markov matrix.

The simplest method is the variational Monte Carlo method, discussed in the next section. Here an approximate expectation value is computed by employing an approximate eigenvector of $G$. Typically, this is an optimized trial state, say $|u_\mathrm{T}\rangle$, in which case variational Monte Carlo yields $X_{\mathrm{TT}}^{(0,0)}$, which is simply the expectation value of $X$ in the trial state. Clearly, variational Monte Carlo estimates of $X_0$ have both systematic and statistical errors.

The variational error can be removed asymptotically by projecting out the dominant eigenstate, *i.e.*, by reducing the spectral weight of sub-dominant eigenstates by means of the power method. The simplest case is obtained if one applies the power method only to the right on the state $|u_\beta\rangle$ but not to the left on $\langle u_\alpha|$ in Eq. (15). Mathematically, this is the essence of diffusion and transfer matrix Monte Carlo, and in this way one obtains the desired result $X_0$ if the operator $X$ commutes with the Monte Carlo evolution operator $G$. In our notation, this means that $X_0$ is given by the statistical estimate of $X_{\mathrm{TT}}^{(0,\infty)}$. In principle, this yields an unbiased estimate of $X_0$, but in practice one has to choose $p$ finite but large enough that the estimated systematic error is less than the statistical error. In some practical situations it can in fact be difficult to ascertain that this indeed is the case. If one is interested in the limiting behavior for infinite $p$ or $p'$, the state $|u_\alpha\rangle$ or $|u_\beta\rangle$ need not be available in closed form. This freedom translates into the flexibility in algorithms design exploited in diffusion and transfer matrix Monte Carlo.

If $G$ and $X$ do not commute, the mixed estimator $X_{\mathrm{TT}}^{(0,\infty)}$ is not the desired result, but the residual systematic error can be reduced by combining the variational and mixed estimates by means of the expression

$$X_0 = 2X_{\mathrm{TT}}^{(0,\infty)} - X_{\mathrm{TT}}^{(0,0)} + \mathcal{O}[(|\psi_0\rangle - |u_\mathrm{T}\rangle)^2] \tag{17}$$

To remove the variational bias systematically, if $G$ and $X$ do not commute, the power method must be used to both the left and the right in Eq. (15). Thus one obtains from $X_{\mathrm{TT}}^{(\infty,\infty)}$ an exact estimate of $X_0$ subject only to statistical errors. Of course, one has to pay the price of the implied double limit in terms of loss of statistical accuracy. In the context of the Monte Carlo algorithms discussed below, such as diffusion and transfer matrix Monte Carlo, this double projection technique to estimate $X_{\mathrm{TT}}^{(\infty,\infty)}$ is called *forward walking* or *future walking*.

We end this section on the power method with a brief discussion of the computational complexity of using the Monte Carlo method for eigenvalue problems. In Monte Carlo computations one can distinguish operations of three levels of computational complexity,

depending on whether the operations have to do with single particles or lattice sites, the whole system, or state space summation or integration. The first typically involves a fixed number of elementary arithmetic operations, whereas this number clearly is at least proportional to the system size in the second case. Exhaustive state-space summation grows exponentially in the total system size, and for these problems Monte Carlo is often the only viable option.

Next, the convergence of the power method itself comes into play. The number of iterations required to reach a certain given accuracy is proportional to $\log |\lambda_0/\lambda_1|$, where the $\lambda_0$ and $\lambda_1$ are the eigenvalues of largest and second largest magnitude. If one is dealing with a single-site transfer matrix of a critical system, that spectral gap is proportional to $L^{-d}$ for a system in $d$ dimensions with a cross section of linear dimension $L$. In this case, a single matrix multiplication is of the complexity of a one-particle problem. In contrast, both for the Markov matrix defined above, and the quantum mechanical evolution operator, the matrix multiplication itself is of system-size complexity. Moreover, both of these operators have their own specific problems. The quantum evolution operator of $G(\tau)$ has a gap on the order of $\tau$, which means that $\tau$ should be chosen large for rapid convergence, but one does not obtain the correct results, because of the time-step error, unless $\tau$ is small. Finally, the spectrum of the Markov matrix displays critical slowing down. This means that the gap of the single spin-flip matrix is on the order of $L^{-d-z}$, where $z$ is typically a little bigger than two.[9] These convergence properties are well understood in terms of the mathematics of the power method. Not well understood, however, are problems that are specific to the Monte Carlo implementation of this method, which in some form or another introduces multiplicative fluctuating weights that are correlated with the quantities of interest.[12,13]

## III. SINGLE-THREAD MONTE CARLO

In the previous section we have presented the mathematical expressions that can be evaluated with the Monte Carlo algorithms to be discussed next. The first algorithm is designed to compute an approximate statistical estimate of the matrix element $X_0$ by means of the variational estimate $X_{\mathrm{TT}}^{(0,0)}$. We write* $\langle \mathbf{S}|u_{\mathrm{T}}\rangle = \langle u_{\mathrm{T}}|\mathbf{S}\rangle \equiv u_{\mathrm{T}}(\mathbf{S})$ and for non-vanishing $u_{\mathrm{T}}(\mathbf{S})$ define the *configurational eigenvalue* $X_{\mathrm{T}}(\mathbf{S})$ by

$$X_{\mathrm{T}}(\mathbf{S})u_{\mathrm{T}}(\mathbf{S}) \equiv \langle \mathbf{S}|X|u_{\mathrm{T}}\rangle = \sum_{\mathbf{S}'}\langle \mathbf{S}|X|\mathbf{S}'\rangle\langle \mathbf{S}'|u_{\mathrm{T}}\rangle \tag{18}$$

This yields

$$X_{\mathrm{TT}}^{(0,0)} = \frac{\sum_{\mathbf{S}} u_{\mathrm{T}}(\mathbf{S})^2 X_{\mathrm{T}}(\mathbf{S})}{\sum_{\mathbf{S}} u_{\mathrm{T}}(\mathbf{S})^2}, \tag{19}$$

---

*We assume throughout that the states we are dealing with are represented by real numbers and that $X$ is represented by a real, symmetric matrix. In many cases, generalization to complex numbers is trivial, but for some physical problems, while formal generalization may still be possible, the resulting Monte Carlo algorithms may be too noisy to be practical.

which shows that $X_{\mathrm{TT}}$ can be evaluated as a time average over a Monte Carlo time series of states $\mathbf{S}_1, \mathbf{S}_2, \ldots$ sampled from the probability distribution $u_{\mathrm{T}}(\mathbf{S})^2$, *i.e.*, a process in which Prob($\mathbf{S}$), the probability of finding a state $\mathbf{S}$ at any time is given by

$$\mathrm{Prob}(\mathbf{S}) \propto u_{\mathrm{T}}(\mathbf{S})^2. \tag{20}$$

For such a process, the ensemble average in Eq. (19) can be written in the form of a time average

$$X_{\mathrm{TT}}^{(0,0)} = \lim_{L\to\infty} \frac{1}{L} \sum_{t=1}^{L} X_{\mathrm{T}}(\mathbf{S}_t). \tag{21}$$

For this to be of practical use, it has to be assumed that the configurational eigenvalue $X_{\mathrm{T}}(\mathbf{S})$ can be computed efficiently, which is the case if the sum over states $\mathbf{S}'$ in $\langle \mathbf{S}|X|u_{\mathrm{T}}\rangle = \sum_{\mathbf{S}'}\langle \mathbf{S}|X|\mathbf{S}'\rangle\langle \mathbf{S}'|u_{\mathrm{T}}\rangle$ can be performed explicitly. For discrete states this means that $X$ should be represented by a sparse matrix; if the states $\mathbf{S}$ form a continuum, $X_{\mathrm{T}}(\mathbf{S})$ can be computed directly if $X$ is diagonal or *near-diagonal, i.e.,* involves no or only low-order derivatives in the representation used. The more complicated case of an operator $X$ with arbitrarily nonvanishing off-diagonal elements will be discussed at the end of this section.

An important special case, relevant for example to electronic structure calculations, is to choose for the operator $X$ the Hamiltonian $\mathcal{H}$ and for $\mathbf{S}$ the $3N$-dimensional real-space configuration of the system. Then, the quantity $X_{\mathrm{T}}$ is called the *local energy,* denoted by $E_{\mathrm{L}}$. Clearly, in the ideal case that $|u_{\mathrm{T}}\rangle$ is an exact eigenvector of the evolution operator $G$, and if $X$ commutes with $G$ then the configurational eigenvalue $X_{\mathrm{T}}(\mathbf{S})$ is a constant independent of $\mathbf{S}$ and equals the true eigenvalue of $X$. In this case the variance of the Monte Carlo estimator in Eq. (21) goes to zero, which is an important zero-variance principle satisfied by variational Monte Carlo. The practical implication is that the efficiency of the Monte Carlo computation of the energy can be improved arbitrarily by improving the quality of the trial function. Of course, usually the time required for the computation of $u_{\mathrm{T}}(\mathbf{S})$ increases as the approximation becomes more sophisticated. For the energy the optimal choice minimizes the product of variance and time; no such optimum exists for an operator that does not commute with $G$ or if one makes the fixed node approximation, described in Section VI A 2, since in these cases the results have a systematic error that depends on the quality of the trial wavefunction.

### A. Metropolis Method

A Monte Carlo process sampling the probability distribution $u_{\mathrm{T}}(\mathbf{S})^2$ is usually generated by means of the generalized Metropolis algorithm, as follows. Suppose a configuration $\mathbf{S}$ is given at time $t$ of the Monte Carlo process. A new configuration $\mathbf{S}'$ at time $t+1$ is generated by means of a stochastic process that consists of two steps: (1) an intermediate configuration $\mathbf{S}''$ is proposed with probability $\Pi(\mathbf{S}''|\mathbf{S})$; (2.a) $\mathbf{S}' = \mathbf{S}''$ with probability $p \equiv A(\mathbf{S}''|\mathbf{S})$, *i.e.,* the proposed configuration is accepted; (2.b) $\mathbf{S}' = \mathbf{S}$ with probability $q \equiv 1 - A(\mathbf{S}''|\mathbf{S})$, *i.e.,* the proposed configuration is rejected and the old configuration $\mathbf{S}$ is promoted to time $t+1$. More explicitly, the Monte Carlo sample is generated by means of a Markov matrix $P$ with elements $P(\mathbf{S}'|\mathbf{S})$ of the form

$$P(\mathbf{S}'|\mathbf{S}) = \begin{cases} A(\mathbf{S}'|\mathbf{S})\,\Pi(\mathbf{S}'|\mathbf{S}) & \text{for } \mathbf{S}' \neq \mathbf{S} \\ 1 \,-\, \sum_{\mathbf{S}'' \neq \mathbf{S}} A(\mathbf{S}''|\mathbf{S})\,\Pi(\mathbf{S}''|\mathbf{S}) & \text{for } \mathbf{S}' = \mathbf{S}, \end{cases} \tag{22}$$

if the states are discrete and

$$P(\mathbf{S}'|\mathbf{S}) = A(\mathbf{S}'|\mathbf{S})\,\Pi(\mathbf{S}'|\mathbf{S}) \,+\, \left[1 \,-\, \int d\mathbf{S}'' A(\mathbf{S}''|\mathbf{S})\,\Pi(\mathbf{S}''|\mathbf{S})\right]\,\delta(\mathbf{S}' - \mathbf{S}) \tag{23}$$

if the states are continuous. Correspondingly, in Eq. (22) $\Pi$ and $P$ are probabilities while in Eq. (23) they are probability densities.

The Markov matrix $P$ is designed to satisfy detailed balance

$$P(\mathbf{S}'|\mathbf{S})\,u_{\mathrm{T}}(\mathbf{S})^2 = P(\mathbf{S}|\mathbf{S}')\,u_{\mathrm{T}}(\mathbf{S}')^2, \tag{24}$$

so that, if the process has a unique stationary distribution, this will be $u_{\mathrm{T}}(\mathbf{S})^2$, as desired. In principle, one has great freedom in the choice of the proposal matrix $\Pi$, but it is necessary to satisfy the requirement that transitions can be made between (almost) any pair of states with nonvanishing probability (density) in a finite number of steps.

Once a proposal matrix $\Pi$ is selected, an acceptance matrix is defined so that detailed balance, Eq. (24), is satisfied

$$A(\mathbf{S}'|\mathbf{S}) = \min\left[1,\, \frac{\Pi(\mathbf{S}|\mathbf{S}')\,u_{\mathrm{T}}(\mathbf{S}')^2}{\Pi(\mathbf{S}'|\mathbf{S})\,u_{\mathrm{T}}(\mathbf{S})^2}\right]. \tag{25}$$

For a given choice of $\Pi$, infinitely many choices can be made for $A$ that satisfy detailed balance but the preceding choice is the one with the largest acceptance. We note that if the preceding algorithm is used, then $X_{\mathrm{T}}(\mathbf{S}_t)$ in the sum in Eq. (21), can be replaced by the expectation value conditional on $\mathbf{S}_t''$ having been proposed:

$$X_{\mathrm{TT}}^{(0,0)} = \lim_{L\to\infty} \frac{1}{L} \sum_{t=1}^{L} \left[p_t X_{\mathrm{T}}(\mathbf{S}_t'') + q_t X_{\mathrm{T}}(\mathbf{S}_t)\right], \tag{26}$$

where $p_t = 1 - q_t$ is the probability of accepting $\mathbf{S}_t''$. This has the advantage of reducing the statistical error somewhat, since now $X_{\mathrm{T}}(\mathbf{S}_t'')$ contributes to the average even for rejected moves, and will increase efficiency if $X_{\mathrm{T}}(\mathbf{S}_t'')$ is readily available.

If the proposal matrix $\Pi$ is symmetric, as is the case if one samples from a distribution uniform over a cube centered at $\mathbf{S}$, such as in the original Metropolis method[14], the factors of $\Pi$ in the numerator and denominator of Eq. (25) cancel.

Finally we note that it is not necessary to sample the distribution $u_{\mathrm{T}}^2$ to compute $X_{\mathrm{TT}}$: any distribution that has sufficient overlap with $u_{\mathrm{T}}^2$ will do. To make this point more explicitly, let us introduce the average of some stochastic variable $X$ with respect to an arbitrary distribution $\rho$:

$$\langle X \rangle_\rho \equiv \frac{\sum_{\mathbf{S}} X(\mathbf{S})\rho(\mathbf{S})}{\sum_{\mathbf{S}} \rho(\mathbf{S})} \tag{27}$$

The following relation shows that the desired results can be obtained by reweighting, *i.e.*, any distribution $\rho$ will suffice as long as the ratio $u_{\mathrm{T}}(\mathbf{S})^2/\rho(\mathbf{S})$ does not fluctuate too wildly:

$$X_{\mathrm{TT}}^{(0,0)} = \langle X \rangle_{u_{\mathrm{T}}^2} = \frac{\langle X u_{\mathrm{T}}^2/\rho \rangle_\rho}{\langle u_{\mathrm{T}}^2/\rho \rangle_\rho}. \tag{28}$$

This is particularly useful for calculation of the difference of expectation values with respect to two closely related distributions. An example of this[15,16] is the calculation of the energy of a molecule as a function of the inter-nuclear distance.

### B. Projector Monte Carlo and Importance Sampling

The generalized Metropolis method is a very powerful way to sample an arbitrary *given* distribution, and it allows one to construct infinitely many Markov processes with the desired distribution as the stationary state. None of these, however, may be appropriate to design a Monte Carlo version of the power method to solve eigenvalue problems. In this case, the evolution operator $G$ is *given,* possibly in approximate form, and its dominant eigenstate may *not* be known. To construct an appropriate Monte Carlo process, the first problem is that $G$ itself is not a Markov matrix, *i.e.*, it may violate one or both of the properties $G(\mathbf{S}'|\mathbf{S}) \geq 0$ and $\sum_{\mathbf{S}'} G(\mathbf{S}'|\mathbf{S}) = 1$. This problem can be solved if we can find a factorization of the evolution matrix $G$ into a Markov matrix $P$ and a weight matrix $g$ with non-negative elements such that

$$G(\mathbf{S}'|\mathbf{S}) = g(\mathbf{S}'|\mathbf{S})P(\mathbf{S}'|\mathbf{S}). \tag{29}$$

The weights $g$ must be finite, and this almost always precludes use of the Metropolis method for continuous systems, as can be understood as follows. Since there is a finite probability that a proposed state will be rejected, the Markov matrix $P(\mathbf{S}'|\mathbf{S})$ will contain terms involving $\delta(\mathbf{S} - \mathbf{S}')$, but generically, $G$ will not have the same structure and will not allow the definition of finite weights $g$ according to Eq. (29). However, the Metropolis algorithm can be incorporated as a component of an algorithm if an approximate stationary state is known and if further approximations are made, as in the diffusion Monte Carlo algorithm discussed in Section VI B.

As a comment on the side we note that violation of the condition that the weight $g$ be positive results in the notorious *sign problem* in one of its guises, which is in most cases unavoidable in the treatment of fermionic or frustrated systems. Many ingenious attempts[17] have been made to solve this problem, but this is still a topic of active research. However, as mentioned, we restrict ourselves in this chapter to the case of evolution operators $G$ with nonnegative matrix elements only.

We resume our discussion of the factorization given in Eq. (29). Suppose for the sake of argument that the left eigenstate $\hat{\psi}_0$ of $G$ is known and that its elements are positive,

$$\sum_{\mathbf{S}'} \hat{\psi}_0(\mathbf{S}')G(\mathbf{S}'|\mathbf{S}) = \lambda_0 \hat{\psi}_0(\mathbf{S}). \tag{30}$$

If in addition, the matrix elements of $G$ are nonnegative, the following matrix $\hat{P}$ is a Markov matrix

$$\hat{P}(\mathbf{S}'|\mathbf{S}) = \frac{1}{\lambda_0}\hat{\psi}_0(\mathbf{S}')G(\mathbf{S}'|\mathbf{S})\frac{1}{\hat{\psi}_0(\mathbf{S})}. \tag{31}$$

Unless one is dealing with a Markov matrix from the outset, the left eigenvector of $G$ is seldom known, but it is convenient, in any event, to perform a so-called *importance sampling* transformation on $G$. For this purpose we introduce a guiding function $u_{\mathrm{g}}$ and define

$$\hat{G}(\mathbf{S}'|\mathbf{S}) = u_{\mathrm{g}}(\mathbf{S}')G(\mathbf{S}'|\mathbf{S})\frac{1}{u_{\mathrm{g}}(\mathbf{S})}. \tag{32}$$

We shall return to the issue of the guiding function, but for the time being the reader can think of it either as an arbitrary, positive function, or as an approximation to the dominant eigenstate of $G$. From a mathematical point of view, anything that can be computed with the original Monte Carlo evolution operator $G$ can also be computed with $\hat{G}$, since the two represent the same abstract operator in a different basis. The representations differ only by normalization constants. All we have to do is to write all expressions derived above in terms of this new basis.

We continue our discussion in terms of the transform $\hat{G}$ and replace Eq. (29) by the factorization

$$\hat{G}(\mathbf{S}'|\mathbf{S}) = \hat{g}(\mathbf{S}'|\mathbf{S})\hat{P}(\mathbf{S}'|\mathbf{S}) \tag{33}$$

and we assume that $\hat{P}$ has the explicitly known distribution $u_{\mathrm{g}}^2$ as its stationary state. The guiding function $u_{\mathrm{g}}$ appears in those expressions, and it should be kept in mind that they can be reformulated by means of the reweighting procedure given in Eq. (28) to apply to processes with different explicitly known stationary states. On the other hand, one might be interested in the infinite projection limit $p \to \infty$. In that case, one might use a Monte Carlo process for which the stationary distribution is not known explicitly. Then, the expressions below should be rewritten so that the unknown distribution does not appear in expressions for the time averages. The function $u_{\mathrm{g}}$ will still appear, but only as a transformation known in closed form and no longer as the stationary state of $P$. Clearly, a process for which the distribution is not known in closed form cannot be used to compute the matrix elements $X_{\alpha\beta}^{(p',p)}$ for finite $p$ and $p'$ and given states $|u_\alpha\rangle$ and $|u_\beta\rangle$.

One possible choice for $\hat{P}$ that avoids the Metropolis method and produces finite weights is the following *generalized heat bath* transition matrix

$$\hat{P}(\mathbf{S}'|\mathbf{S}) = \frac{\hat{G}(\mathbf{S}'|\mathbf{S})}{\sum_{\mathbf{S}_1} \hat{G}(\mathbf{S}_1|\mathbf{S})}. \tag{34}$$

If $G(\mathbf{S}'|\mathbf{S})$ is symmetric, this transition matrix has a known stationary distribution, *viz.*, $G_{\mathrm{g}}(\mathbf{S})u_{\mathrm{g}}^2(\mathbf{S})$, where $G_{\mathrm{g}}(\mathbf{S}) = \langle\mathbf{S}|G|u_{\mathrm{g}}\rangle/u_{\mathrm{g}}(\mathbf{S})$, the configurational eigenvalue of $G$ in state $\mathbf{S}$. $\hat{P}$ must be chosen such that the corresponding transitions can be sampled directly. This is usually not feasible unless $\hat{P}$ is sparse or near-diagonal, or can be transformed into a form involving non-interacting degrees of freedom. We note that if $\hat{P}$ is defined by Eq. (34), the weight matrix $\hat{g}$ depends only on $\mathbf{S}$.


### C. Matrix Elements

We now address the issue of computing the matrix elements $X_{\alpha\beta}^{(p',p)}$, assuming that the stationary state $u_{\mathrm{g}}^2$ is known explicitly and that the weight matrix $\hat{g}$ has finite elements.

We shall discuss the following increasingly complex possibilities: (a) $[X, G] = 0$ and $X$ is near-diagonal in the $\mathbf{S}$ representation; (b) $X$ is diagonal in the $\mathbf{S}$ representation; (c) $X(\mathbf{S}|\mathbf{S}')$ is truly off-diagonal. The fourth case, *viz.*, $[X, G] = 0$ and $X$ is not near-diagonal is omitted since it can easily be constructed from the three cases discussed explicitly. When discussing case (c) we shall introduce the concept of *side walks* and explain how these can be used to compute matrix elements of a more general nature than discussed up to that point. After deriving the expressions, we shall discuss the practical problems they give rise to, and ways to reduce the variance of the statistical estimators. Since this yields expressions in a more symmetric form, we introduce the transform

$$\hat{X}(\mathbf{S}'|\mathbf{S}) = u_{\mathrm{g}}(\mathbf{S}') X(\mathbf{S}'|\mathbf{S}) \frac{1}{u_{\mathrm{g}}(\mathbf{S})}. \tag{35}$$

### 1. $[X, G] = 0$ and $X$ Near-Diagonal

In this case, $X_{\alpha\beta}^{(0,p'+p)} = X_{\alpha\beta}^{(p',p)}$ and it suffices to consider the computation of $X_{\alpha\beta}^{(0,p)}$. By repeated insertion in Eq. (15) of the resolution of the identity in the $\mathbf{S}$-basis, one obtains the expression

$$X_{\alpha\beta}^{(0,p)} = \frac{\sum_{\mathbf{S}_p,\ldots,\mathbf{S}_0} u_{\alpha}(\mathbf{S}_p) \, X_{\alpha}(\mathbf{S}_p) \, [\prod_{i=0}^{p-1} G(\mathbf{S}_{i+1}|\mathbf{S}_i)] \, u_{\beta}(\mathbf{S}_0)}{\sum_{\mathbf{S}_p,\ldots,\mathbf{S}_0} u_{\alpha}(\mathbf{S}_p) \, [\prod_{i=0}^{p-1} G(\mathbf{S}_{i+1}|\mathbf{S}_i)] \, u_{\beta}(\mathbf{S}_0)} \tag{36}$$

In the steady state, a series of subsequent states $\mathbf{S}_t, \mathbf{S}_{t+1}, \ldots, \mathbf{S}_{t+p}$ occurs with probability

$$\mathrm{Prob}(\mathbf{S}_t, \mathbf{S}_{t+1}, \ldots, \mathbf{S}_{t+p}) \propto [\prod_{i=0}^{p-1} \hat{P}(\mathbf{S}_{t+i+1}|\mathbf{S}_{t+i})] \, u_{\mathrm{g}}(\mathbf{S}_t)^2. \tag{37}$$

To relate products of the matrix $P$ to those of $G$, it is convenient to introduce the following definitions

$$\hat{W}_t(p, q) = \prod_{i=q}^{p-1} \hat{g}(\mathbf{S}_{t+i+1}|\mathbf{S}_{t+i}) \tag{38}$$

Also, we define

$$\hat{u}_{\omega}(\mathbf{S}) = \frac{u_{\omega}(\mathbf{S})}{u_{\mathrm{g}}(\mathbf{S})}, \tag{39}$$

where $\omega$ can be any of a number of subscripts.

With these definitions, combining Eqs. (29), (36), and (37), one finds

$$X_{\alpha\beta}^{(0,p)} = \lim_{L\to\infty} \frac{\sum_{t=1}^{L} \hat{u}_{\alpha}(\mathbf{S}_{t+p}) \, X_{\alpha}(\mathbf{S}_{t+p}) \, \hat{W}_t(p, 0) \hat{u}_{\beta}(\mathbf{S}_t)}{\sum_{t=1}^{L} \hat{u}_{\alpha}(\mathbf{S}_{t+p}) \, \hat{W}_t(p, 0) \hat{u}_{\beta}(\mathbf{S}_t)}. \tag{40}$$

14

## 2. Diagonal X

The preceding discussion can be generalized straightforwardly to the case in which $X$ is diagonal in the $\mathbf{S}$ representation. Again by repeated insertion of the resolution of the identity in the $\mathbf{S}$-basis in the Eq. (15) for $X_{\alpha\beta}^{(p',p)}$, one obtains the identity

$$X_{\alpha\beta}^{(p',p)} = \frac{\sum_{\mathbf{S}_{p'+p},\ldots,\mathbf{S}_0} u_\alpha(\mathbf{S}_{p'+p}) \left[\prod_{i=p}^{p'+p-1} G(\mathbf{S}_{i+1}|\mathbf{S}_i)\right] X(\mathbf{S}_p|\mathbf{S}_p) \left[\prod_{i=0}^{p-1} G(\mathbf{S}_{i+1}|\mathbf{S}_i)\right] u_\beta(\mathbf{S}_0)}{\sum_{\mathbf{S}_{p'+p},\ldots,\mathbf{S}_0} u_\alpha(\mathbf{S}_{p'+p}) \left[\prod_{i=0}^{p'+p-1} G(\mathbf{S}_{i+1}|\mathbf{S}_i)\right] u_\beta(\mathbf{S}_0)}.$$

(41)

Again by virtue of Eq. (37), we find

$$X_{\alpha\beta}^{(p',p)} = \lim_{L\to\infty} \frac{\sum_{t=1}^{L} \hat{u}_\alpha(\mathbf{S}_{t+p'+p}) \hat{W}_t(p'+p,p) X(\mathbf{S}_{t+p}|\mathbf{S}_{t+p}) \hat{W}_t(p,0)\hat{u}_\beta(\mathbf{S}_t)}{\sum_{t=1}^{L} \hat{u}_\alpha(\mathbf{S}_{t+p'+p}) \hat{W}_t(p'+p,0)\hat{u}_\beta(\mathbf{S}_t)}.$$

(42)

## 3. Nondiagonal X

If the matrix elements of $G$ vanish only if those of $X$ do, the preceding method can be generalized immediately to the final case in which $X$ is nondiagonal. Then, the analog of Eq. (42) is

$$X_{\alpha\beta}^{(p',p)} = \lim_{L\to\infty} \frac{\sum_{t=1}^{L} \hat{u}_\alpha(\mathbf{S}_{t+p'+p}) \hat{W}_t(p'+p,p+1) x(\mathbf{S}_{t+p+1}|\mathbf{S}_{t+p}) \hat{W}_t(p,0)\hat{u}_\beta(\mathbf{S}_t)}{\sum_{t=1}^{L} \hat{u}_\alpha(\mathbf{S}_{t+p'+p}) \hat{W}_t(p'+p,0)\hat{u}_\beta(\mathbf{S}_t)}.$$

(43)

where the $x$ matrix elements are defined by

$$x(\mathbf{S}'|\mathbf{S}) = \frac{X(\mathbf{S}'|\mathbf{S})}{P(\mathbf{S}'|\mathbf{S})} = \frac{\hat{X}(\mathbf{S}'|\mathbf{S})}{\hat{P}(\mathbf{S}'|\mathbf{S})}.$$

(44)

Clearly, the preceding definition of $x(\mathbf{S}'|\mathbf{S})$ fails when $\hat{P}(\mathbf{S}'|\mathbf{S})$ vanishes but $\hat{X}(\mathbf{S}'|\mathbf{S})$ does not. If that can happen, a more complicated scheme can be employed in which one introduces *side-walks*. This is done by interrupting the continuing stochastic process at time $t+p$ by introducing a finite series of auxiliary states $\mathbf{S}'_{t+p+1},\ldots,\mathbf{S}'_{t+p'+p}$. The latter are generated by a separate stochastic process so that in equilibrium, the sequence of subsequent states $\mathbf{S}_t,\mathbf{S}_{t+1},\ldots,\mathbf{S}_{t+p},\mathbf{S}'_{t+p+1},\ldots,\mathbf{S}'_{t+p'+p}$ occurs with probability

$$\mathrm{Prob}[(\mathbf{S}_t,\mathbf{S}_{t+1},\ldots,\mathbf{S}_{t+p},\mathbf{S}'_{t+p+1},\ldots,\mathbf{S}'_{t+p'+p})] \propto$$
$$\left[\prod_{i=p+1}^{p'+p-1} \hat{P}(\mathbf{S}'_{t+i+1}|\mathbf{S}'_{t+i})\right] \hat{P}_X(\mathbf{S}'_{t+p+1}|\mathbf{S}_{t+p}) \left[\prod_{i=0}^{p-1} \hat{P}(\mathbf{S}_{t+i+1}|\mathbf{S}_{t+i})\right] u_{\mathrm{g}}(\mathbf{S}_t)^2$$

(45)

where $\hat{P}_X$ is a Markov matrix chosen to replace $\hat{P}$ in Eq. (44) so as to yield finite weights $x$. In this scheme, one generates a continuing thread identical to the usual Monte Carlo process in which each state $\mathbf{S}_t$ is sampled from the stationary state of $\hat{P}$, at least if one ignores the initial equilibration. Each state $\mathbf{S}_t$ of this backbone forms the beginning of a side walk, the first step of which is sampled from $\hat{P}_X$, while $\hat{P}$ again generates subsequent ones. Clearly,

15

with respect to the side walk, the first step disrupts the stationary state, so that the $p'$ states $\mathbf{S}'_{t'}$, which form the side walk, do not sample the stationary state of the original stochastic process generated by $\hat{P}$, unless $\hat{P}_X$ coincidentally has the same stationary state as $\hat{P}$.

A problem with the matrix elements we dealt with up to now is that in the limit $p'$ or $p \to \infty$ all of them reduce to matrix elements involving the dominant eigenstate, although symmetries might be used to yield other eigenstates besides the absolute dominant one. However, if symmetries fail, one has to employ the equivalent of an orthogonalization scheme, such as, discussed in the next section, or one is forced to resort to evolution operators that contain, in exact or in approximate form, the corresponding projections. An example of this are matrix elements computed in the context of the fixed-node approximation[18], discussed in Section VI A 2. Within the framework of this approximation, one considers quantities of the form

$$X_{\alpha\beta}^{(p',p)} = \frac{\langle u_\alpha | G_1^{p'} X G_2^p | u_\beta \rangle}{\sqrt{\langle u_\alpha | G_1^{2p'} | u_\alpha \rangle \langle u_\beta | G_2^{2p} | u_\beta \rangle}}, \tag{46}$$

where the $G_i$ are evolution operators combined with appropriate projectors, which in the fixed-node approximation are defined by the nodes of the states $u_\alpha(\mathbf{S})$ and $u_\beta(\mathbf{S})$. We shall describe how the preceding expression, Eq. (46), can be evaluated, but rather than writing out all the expressions explicitly, we present just the essence of the Monte Carlo method.

To deal with these expressions, one generates a backbone time series of states sampled from any distribution, say, $u_g(\mathbf{S})^2$, that has considerable overlap with the the states $|u_\alpha(\mathbf{S})|$ and $|u_\beta(\mathbf{S})|$. Let us distinguish those backbone states by a superscript 0. Consider any such state $S^{(0)}$ at some given time. It forms the starting point of two side walks. We denote the states of these side walks by $\mathbf{S}_{t_i}^{(i)}$ where $i = 1, 2$ identifies the side walk and $t_i$ labels the side steps. The side walks are generated from factorizations of the usual form, defined in Eq. (33), say $\hat{G}_i = \hat{g}_i \hat{P}_i$. A walk

$$\mathcal{S} = [\mathbf{S}^{(0)}, (\mathbf{S}_1^{(1)}, \mathbf{S}_2^{(1)}, \ldots), (\mathbf{S}_1^{(2)}, \mathbf{S}_2^{(2)}, \ldots)] \tag{47}$$

occurs with probability

$$\text{Prob}(\mathcal{S}) = u_g(\mathbf{S}(0))^2 \, \hat{P}_1(\mathbf{S}^{(0)}|\mathbf{S}_1^{(1)}) \ldots \hat{P}_2(\mathbf{S}^{(0)}|\mathbf{S}_1^{(2)}) \ldots . \tag{48}$$

We leave it to the reader to show that this probability suffices for the computation of all expressions appearing in numerator and denominator of Eq. (46), in the case that $X$ is diagonal, and to generate the appropriate generalizations to other cases.

In the expressions derived above, the power method projections precipitate products of reweighting factors $\hat{g}$, and, as the projection times $p$ and $p'$ increase, the variance of the Monte Carlo estimators grows at least exponentially in the square root of the projection time. Clearly, the presence of the fluctuating weights $\hat{g}$ is due to the fact that the evolution operator $\hat{G}$ is not Markovian in the sense that it fails to conserve probability. The importance sampling transformation Eq. (32) was introduced to mitigate this problem. In Section V, an algorithm involving branching walks will be introduced, which is a different strategy devised to deal with this problem. In diffusion and transfer matrix Monte Carlo, both strategies, importance sampling and branching, are usually employed simultaneously.

## D. Excited States

Given a set of basis states, excited eigenstates can be computed variationally by solving a linear variational problem and the Metropolis method can be used to evaluate the required matrix elements. The methods involving the power method, as described above, can then be used to remove the variational bias systematically.[13,19,20]

In this context matrix elements appear in the solution of the following variational problem. As was mentioned several times before, the price paid for reducing the variational bias is increased statistical noise, a problem which appears in this context with a vengeance. Again, the way to keep this problem under control is the use of optimized trial vectors.

The variational problem to be solved is the following one. Given $n$ basis functions $|u_i\rangle$, find the $n \times n$ matrix of coefficients $d_i^{(j)}$ such that

$$|\tilde{\psi}_j\rangle = \sum_{i=1}^{n} d_i^{(j)}|u_i\rangle \tag{49}$$

are the best variational approximations for the $n$ lowest eigenstates $|\psi_i\rangle$ of some Hamiltonian $\mathcal{H}$. In this problem we shall use the language of the quantum mechanical systems, where one has to distinguish the Hamiltonian from the evolution operator $\exp(-\tau\mathcal{H})$. In the statistical mechanical applications, one has only the equivalent of the latter. In the expressions to be derived below the substitution $\mathcal{H}G^p \to G^{p+1}$ will produce the expressions required for the statistical mechanical applications, at least if we assume that the nonsymmetric matrices that appear in that context have been symmetrized.[†]

One seeks a solution to the linear variational problem in Eq. (49) in the sense that for all $i$ the Rayleigh quotient $\langle\tilde{\psi}_i|\mathcal{H}|\tilde{\psi}_i\rangle/\langle\tilde{\psi}_i|\tilde{\psi}_i\rangle$ is stationary with respect to variation of the coefficients $d$. The solution is that the matrix of coefficients $d$ has to satisfy the following generalized eigenvalue equation

$$\sum_{i=1}^{n} H_{ki}d_i^{(j)} = \tilde{E}_j \sum_{i=1}^{n} N_{ki}d_i^{(j)}, \tag{50}$$

where

$$H_{ki} = \langle u_k|\mathcal{H}|u_i\rangle, \tag{51}$$

and

$$N_{ki} = \langle u_k|u_i\rangle. \tag{52}$$

Before discussing Monte Carlo issues, we note a number of important properties of this scheme. First of all, the basis states $|u_i\rangle$ in general are not orthonormal and this is reflected by the fact that the matrix elements of $N$ have to be computed. Secondly, it is clear that any

---

[†]This is not possible in general for transfer matrices of systems with helical boundary conditions, but the connection between left and right eigenvectors of the transfer matrix (see Section I B) can be used to generalize the approach discussed here.

nonsingular linear combination of the basis vectors will produce precisely the same results, obtained from the correspondingly transformed version of Eq. (50). The final comment is that the variational eigenvalues bound the exact eigenvalues, *i.e.*, $\tilde{E}_i \geq E_i$. One recovers exact eigenvalues $E_i$ and the corresponding eigenstates, if the $|u_i\rangle$ span the same space as the exact eigenstates.

The required matrix elements can be computed using the variational Monte Carlo method discussed in the previous section. Furthermore, the power method can be used to reduce the variational bias. Formally, one simply defines new basis states

$$|u_i^{(p)}\rangle = G^p|u_i\rangle \tag{53}$$

and substitutes these new basis states for the original ones. The corresponding matrices

$$H_{ki}^{(p)} = \langle u_k^{(p)}|\mathcal{H}|u_i^{(p)}\rangle \tag{54}$$

and

$$N_{ki}^{(p)} = \langle u_k^{(p)}|u_i^{(p)}\rangle \tag{55}$$

can again be computed by applying the methods introduced in Section III for the computation of general matrix elements by a Monte Carlo implementation of the power method.

As an explicit example illustrating the nature of the Monte Carlo time-averages that one has to evaluate in this approach, we write down the expression for $N_{ij}^{(p)}$ as used for the computation of eigenvalues of the Markov matrix relevant to the problem of critical slowing down:

$$N_{ij}^{(p)} \approx \sum_t \frac{u_i(\mathbf{S}_t)}{\psi_B(\mathbf{S}_t)} \frac{u_j(\mathbf{S}_{t+p})}{\psi_B(\mathbf{S}_{t+p})}, \tag{56}$$

where the $\mathbf{S}_t$ are configurations forming a time series which, as we recall, is designed to sample the distribution of a system in thermodynamic equilibrium, *i.e.*, the Boltzmann distribution $\psi_B^2$. The expression given in Eq. (56) yields the $u/\psi_B$-auto-correlation function at lag $p$. The expression for $H_{ij}^{(p)}$ is similar, and represents a cross-correlation function involving the configurational eigenvalues of the Markov matrix in the various basis states. Compared to the expressions derived in Section III, Eq. (56) takes a particularly simple form in which products of fluctuating weights are absent, because in this particular problem one is dealing with a probability conserving evolution operator from the outset.

Eq. (56) shows why this method is sometimes called correlation function Monte Carlo, but it also illustrates a new feature, namely, that it is efficient to compute all required matrix elements simultaneously. This can be done by generating a Monte Carlo process with a known distribution which has sufficient overlap with all $|u_i(\mathbf{S})| \equiv |\langle \mathbf{S}|u_i\rangle|$. This can be arranged, for example, by sampling a guiding function $u_g(\mathbf{S})$ defined by

$$u_g(\mathbf{S}) = \sqrt{\sum_{i=1}^n a_i u_i(\mathbf{S})^2}, \tag{57}$$

where the coefficients $a_i$ approximately normalize the basis states $|u_i\rangle$, which may require a preliminary Monte Carlo run. See Ceperley and Bernu[13] for an alternative choice for a

guiding function. In the computations[10] to obtain the spectrum of the Markov matrix in critical dynamics, as illustrated by Eq. (56), the Boltzmann distribution, is used as a guiding function. It apparently has enough overlap with the decaying modes that no special purpose distribution has to be generated.

## E. How to Avoid Reweighting

Before discussing the branching algorithms designed to deal more efficiently with the reweighting factors appearing in the expressions discused above, we briefly mention an alternative that has surfaced occasionally without being studied extensively, to our knowledge. The idea will be illustrated in the case of the computation of the matrix element $X_{\alpha\beta}^{(0,p)}$, and we take Eqn. (36) as our starting point. In statistical mechanical language, we introduce a reduced Hamiltonian

$$\mathcal{H} = \ln u_{\mathrm{g}}(\mathbf{S}_p) + \sum_{i=0}^{p-1} \ln G(\mathbf{S}_{i+1}|\mathbf{S}_i) + \ln u_{\mathrm{g}}(\mathbf{S}_0) \tag{58}$$

and the corresponding Boltzmann distribution $\exp -\mathcal{H}(\mathbf{S}_p, \ldots, \mathbf{S}_0)$. One can now use the standard Metropolis algorithm to sample this distribution for this system consisting of $p+1$ layers bounded by the layers 0 and $p$. For the evaluation of Eq. (36) by Monte Carlo, this expression then straightforwardly becomes a ratio of correlation functions involving quantities defined at the boundaries. To see this, all one has to do is to divide the numerator and denominator of Eq. (36) by the partition function

$$Z = \sum_{\mathbf{S}_p, \ldots, \mathbf{S}_0} e^{-\mathcal{H}(\mathbf{S}_p, \ldots, \mathbf{S}_0)} \tag{59}$$

Note that in general, boundary terms involving some appropriately defined $u_{\mathrm{g}}$ should be introduced to ensure the non-negativity of the distribution. For the simultaneous computation of matrix elements for several values of the indices $\alpha$ and $\beta$, a guiding function $u_{\mathrm{g}}$ should be chosen that has considerable overlap with the corresponding $|u_\alpha|$ and $|u_\beta|$.

The Metropolis algorithm can of course be used to sample any probability distribution, and the introduction of the previous Hamiltonian illustrates just one particular point of view. If one applies the preceding idea to the case of the imaginary-time quantum mechanical evolution operator, one obtains a modified version of the standard path-integral Monte Carlo method, in which case the layers are usually called time slices. Clearly, this method has the advantage of suppressing the fluctuating weights in estimators. However, the disadvantage is that sampling the full, layered system yields a longer correlation time than sampling the single-layer distribution $u_{\mathrm{g}}^2$. This is a consequence of the fact that the microscopic degrees of freedom are more strongly correlated in a layered system than in a single layer. Our limited experience suggests that for small systems reweighting is more efficient, whereas the Metropolis approach tends to become more efficient as the system grows in size.[61]

## IV. TRIAL FUNCTION OPTIMIZATION

In the previous section it was shown that eigenvalue estimates can be obtained as the eigenvalues of the matrix $N^{(p)^{-1}} H^{(p)}$. The variational error in these estimates decreases

as $p$ increases. In general, these expressions involve weight products of increasing length, and consequently the errors grow exponentially, but even in the simple case of a probability conserving evolution operator, errors grow exponentially. This is a consequence of the fact that the auto-correlation functions in $N^{(p)}$, and the cross-correlation functions in $H^{(p)}$, in the limit $p \to \infty$ reduce to quantities that contain an exponentially vanishing amount of information about the subdominant or excited-state eigenvalues, since the spectral weight of all but the dominant eigenstate is reduced to zero by the power method.

The practical implication is that this information has to be retrieved with sufficient accuracy for small values of $p$, before the signal disappears in the statistical noise. The projection time $p$ can be kept small by using optimized basis states constructed to reduce the overlap of the linear space spanned by the basis states $|u_i\rangle$ with the space spanned by the eigenstates beyond the first $n$ of interest. We shall describe, mostly qualitatively, how this can be done by a generalization of a method used for optimization of individual basis states[3,21−23], *viz.* minimization of variance of the configurational eigenvalue, the local energy in quantum Hamiltonian problems.

Suppose that $u_{\mathrm{T}}(\mathbf{S}, v)$ is the value of the trial function $u_{\mathrm{T}}$ for configuration $\mathbf{S}$ and some choice of the parameters $v$ to be optimized. As in Eq. (18), the *configurational eigenvalue* $\lambda(S, v)$ of configuration $\mathbf{S}$ is defined by

$$u'_{\mathrm{T}}(\mathbf{S}, v) \equiv \lambda(\mathbf{S}, v) u_{\mathrm{T}}(\mathbf{S}, v), \tag{60}$$

where a prime is used to denote, for arbitrary $|f\rangle$, the components of $G|f\rangle$, or $\mathcal{H}|f\rangle$ as is more convenient for quantum mechanical applications. The optimal values of the variational parameters are obtained by minimization of the variance of $\lambda(\mathbf{S}, v)$, estimated as an average over a small Monte Carlo sample. In the ideal case, *i.e.,* if an exact eigenstate can be reproduced by some choice of the parameters of $u_{\mathrm{T}}$, the minimum of the variance yields the exact eigenstate not only if it were to be computed exactly, but even if it is approximated by summation over a Monte Carlo sample. A similar zero-variance principle holds for the method of simultaneous optimization of several trial states to be discussed next. This is in sharp contrast with the more traditional Rayleigh-Ritz extremization of the Rayleigh quotient, which frequently can produce arbitrarily poor results if minimized over a small sample of configurations.

For conceptual simplicity, we first generalize the preceding method to the more general ideal case that reproduces the exact values of the desired $n$ eigenstates of the evolution operator $G$. As a byproduct, our discussion will produce an alternative to the derivation of Eq. (50). To compute $n$ eigenvalues, we have to optimize the $n$ basis states $|u_i\rangle$, where we have dropped the index "T", and again we assume we have a sample of $M$ configurations $\mathbf{S}_\alpha$, $\alpha = 1, \ldots, M$ sampled from $u_{\mathrm{g}}^2$. The case we consider is ideal in the sense that we assume that these basis states $|u_i\rangle$ span an $n$-dimensional invariant subspace of $G$. In that case, by definition there exists a matrix $\Lambda$ of order $n$ such that

$$u_i{}'(\mathbf{S}_\alpha) = \sum_{j=1}^{n} \Lambda_{ij} u_j(\mathbf{S}_\alpha). \tag{61}$$

Again, the prime on the left-hand side of this equation indicates multiplication by $G$ or by $\mathcal{H} = -\tau^{-1} \ln G$. If the number of configurations is large enough, $\Lambda$ is for all practical purposes determined uniquely by the set of equations (61) and one finds

$$\Lambda = N^{-1}H, \tag{62}$$

where

$$
\begin{aligned}
N_{ij} &= \tfrac{1}{M} \sum_{\alpha=1}^{M} u_i(S_\alpha) u_j(S_\alpha)/u_{\mathrm{g}}(S_\alpha)^2, \\
H_{ij} &= \tfrac{1}{M} \sum_{\alpha=1}^{M} u_i(S_\alpha) u_j'(S_\alpha)/u_{\mathrm{g}}(S_\alpha)^2.
\end{aligned} \tag{63}
$$

In the limit $M \to \infty$ this indeed reproduces the matrices $N$ and $H$ in Eq. (50). In the nonideal case, the space spanned by the $n$ basis states $|u_i\rangle$ is not an invariant subspace of the matrix $G$. In that case, even though Eq. (61) generically has no true solution, Eqs. (62) and (63) still constitute a solution in the least-squares sense, as the reader is invited to show for himself by solving the normal equations.

Next, let us consider the construction of a generalized optimization criterion. As mentioned before, if a set of basis states span an invariant subspace, so does any nonsingular linear combination. In principle, the optimization criterion should have the same invariance. The *matrix* $\Lambda$ lacks this property, but its *spectrum* is invariant. Another consideration is that, while the local eigenvalue is defined by a single configuration $\mathbf{S}$, it takes at least $n$ configurations to determine the "local" matrix $\Lambda$. This suggests that one subdivide the sample into subsamples of at least $n$ configurations each and minimize the variance of the *local spectrum* over these *subsamples*. Again in principle, this has the advantage that the optimization can exploit the fact that linear combinations of the basis states have more variational freedom to represent the eigenstates than does each variational basis function separately. In practice, however, this advantage seems to be negated by the difficulty of finding good optimal parameters. This is a consequence of the fact that invariance under linear transformation usually can be mimicked by variation of the parameters of the basis states. In other words, a linear combination of basis states can be represented accurately, at least relative to the noise in the local spectrum, by a single basis state with appropriately chosen parameters. Consequently, intrinsic flaws of the trial states exceed what can be gained in reducing the variance of the local spectrum by exploiting the linear variational freedom, most of which is already used anyway in the linear variational problem that was discussed at the beginning of this section. This means that one has to contend with a near-singular nonlinear optimization problem. In practice, to avoid the concomitant slow convergence, it seems to be more efficient to break the "gauge symmetry" and select a preferred basis, which most naturally is done by requiring that each basis state *itself* is a good approximate eigenstate.

The preceding considerations, of course, leave us with two criteria, *viz.* minimization of the variance of the local spectrum as a whole, and minimization of the variance of the configurational eigenvalue separately. To be of practical use, both criteria have to be combined, since if one were to proceed just by minimization of the variance of the configurational eigenvalues separately, one would simply keep reproducing the same eigenstate. In a non-Monte Carlo context this can be solved simply by some orthogonalization scheme, but as far as Monte Carlo is concerned, that is undesirable since it fails to yield a zero-variance optimization principle.

# V. BRANCHING MONTE CARLO

In Section III we discussed a method to compute Monte Carlo averages by exploiting the power method to reduce the spectral weight of undesirable, subdominant eigenstates. We saw that this leads to products of weights of subsequent configurations sampled by a Monte Carlo time series. To suppress completely the systematic errors due to finite projection times, *i.e.,* the variational bias, one has to take averages of infinite products of weights. This limit would produce an "exact" method with infinite variance, which is of no practical use.

We have also discussed how optimized trial states can be used to reduce the variance of this method. The variance reduction may come about in two ways. In the first place, by starting with optimized trial states of higher quality, the variational bias is smaller to begin with so that fewer power method projections are required. In practical terms, this leads to a reduction of the number of factors in the fluctuating products. Secondly, a good estimate of the dominant eigenstate, can be used to reduce the amount by which the evolution operator, divided by an appropriate constant, violates conservation of probability, which reduces the variance of the individual fluctuating weight factors. All these considerations also apply to the branching Monte Carlo algorithm discussed in this section, which can be modified accordingly and in complete analogy with our previous discussion.

Before discussing the details of the branching algorithm, we mention that the algorithm presented here[8] contains the mathematical essence of both the diffusion and transfer matrix Monte Carlo algorithms. A related algorithm, *viz.*, Green function Monte Carlo, adds yet another level of complexity due to the fact that the evolution operator is known only as an infinite series. This series is stochastically summed at each step of the power method iterations. In practice this implies that even the time step becomes stochastic and intermediate Monte Carlo configurations are generated that do not contribute to expectation values. Neither Green function Monte Carlo, nor its generalization designed to compute quantities at non-zero temperature[24], will be discussed in this chapter and we refer the interested reader to the literature for further details.[25−28]

Let us consider in detail the mechanism that produces large variance. This will allow us to explain what branching accomplishes if one has to compute products of many (ideally infinitely many) fluctuating weights. The time average over these products will typically be dominated by only very few large terms; the small terms are equally expensive to compute, but play no significant role in the average. This problem can be solved by performing many simultaneous Monte Carlo walks. One evolves a collection of walkers from one generation to the next and the key idea is to eliminate the light-weight walkers which produce relatively small contributions to the time average. To keep the number of walkers reasonably constant, heavy-weight walkers are duplicated and the clones are subsequently evolved (almost) independently.

An algorithm designed according to this concept does not cut off the products over weights and therefore seems to correspond to infinite projection time. It would therefore seem that the time average over a stationary branching process corresponds to an average over the exact dominant eigenstate of the Monte Carlo evolution operator, but, as we shall see, this is rigorously the case only in the limit of an infinite number of walkers[29,30]; for any finite number of walkers, the stationary distribution has a bias inversely proportional to the

number of walkers, the so-called *population control bias.* If the fluctuations in the weights are small and correlations (discussed later) decay rapidly, this bias tends to be small. In many applications this appears to be the case and the corresponding bias is statistically insignificant. However, if these methods are applied to statistical mechanical systems at the critical point, significant bias can be introduced. We shall discuss a simple method of nearly vanishing computational cost to detect this bias and correct for it in all but the worst-cases scenarios.

To discuss the branching Monte Carlo version of the power method, we continue to use the notation introduced above and again consider the Monte Carlo evolution operator $G(\mathbf{S}'|\mathbf{S})$. As above, the states $\mathbf{S}$ and $\mathbf{S}'$ will be treated here as discrete, but in practice the distinction between continuous and discrete states is a minor technicality, and generalization to the continuous case follows immediately by replacing sums by integrals and by replacing Kronecker $\delta$'s by Dirac $\delta$ functions.

To implement the power method iterations in Eq. (14) by a branching Monte Carlo process, $|u^{(t)}\rangle$ is represented by a collection of $N_t$ walkers, where a walker by definition is a state-weight pair $(\mathbf{S}_\alpha, w_\alpha)$, $\alpha = 1, \ldots, N_t$. As usual, the state variable $\mathbf{S}_\alpha$ represents a possible configuration of the system evolving according to $G$, and $w_\alpha$ represents the statistical weight of walker $\alpha$. These weights appear in averages and the efficiency of the branching Monte Carlo algorithm is realized by maintaining the weights in some range $w_\mathrm{l} < w_\alpha < w_\mathrm{u}$, where $w_\mathrm{l}$ and $w_\mathrm{u}$ are bounds introduced so as to keep all weights $w_\alpha$ of the same order of magnitude.

The first idea is to interpret a collection of walkers that make up generation $t$ as a representation of the (sparse) vector $|\underline{u}^{(t)}\rangle$ with components

$$\langle \mathbf{S}|\underline{u}^{(t)}\rangle \equiv \underline{u}^{(t)}(\mathbf{S}) = \sum_{\alpha=1}^{N_t} w_\alpha \delta_{\mathbf{S},\mathbf{s}_\alpha}, \tag{64}$$

where $\delta$ is the usual Kronecker $\delta$-function. The underbar is used to indicate that the $\underline{u}^{(t)}(\mathbf{S})$ represent a stochastic vector $|\underline{u}^{(t)}\rangle$. Of course, the same is true formally for the single thread Monte Carlo. The new feature is that one can think of the collective of walkers as a reasonably accurate representation of the stationary state at each time step, rather than in the long run.

The second idea is to define a stochastic process in which the walkers evolve with transition probabilities such that the expectation value of $c_{t+1}|\underline{u}^{(t+1)}\rangle$, as represented by the walkers of generation $t + 1$, equals $G|\underline{u}^{(t)}\rangle$ for any given collection of walkers representing $|\underline{u}^{(t)}\rangle$. It is tempting to conclude that, owing to this construction, the basic recursion relation of the power method, Eq. (14), is satisfied in an average sense, but this conclusion is not quite correct. The reason is that in practice, the constants $c_t$ are defined on the fly. Consequently, $c_{t+1}$ and $|\underline{u}^{(t+1)}\rangle$ are correlated random variables and therefore there is no guarantee that the stationary state expectation value of $|\underline{u}^{(t)}\rangle$ is *exactly* an eigenstate of $G$, except in the limit of nonfluctuating normalization constants $c_t$, which, as we shall see, is tantamount to an infinite number of walkers. More explicitly, the problem is that if one takes the time average of Eq. (14) and if the fluctuations of the $c_{t+1}$ are correlated with $|\underline{u}^{(t)}\rangle$ or $|\underline{u}^{(t+1)}\rangle$ one does not produce the same state on the left- and right-hand sides of the time-averaged equation and therefore the time-averaged state need not satisfy the eigenvalue equation. The resulting bias has been discussed in the various contexts.[12,30,31]

One way to define a stochastic process is to rewrite the power method iteration Eq. (14) as

$$u^{(t+1)}(\mathbf{S}') = \frac{1}{c_{t+1}} \sum_{\mathbf{S}} P(\mathbf{S}'|\mathbf{S})g(\mathbf{S})u^{(t)}(\mathbf{S}), \tag{65}$$

where

$$g(\mathbf{S}) = \sum_{\mathbf{S}'} G(\mathbf{S}'|\mathbf{S}) \text{ and } P(\mathbf{S}'|\mathbf{S}) = G(\mathbf{S}'|\mathbf{S})/g(\mathbf{S}). \tag{66}$$

This is in fact what how transfer matrix Monte Carlo is defined. Referring the reader back to the discussion of Eq. (29), we note that in diffusion Monte Carlo the weight $D$ is defined so that it is not just a function of the initial state $\mathbf{S}$, but also depends on the final $\mathbf{S}'$. The algorithm given below can trivially be generalized to accommodate this by making the substitution $g(\mathbf{S}) \rightarrow g(\mathbf{S}'|\mathbf{S})$.

Equation (65) describes a process represented by a Monte Carlo run which, after a few initial equilibration steps, consists of a time series of $M_0$ updates of all walkers at times labeled by $t = \ldots, 0, 1, \ldots, M_0$. The update at time $t$ consists of two steps designed to perform stochastically the matrix multiplications in Eq. (65). Following Nightingale and Blöte,[30] the process is defined by the following steps. Let us consider one of these updates, the one that transforms the generation of walkers at time $t$ into the generation at time $t+1$. We denote variables pertaining to times $t$ and $t + 1$ respectively by unprimed and primed symbols.

1. Update the old walker $(\mathbf{S}_i, w_i)$ to yield a temporary walker $(\mathbf{S}'_i, w'_i)$ according to the transition probability $P(\mathbf{S}'_i|\mathbf{S}_i)$, where $w'_i = g(\mathbf{S}_i)w_i/c'$, for $i = 1, \ldots, N_t$. Step two, defined below, can change the number of walkers. To maintain their number close to a target number, say $N_0$, choose $c' = \hat{\lambda}_0(N_t/N_0)^{1/s}$, where $\hat{\lambda}_0$ is a running estimate of the eigenvalue $\lambda_0$ to be calculated, where $s \geq 1$ [see discussion after Eq. (68)].

2. From the temporary walkers construct the new generation of walkers as follows

   (a) Split each walker $(\mathbf{S}', w')$ for which $w' > b_{\mathrm{u}}$ into two walkers $(\mathbf{S}', \frac{1}{2}w')$. The choice $b_{\mathrm{u}} = 2$ is a reasonable one.

   (b) Join pairs $(\mathbf{S}'_i, w'_i)$ and $(\mathbf{S}'_j, w'_j)$ with $w'_i < b_{\mathrm{l}}$ and $w'_j < b_{\mathrm{l}}$ to produce a single walker $(\mathbf{S}'_k, w'_i + w'_j)$, where $\mathbf{S}'_k = \mathbf{S}'_i$ or $\mathbf{S}'_k = S'_j$ with relative probabilities $w'_i$ and $w'_j$. Here $b_{\mathrm{l}} = 1/2$ is reasonable.

   (c) Walkers for which $b_{\mathrm{l}} < w'_i < b_{\mathrm{u}}$, or left single in step 2b, become members of the new generation of walkers.

Note that, if the weights $g(\mathbf{S})$ fluctuate on average more than by a factor of two, multiple split and join operations may be needed.

It may help to explicate why wildly fluctuating weight adversely impact the efficiency of the algorithm. In that case, some walkers will have multiple descendents in the next generations, whereas others will have none. This leads to an inefficient algorithm since any generation will have several walkers that are either identical or are closely related, which will produce strongly correlated contributions to the statistical time averages. In its final

analysis, this is the same old problem that we encountered in a single-thread algorithm, where averages would be dominated by few terms with relatively large, explicitly given statistical weights. Branching mitigates this problem since walkers that are descendants of a given walker eventually decorrelate, but, as discussed in Sections III and VI, the best cure is importance sampling and in practice both strategies are used simultaneously.

The algorithm described above was constructed so that for any given realization of $|\underline{u}^{(t)}\rangle$, the expectation value of $c_{t+1}|\underline{u}^{(t+1)}\rangle$, in accordance with Eq. (14), satisfies

$$\mathrm{E}\left[c_{t+1}|\underline{u}^{(t+1)}\rangle\right] = G|\underline{u}^{(t)}\rangle, \tag{67}$$

where $\mathrm{E}(\cdot)$ denotes the conditional average over the transitions defined by the preceding stochastic process. More generally, by $p$-fold iteration one finds[32]

$$\mathrm{E}\left[\left(\prod_{b=1}^{p} c_{t+b}\right)|\underline{u}^{(t+p)}\rangle\right] = G^p|\underline{u}^{(t)}\rangle. \tag{68}$$

The stationary state average of $|\underline{u}^{(t)}\rangle$ is close to the dominant eigenvector of $G$, but, as mentioned above, it has a systematic bias, proportional to $1/N_t$, when the number $N_t$ of walkers is finite. If, as is the case in some applications, this bias exceeds the statistical errors, one can again rely on the power method to reduce this bias by increasing $p$. If that is done, one is back to the old problem of having to average products of fluctuating weights, and, as usual, the variance of the corresponding estimators increases as their bias decreases. Fortunately, in practice the population control bias of the stationary state is quite small, if at all detectable, but even in those cases, expectation values involving several values of $p$ should be computed to verify the absence of population control bias. The reader is referred to Refs. 2,12,31–33 for a more detailed discussion of this problem. Suffice it to mention here, first of all, that $s$, as defined in the first step of the branching algorithm given above is the expected number of time steps it takes to restore the number of walkers to its target value $N_0$ and, secondly, that strong population control ($s = 1$) tends to introduce a stronger bias than weaker control ($s > 1$).

With Eq. (68) one constructs an estimator[32] of the dominant eigenvector $|u^{(\infty)}\rangle$ of the evolution operator $G$:

$$|\check{u}^{(p)}\rangle = \frac{1}{M_0} \sum_{t=1}^{M_0} \left(\prod_{b=0}^{p-1} c_{t-b}\right)|\underline{u}^{(t)}\rangle. \tag{69}$$

For $p = 0$, in which case the product over $b$ reduces to unity, this yields the stationary state of the branching Monte Carlo which frequently is treated as the dominant eigenstate of $G$.

Clearly, this branching Monte Carlo algorithm can be used to compute the right-projected mixed estimators that were denoted by $X_{\mathrm{TT}}^{(0,\infty)}$ in Section III. For this purpose one sets $p' = 0$ in Eq. (15) and makes the substitution $\langle u_\alpha| = \langle u_{\mathrm{T}}|$ and $G^p|u_\beta\rangle| = |\check{u}^{(p)}\rangle$. Expressions for several values of $p$ can be computed simultaneously and virtually for the price of one. We explicitly mention the important special case obtained by choosing for the general operator $X$ the evolution operator $G$ itself. This yields the following estimator for the dominant eigenvalue $\lambda_0$ of $G$:

$$\lambda_0 \approx \frac{\sum_{t=1}^{M_0} \left( \prod_{b=0}^{p} c_{t-b} \right) W^{(t)}}{\sum_{t=1}^{M_0} \left( \prod_{b=0}^{p-1} c_{t-b} \right) W^{(t-1)}}, \tag{70}$$

where

$$W^{(t)} = \sum_{i=1}^{N_t} w_i^{(t)} u_{\mathrm{T}}(\mathbf{S}_i). \tag{71}$$

In diffusion Monte Carlo this estimator can be used to construct the *growth estimate* of the ground state energy. That is, since in that special case $G \approx \exp(-\tau \mathcal{H})$, eigenvalues of the evolution operator and the Hamiltonian are related by

$$E_0 = -\frac{1}{\tau} \ln \lambda_0. \tag{72}$$

Besides expressions such as Eq. (70) one can construct expresssions with reduced variance. These involve the configurational eigenvalue of $G$ or $\mathcal{H}$ in the same way this was done in our discussion of the single-thread algorithm.

Again, in practical applications it is important to combine the raw branching algorithm defined above with importance sampling. Mathematically, this works in precisely the same way as in Section III in that one reformulates the same algorithm in terms of the similarity transform $\hat{G}$ with $u_{\mathrm{g}} = u_{\mathrm{T}}$ chosen to be an accurate, approximate dominant eigenstate [see Eq. (32)]. In the single-thread algorithm, the result is that the fluctuations of the weights $g$ and their products are reduced. In the context of the branching algorithm, this yields reduced fluctuations in the weight of walkers individually and in the size of the walker population. One result is that the population control bias is reduced. If we ignore this bias, a more fundamental difference is that the steady state of the branching algorithm is modified. That is, in the raw algorithm the walkers sample the dominant eigenstate of $G$, *i.e.*, $\psi_0(\mathbf{S})$, but, if the trial state $|u_{\mathrm{T}}\rangle$ is used for importance sampling, the distribution is $u_{\mathrm{T}}(\mathbf{S})\psi_0(\mathbf{S})$, which, of course, is simply the dominant eigenstate of $\hat{G}$.

So far, we have only discussed how mixed expectation values can be computed with the branching Monte Carlo algorithm and, as was mentioned before, this yields the desired result only if one deals with operators that commute with the evolution operator $G$. This algorithm can, however, also be used to perform power method projections to the left. In fact, most of the concepts discussed in Sections III C 1, III C 2, and III C 3 can be implemented straightforwardly. To illustrate this point we shall show how one can compute the left- and right-projected expectation value of a diagonal operator $X$. Since the branching algorithm is designed to explicitly perform the multiplication by $G$ including all weights, all that is required is the following generalization[34], called *forward* or *future walking.*

Rather than defining a walker to be the pair formed by a state and a weight, for forward walking we define the walker to be of the form $[\mathbf{S}, w, X(\mathbf{S}_{-1}), ..., X(\mathbf{S}_{-p'})]$, where $\mathbf{S}_{-1}, \mathbf{S}_{-2}, \ldots$ are previous states of the walker. In other words, each walker is equipped with a finite stack of depth $p'$ of previous values of the diagonal operator $X$. In going from one generation of walkers to the next, the state and weight of a walker are updated just as before to $\mathbf{S}'$ and $w'$. The only new feature is that the value $X(\mathbf{S})$ is pushed on the stack: $[\mathbf{S}, w, X(\mathbf{S}_{-1}), ..., X(\mathbf{S}_{-p'})] \rightarrow [\mathbf{S}', w', X(\mathbf{S}), X(\mathbf{S}_{-1}), ..., X(\mathbf{S}_{-p'+1})]$. In this way, the $p'$ times left-projected expectation value of $X$ is obtained simply by replacing $X(\mathbf{S})$ by $X(\mathbf{S}_{-p'})$. Note that one saves the history of $X$ rather than a history of configurations only for the purpose of efficient use of computer memory.

26

## VI. DIFFUSION MONTE CARLO

### A. Simple Diffusion Monte Carlo Algorithm

The diffusion Monte Carlo method[36–40] discussed in this section is an example of the general class of algorithms discussed in this chapter, all of which rely on stochastic implementation of the power method to increase the relative spectral weight of the eigenstates of interest. In the quantum mechanical context of the current section, the latter is usually the ground state. In this general sense, there is nothing new in this section. However, a number of features enter that lead to technical challenges that cannot be dealt with in a general setting. There is the problem that the projector, the evolution operator $G = \exp(-\tau\mathcal{H})$, is only known in the limit of small imaginary time $\tau$, and in addition, for applications to electronic structure problems, there are specific practical difficulties associated with nodes and cusps in the wave functions. Bosonic ground states are simpler in that they lack nodes, and we shall deal with those systems only in passing. As in the rest of this chapter, we deal only with the basic concepts. In particular, we focus on considerations relevant to the design of an efficient diffusion Monte Carlo algorithm. The reader is referred to Refs. 41–43 for applications and other issues not covered here.

For quantum mechanical problems, the power method iteration Eq. (14) takes the form

$$|\psi(t+\tau)\rangle = e^{-(\mathcal{H}-E_\mathrm{T})\tau}\, |\psi(t)\rangle. \tag{73}$$

Here $E_\mathrm{T}$ is a shift in energy such that $E_0 - E_\mathrm{T} \approx 0$, where $E_0$ is the ground state energy. In the real-space representation we have

$$\psi(\mathbf{R}', t+\tau) = \int d\mathbf{R}\ \langle\mathbf{R}'|e^{-(\mathcal{H}-E_\mathrm{T})\tau}|\mathbf{R}\rangle\ \langle\mathbf{R}|\psi(t)\rangle = \int d\mathbf{R}\ G(\mathbf{R}',\mathbf{R},\tau)\ \psi(\mathbf{R},t), \tag{74}$$

In practical Monte Carlo settings, the shift $E_\mathrm{T}$ is computed on the fly and consequently is a slowly varying, nearly constant function of time, but for the moment we take it to be constant. The wavefunction $\psi(\mathbf{R},t)$ is the solution to the Schrödinger equation in imaginary time

$$-\frac{1}{2\mu}\nabla^2\psi(\mathbf{R},t) + [\mathcal{V}(\mathbf{R}) - E_\mathrm{T}]\psi(\mathbf{R},t) = -\frac{\partial\psi(\mathbf{R},t)}{\partial t}. \tag{75}$$

To make contact with Eq. (29) we should factor the Green function into a probability conserving part and a weight. For small times, this can be accomplished by the following approximation. If, on the left-hand side, we just had the first term, Eq. (75) would reduce to a diffusion equation, whence the method gets its name. For diffusion, the Green function is probability conserving and is given by

$$P(\mathbf{R}',\mathbf{R},\tau) = \frac{e^{-\mu(\mathbf{R}'-\mathbf{R})^2/2\tau}}{(2\pi\tau/\mu)^{3n/2}}. \tag{76}$$

If, on the other hand, we just had the second term then Eq. (75) would reduce to a rate equation for which the Green function is $\delta(\mathbf{R}' - \mathbf{R})$ with prefactor

$$g(\mathbf{R}',\mathbf{R},\tau) = e^{\tau\{E_\mathrm{T}-[\mathcal{V}(\mathbf{R})+\mathcal{V}(\mathbf{R}')]/2\}}. \tag{77}$$

27

In combining these ingredients, we have to contend with the following general relation for noncommuting operators $\mathcal{H}_i$

$$e^{(\mathcal{H}_1+\mathcal{H}_2)\tau} = e^{\frac{1}{2}\mathcal{H}_1\tau}e^{\mathcal{H}_2\tau}e^{\frac{1}{2}\mathcal{H}_1\tau} \; + \; \mathcal{O}(\tau^3) = e^{\mathcal{H}_1\tau}e^{\mathcal{H}_2\tau} \; + \; \mathcal{O}(\tau^2). \tag{78}$$

As long as one is dealing with a $\delta$-function, the weight in Eq. (77) is evaluated always at $\mathbf{R}' = \mathbf{R}$, and therefore the expression on the right can be written also in non-symmetric form. However, Eq. (78) suggests that the exponent in Eq. (77) should be used in symmetric fashion as written. This is indeed the form we shall employ for the time being, but we note that the final version of the diffusion algorithm employs a nonsymmetric split of the exponent, since proposed moves are not always accepted. Since there are other sources of $\mathcal{O}(\tau^2)$ contributions anyway this does not degrade the performance of the algorithm. Combination of the preceding ingredients yields the following approximate, short-time Green function for Eq. (75)

$$\begin{aligned} G(\mathbf{R}', \mathbf{R}, \tau) &= \frac{e^{-\mu(\mathbf{R}'-\mathbf{R})^2/2\tau}}{(2\pi\tau/\mu)^{3n/2}} \; e^{\tau\{E_\mathrm{T}-[\mathcal{V}(\mathbf{R}')+\mathcal{V}(\mathbf{R})]/2\}} \; + \; \mathcal{O}(\tau^3) \\ &= \frac{e^{-\mu(\mathbf{R}'-\mathbf{R})^2/2\tau}}{(2\pi\tau/\mu)^{3n/2}} \; e^{\tau[E_\mathrm{T}-\mathcal{V}(\mathbf{R})]} \; + \; \mathcal{O}(\tau^2) \end{aligned} \tag{79}$$

For this problem, the power method can be implemented by Monte Carlo by means of both the single-thread scheme discussed in Section III and the branching algorithm of Section V. We shall use the latter option in our discussion. In this case, one performs the following steps. Walkers of the zeroth generation are sampled from $\psi(\mathbf{R}, 0) = \psi_\mathrm{T}(\mathbf{R})$ using the Metropolis algorithm. The walkers are propagated forward an imaginary time $\tau$ by sampling a new position $\mathbf{R}'$ from a multivariate Gaussian centered at the old position $\mathbf{R}$ and multiplying the weights of the walkers by the second factor in Eq. (79). Then the split/join step of the branching algorithm is performed to obtain the next generation of walkers, with weights in the targeted range.

### 1. Diffusion Monte Carlo with Importance Sampling

For many problems of interest, the potential energy $\mathcal{V}(\mathbf{R})$ exhibits large variations over coordinate space and in fact may diverge at particle coincidence points. As we have discussed in the general case, the fluctuations of weights $g$, produce noisy statistical estimates. As described in Sections III and V, this problem can be greatly mitigated by applying the similarity (or importance sampling) transformation[36,27] to the evolution operator. Employing the general mathematical identity $S \exp(-\tau\mathcal{H})S^{-1} = \exp(-\tau S\mathcal{H}S^{-1})$, this transformation can be applied conveniently to the Hamiltonian. That is, given a trial function $\psi_\mathrm{T}(\mathbf{R})$ one can introduce a distribution $f(\mathbf{R}, t) = \psi_\mathrm{T}(\mathbf{R})\psi(\mathbf{R}, t)$. If $\psi(\mathbf{R}, t)$ satisfies the Schrödinger equation [Eq. (75)], it is a simple calculus exercise to show that $f$ is a solution of the equation[39,40]

$$\begin{aligned} &\psi_\mathrm{T}(\mathbf{R})(\mathcal{H} - E_\mathrm{T})\psi_\mathrm{T}(\mathbf{R})^{-1}f(\mathbf{R}, t) = \\ &-\frac{1}{2\mu}\vec{\nabla}^2 f(\mathbf{R}, t) + \vec{\nabla} \cdot [\mathbf{V}(\mathbf{R})f(\mathbf{R}, t)] - S(\mathbf{R})f(\mathbf{R}, t) = -\frac{\partial f(\mathbf{R}, t)}{\partial t}. \end{aligned} \tag{80}$$

Here the *velocity* $\mathbf{V}$ is a *function* (not an operator), often referred to in the literature as the quantum force, and is given by

$$\mathbf{V}(\mathbf{R}) = (\mathbf{v}_1, \ldots, \mathbf{v}_n) = \frac{1}{\mu} \frac{\vec{\nabla}\psi_{\mathrm{T}}(\mathbf{R})}{\psi_{\mathrm{T}}(\mathbf{R})}. \tag{81}$$

The coefficient of the source term, which is responsible for branching in the diffusion Monte Carlo context, is

$$S(\mathbf{R}) = E_{\mathrm{T}} - E_{\mathrm{L}}(\mathbf{R}), \tag{82}$$

which is defined in terms of the *local energy*

$$E_{\mathrm{L}}(\mathbf{R}) = \frac{\mathcal{H}\psi_{\mathrm{T}}(\mathbf{R})}{\psi_{\mathrm{T}}(\mathbf{R})} = -\frac{1}{2\mu} \frac{\vec{\nabla}^2\psi_{\mathrm{T}}(\mathbf{R})}{\psi_{\mathrm{T}}(\mathbf{R})} + \mathcal{V}(\mathbf{R}), \tag{83}$$

the equivalent of the configurational eigenvalue introduced in Eq. (18).

Compared to the original Schrödinger equation, to which of course it reduces for the case $\psi_{\mathrm{T}} \equiv 1$, the second term in Eq. (80) is new, and corresponds to drift. Again, one can explicitly write down the Green function of the equation with just a single term on the left-hand side. The drift Green function $G_{\mathrm{D}}$ of the equation obtained by suppressing all but this term is

$$G_{\mathrm{D}}(\mathbf{R}', \mathbf{R}, \tau) = \delta[\mathbf{R}' - \tilde{\mathbf{R}}(\tau)] \tag{84}$$

where $\tilde{\mathbf{R}}(\tau)$ satisfies the differential equation

$$\frac{d\tilde{\mathbf{R}}}{dt} = \mathbf{V}(\tilde{\mathbf{R}}) \tag{85}$$

subject to the boundary condition $\tilde{\mathbf{R}}(0) = \mathbf{R}$. Again, at short times the noncommutativity of the various operators on the left-hand side of the equation can be ignored and thus one obtains the following short-time Green function.

$$G(\mathbf{R}', \mathbf{R}, \tau) = \int d\mathbf{R}'' \; \delta\left[\mathbf{R}'' - \tilde{\mathbf{R}}(\tau)\right] \frac{e^{-\mu(\mathbf{R}'-\mathbf{R}'')^2/2\tau}}{(2\pi\tau/\mu)^{3n/2}} \; e^{\tau\{E_{\mathrm{T}} - [E_{\mathrm{L}}(\mathbf{R}') + E_{\mathrm{L}}(\mathbf{R})]/2\}} \; + \; \mathcal{O}(\tau^2)$$

$$= \frac{e^{-\mu[\mathbf{R}'-\tilde{\mathbf{R}}(\tau)]^2/2\tau}}{(2\pi\tau/\mu)^{3n/2}} \; e^{\tau\{E_{\mathrm{T}} - [E_{\mathrm{L}}(\mathbf{R}') + E_{\mathrm{L}}(\mathbf{R})]/2\}} \; + \; \mathcal{O}(\tau^2). \tag{86}$$

Eq. (86) again can be viewed in our general framework by defining the probability conserving generalization of Eq. (76)

$$P(\mathbf{R}', \mathbf{R}, \tau) = \frac{e^{-\mu[\mathbf{R}'-\tilde{\mathbf{R}}(\tau)]^2/2\tau}}{(2\pi\tau/\mu)^{3n/2}} \tag{87}$$

and the remainder of the Green function is $\delta[\mathbf{R}' - \tilde{\mathbf{R}}(\tau)]$ with prefactor

$$g(\mathbf{R}', \mathbf{R}, \tau) = e^{\tau\{E_{\mathrm{T}} - [E_{\mathrm{L}}(\mathbf{R}') + E_{\mathrm{L}}(\mathbf{R})]/2\}}, \tag{88}$$

which is the analog of Eq. (77).

When employed for the branching Monte Carlo algorithm, the factorization given in Eqs. (87) and (88) differs from the original factorization Eqs. (76) and (77) in two respects: (a) the walkers do not only diffuse but also drift towards the important regions, *i.e.,* in the direction in which $|\psi_T(\mathbf{R})|$ is increasing; and (b) the branching term is better behaved since it depends on the local energy rather than the potential. In particular, if the trial function $\psi_T(\mathbf{R})$ obeys cusp conditions[44] then the local energy at particle coincidences is finite even though the potential may be infinite. If one were so fortunate that $\psi_T(\mathbf{R})$ is the exact ground state, the branching factor would reduce to a constant, which can be chosen to be unity by choosing $E_T$ to be the exact energy of that state.

The expressions, as written, explicitly contain the expression $\tilde{\mathbf{R}}(\tau)$, which has to be obtained by integration of the velocity $\mathbf{V}(\mathbf{R})$. Since we are using a short-time expansion anyway, this exact expression may be replaced by the approximation

$$\tilde{\mathbf{R}}(\tau) = \mathbf{R} + \mathbf{V}(\mathbf{R})\tau + \mathcal{O}(\tau^2). \qquad (89)$$

In the improvements discussed below, designed to reduce time-step error, this expression is improved upon so that regions where $\mathbf{V}$ diverges do not make large contributions to the overall time-step error.

## 2. Fixed-Node Approximation

The absolute ground state of a Hamiltonian with particle exchange symmetry is bosonic. Consequently, unless one starts with a fermionic, *i.e.,* antisymmetric wavefunction and implements the power method in a way that respects this antisymmetry, the power method will reduce the spectral weight of the fermionic eigenstate to zero relative to the weight of the bosonic state. The branching algorithm described above assumes all weights are positive and therefore is incompatible with the requirement of preserving antisymmetry. The algorithm needs modification, if we are interested in the fermionic ground state.

If the nodes of the fermionic ground state were known, they could be imposed as boundary conditions[37] and the problem could be dealt with by solving the Schrödinger equation within a single, connected region bounded by the nodal surface, a region we shall refer to as a *nodal pocket.* Since all the nodal pockets of the ground state of a fermionic system are equivalent, this would yield the exact solution of the problem everywhere. Unfortunately, the nodes of the wavefunction of an $n$-particle system form a $(3n-1)$-dimensional surface, which should not be confused with the nodes of single-particle orbitals. Of this full surface, in general, only the $(3n-3)$-dimensional subset, corresponding to the coincidence of two like-spin particles, is known. Hence, we are forced to employ an approximate nodal surface as a boundary condition to be satisfied by the solution of the Schrödinger equation. This is called the *fixed-node* approximation. Usually, one chooses for this purpose the nodal surface of an optimized trial wave function[37], and such nodes can at times yield a very accurate results if sufficient effort is invested in optimizing the trial wavefunction.

Since the imposition of the boundary condition constrains the solution, it is clear that the fixed-node energy is an upper bound on the true fermionic ground state energy. In diffusion Monte Carlo applications, the fixed-node energy typically has an error which is

five to ten times smaller than the error of the variational energy corresponding to the same trial wavefunction, though this can vary greatly depending on the system and the trial wavefunction employed.

For the Monte Carlo implementation of this approach one has to use an approximate Green function, which, as we have seen, may be obtained by iteration of a short time approximation. To guide the choice of an approximant accurate over a wide time range, it is useful to consider some general properties of the fixed-node approximation. Mathematically, the latter amounts to solution of the Schrödinger equation in a potential that equals the original physical potential inside the nodal pocket of choice, and is infinite outside. The corresponding eigenstates of the Hamiltonian are continuous and vanish outside the nodal pocket. Note that the solution can be analytically continued outside the initial nodal pocket only if the nodal surface is exact, otherwise there is a derivative discontinuity at the nodal surface. These properties are shared by the Green function consistent with the boundary conditions. This can be seen by writing down the spectral decomposition for the evolution operator in the position representation

$$G(\mathbf{R}', \mathbf{R}, \tau) = \langle \mathbf{R}'|e^{-\tau\mathcal{H}}|\mathbf{R}\rangle = \sum_i \psi_i(\mathbf{R}')e^{-\tau E_i}\psi_i(\mathbf{R}) \tag{90}$$

where the $\psi_i$ are the eigenstates satisfying the required boundary conditions. For notational convenience only, we have assumed that the spectrum has a discrete part only and that the wavefunctions can be chosen to be real. The Green function vanishes outside the nodal pocket and generically vanishes linearly at the nodal surface, as do the wavefunctions.

The Green function of interest in practical applications is the one corresponding to importance sampling, the similarity transform of Eq. (90)

$$\hat{G}(\mathbf{R}', \mathbf{R}, \tau) = \psi_{\mathrm{T}}(\mathbf{R}') \langle \mathbf{R}'|e^{-\tau\mathcal{H}}|\mathbf{R}\rangle \frac{1}{\psi_{\mathrm{T}}(\mathbf{R})} \tag{91}$$

This Green function vanishes at the nodes quadratically in its first index, which, in the Monte Carlo context is the one that determines to which state a transition is made from a given initial state.

The approximate Green functions of Eqs. (79) and (86) have tails that extend beyond the nodal surface and, consequently, walkers sampled from these Green functions have a finite probability of attempting to cross the node. Since expectation values ought to be calculated in the $\tau \to 0$ limit the relevant quantity to consider is what fraction of walkers attempt to cross the node *per unit time* in the $\tau \to 0$ limit. If the Green function of Eq. (79) is employed, this is a finite number, whereas, if the importance-sampled Green function of Eq. (86) is employed, no walkers cross the surface since the velocity $\mathbf{V}$ is directed away from the nodes and diverges at the nodes. In practice, of course, the calculations are performed for finite $\tau$, but the preceding observation leads to the conclusion that in the former case it is necessary to reduce to zero the weight of a walker that has crossed a node, *i.e.,* to kill the walker, while in the latter case one can either kill the walkers or reject the proposed move, since in the $\tau \to 0$ limit they yield the same result.

We now argue that, for finite $\tau$, rejecting moves is the choice with the smaller time-step error when the importance-sampled Green function is employed. Sufficiently close to a node, the component of the velocity perpendicular to the node dominates all other terms in the

Green function and it is illuminating to consider a free particle in one dimension subject to the boundary condition that $\psi$ have a node at $x = 0$. The exact Green function for this problem is

$$G(x', x, \tau) = \frac{1}{\sqrt{2\pi\tau/\mu}}[e^{-\mu(x'-x)^2/2\tau} - e^{-\mu(x'+x)^2/2\tau}], \qquad (92)$$

while the corresponding importance sampled Green function is

$$\hat{G}(x', x, \tau) = \frac{x'}{x\sqrt{2\pi\tau/\mu}}[e^{-\mu(x'-x)^2/2\tau} - e^{-\mu(x'+x)^2/2\tau}]. \qquad (93)$$

We note that the integral of the former, over the region $x > 0$, is less than one and decreases with time. In terms of the usual language used for diffusion problems, this is because of absorption at the $x = 0$ boundary. In our case, this provides the mathematical justification for killing the walkers that cross. On the other hand, the integral of the Green function of Eq. (93) equals one. Consequently, for finite $\tau$ it seems likely that algorithms that reject moves across the node, such as the one discussed in Section VI B yield a better approximation than algorithms that kill the walkers that cross the node.

As mentioned, it can be shown that all the nodal pockets of the ground state of a fermionic system are equivalent and trial wavefunctions are constructed to have the same property. Consequently, the Monte Carlo averages will not depend on the initial distribution of the walkers over the nodal pockets. The situation is more complicated for excited states, since different nodal pockets of excited-state wavefunctions are not necessarily equivalent, neither for bosons nor for fermions. Any initial state with walkers distributed randomly over nodal pockets pockets, will evolve to a steady state distribution with walkers only in the pocket with the lowest average local energy, at least if we ignore multiple-node crossings and assume a sufficiently large number of walkers, so that fluctuations in the average local energy can be ignored.

### 3. Problems with Simple Diffusion Monte Carlo

The diffusion Monte Carlo algorithm corresponding to Eq. (86) is in fact not viable for a wavefunction with nodes for the following two reasons. Firstly, in the vicinity of the nodes the local energy of the trial function $\psi_T$ diverges inversely proportional to the distance to the nodal surface. For nonzero $\tau$, there is a nonzero density of walkers at the nodes. Since the nodal surface for a system with $n$ electrons is $3n - 1$ dimensional, the variance of the local energy diverges for any finite $\tau$. In fact, the expectation value of the local energy also diverges, but only logarithmically. Secondly, the velocity of the electrons at the nodes diverges inversely as the distance to the nodal surface. The walkers that are close to a node at one time step, drift at the next time step to a distance inversely proportional to the distance from the node. This results in a charge distribution with a component that falls off as the inverse square of distance from the atom or molecule, whereas in reality the decay is exponential. These two problems are often remedied by introducing cut-offs in the values of the local energy and the velocity[45,46], chosen such that they have no effect in the $\tau \to 0$ limit, so that the results extrapolated to $\tau = 0$ are correct. In the next section better remedies are presented.

## B. Improved Diffusion Monte Carlo Algorithm

### 1. The limit of Perfect Importance Sampling

In the limit of perfect importance sampling, that is if $\psi_{\mathrm{T}}(\mathbf{R}) = \psi_0(\mathbf{R})$, the energy shift $E_{\mathrm{T}}$ can be chosen such that the branching term in Eq. (80) vanishes identically for all $\mathbf{R}$. In this case, even though the energy can be obtained with zero variance, the steady state distribution of the diffusion Monte Carlo algorithm discussed above is only approximately the desired distribution $\psi_{\mathrm{T}}^2$, because of the time-step error in the Green function. However, since one has an explicit expression, $\psi_{\mathrm{T}}^2$, for the distribution to be sampled, it is possible to use the Metropolis algorithm, described in Section III A, to sample the desired distribution exactly. Although the ideal $\psi_{\mathrm{T}}(\mathbf{R}) = \psi_0(\mathbf{R})$ is never achieved in practice, this observation leads one to devise an improved algorithm that can be used when moderately good trial wavefunctions are known.

If for the moment we ignore the branching term in Eq. (80), then we have the equation

$$-\frac{1}{2\mu}\vec{\nabla}^2 f + \vec{\nabla} \cdot (\mathbf{V} f) = -\frac{\partial f}{\partial t}. \tag{94}$$

This equation has a known steady-state solution $f = \psi_{\mathrm{T}}^2$ for any $\psi_{\mathrm{T}}$, which in the limit of perfect importance sampling is the desired distribution. However, the approximate drift-diffusion Green function used in the Monte Carlo algorithm defined by Eq. (86) without the branching factor, is not the exact Green function of Eq. (94). Therefore, for any finite time step $\tau$, we do not obtain $\psi_{\mathrm{T}}^2$ as a steady state, even in the ideal case. Following Reynolds *et al.*[39], we can change the algorithm in such a way that it *does* sample $\psi_{\mathrm{T}}^2$ in the ideal case, which also reduces the time-step error in nonideal, practical situations. This is accomplished by using a generalized[47−49] Metropolis algorithm[14]. The approximate drift-diffusion Green function is used to propose moves, which are then accepted with probability

$$p = \min\left(\frac{\tilde{G}(\mathbf{R}, \mathbf{R}\,', \tau)\psi_{\mathrm{T}}(\mathbf{R}\,')^2}{\tilde{G}(\mathbf{R}\,', \mathbf{R}, \tau)\psi_{\mathrm{T}}(\mathbf{R})^2}, 1\right) \equiv 1 - q, \tag{95}$$

in accordance with the detailed balance condition.

As was shown above, the true fixed-node Green function vanishes outside the nodal pocket of the trial wavefunction. However, since we are using an approximate Green function, moves across the nodes will be proposed for any finite $\tau$. To satisfy the boundary conditions of the fixed-node approximation these proposed moves are always rejected.

If we stopped here, we would have an exact and efficient variational Monte Carlo algorithm to sample from $\psi_{\mathrm{T}}^2$. Now, we reintroduce the branching term to convert the steady-state distribution from $\psi_{\mathrm{T}}^2$ to $\psi_{\mathrm{T}}\psi_0$. This is accomplished by reweighting the walkers with the branching factor [see Eq. (86)]

$$\Delta w = \begin{cases} \exp\{\frac{1}{2}[S(\mathbf{R}\,') + S(\mathbf{R})]\tau_{\mathrm{eff}}\} & \text{for an accepted move,} \\ \exp[S(\mathbf{R})\tau_{\mathrm{eff}}] & \text{for a rejected move,} \end{cases} \tag{96}$$

where $S$ is defined in Eq. (82). An effective time step $\tau_{\mathrm{eff}}$, which will be defined presently, is required because the Metropolis algorithm introduces a finite probability of not moving

forward and rejecting the proposed configuration. Before defining $\tau_{\text{eff}}$, we note that an alternative to expression (96) is obtained by replacing the two reweighting factors by a single expression,

$$\Delta w = \exp\left[\left\{\frac{p}{2}\left(S(\mathbf{R}') + S(\mathbf{R})\right) + qS(\mathbf{R})\right\}\tau_{\text{eff}}\right] \quad \text{for all moves} \tag{97}$$

This expression, written down with Eq. (26) in mind, yields somewhat smaller fluctuations and time-step error than expression (96).

Following Reynolds *et al.*[39], an effective time step $\tau_{\text{eff}}$ is introduced in Eq.( 97) to account for the changed rate of diffusion. We set

$$\tau_{\text{eff}} = \tau\frac{\langle p\,\Delta R^2\rangle}{\langle\Delta R^2\rangle}, \tag{98}$$

where the angular brackets denote the average over all attempted moves, and $\Delta R$ are the displacements resulting from diffusion. This equals $\langle\Delta R^2\rangle_{\text{accepted}}/\langle\Delta R^2\rangle$ but again has somewhat smaller fluctuations.

An estimate of $\tau_{\text{eff}}$ is readily obtained iteratively from sets of equilibration runs. During the initial run, $\tau_{\text{eff}}$ is set equal to $\tau$. For the next runs, the value of $\tau_{\text{eff}}$ is obtained from the values of $\tau_{\text{eff}}$ computed with Eq. (98) during the previous equilibration run. In practice, this procedure converges in two iterations, which typically consume less than 2% of the total computation time. Since the statistical errors in $\tau_{\text{eff}}$ affect the results obtained, the number of Monte Carlo steps performed during the equilibration phase needs to be sufficiently large that this is not a major component of the overall statistical error.

The value of $\tau_{\text{eff}}$ is a measure of the rate at which the Monte Carlo process generates uncorrelated configurations, and thus a measure of the efficiency of the computation. Since the acceptance probability decreases when $\tau$ increases, $\tau_{\text{eff}}$ has a maximum as a function of $\tau$. However, since the time-step error increases with $\tau$, it is advisable to use values of $\tau$ that are smaller than this "optimum".

Algorithms that do not exactly simulate the equilibrium distribution of the drift-diffusion equation if the branching term is suppressed, *i.e.*, algorithms that do not use the Metropolis accept/reject mechanism, can for sufficiently large $\tau$ have time-step errors that make the energy estimates higher than the variational energy. On the other hand, if the drift-diffusion terms are treated exactly by including an accept/reject step, the energy, evaluated for any $\tau$, must lie below the variational energy, since the branching term enhances the weights of the low-energy walkers relative to that of the high-energy walkers.

### 2. Persistent Configurations

As mentioned above, the accept/reject step has the desirable feature of yielding the exact electron distribution in the limit that the trial function is the exact ground state. However, in practice the trial function is less than perfect and as a consequence the accept/reject procedure can lead to the occurrence of persistent configurations, as we now discuss.

For a given configuration $\mathbf{R}$, consider the quantity $Q = \langle q\Delta w\rangle$, where $q$ and $\Delta w$ are the rejection probability and the branching factor given by Eqs. (95) and (97). The average in
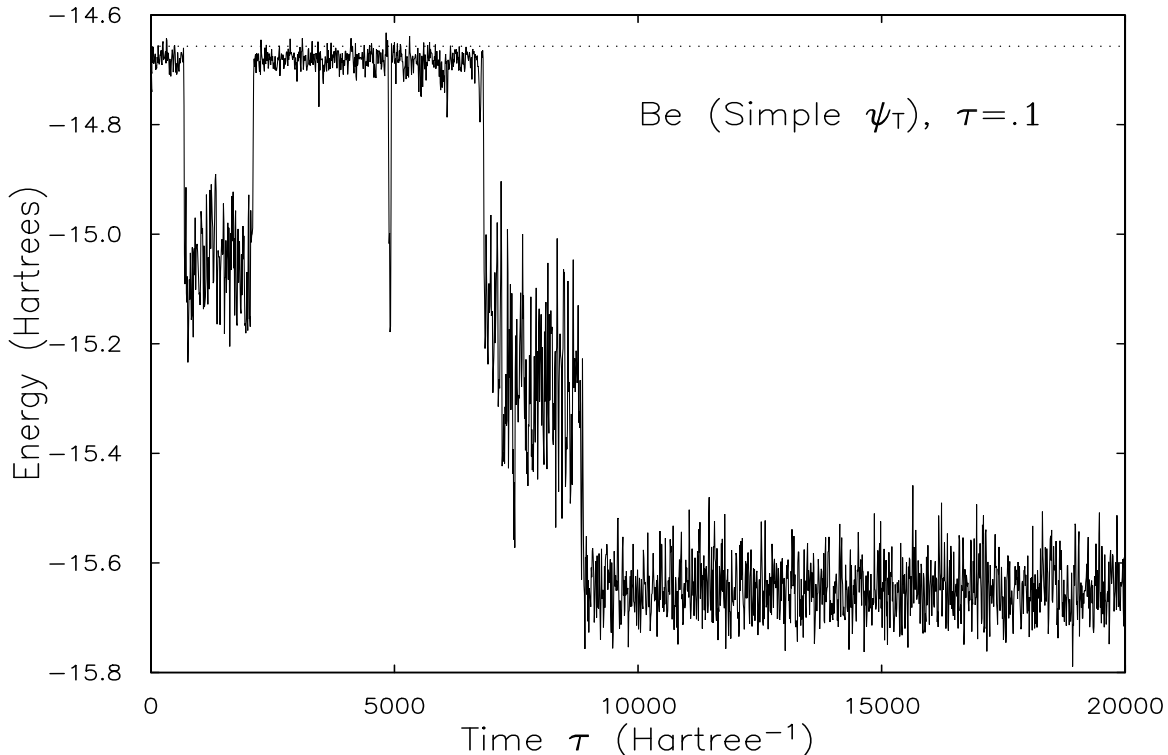
FIG. 2. Illustration of the persistent configuration catastrophe. The dotted horizontal line is the true fixed-node energy for a simple Be wavefunction extrapolated to $\tau = 0$.

the definition of $Q$ is over all possible moves for the configuration $\mathbf{R}$ under consideration. If the local energy at $\mathbf{R}$ is relatively low and if $\tau_{\text{eff}}$ is sufficiently large, $Q$ may be in excess of one. In that case, the weight of the walker at $\mathbf{R}$, or more precisely, the total weight of all walkers in that configuration will increase with time, except for fluctuations, until the time-dependent trial energy $E_{\text{T}}$ adjusts downward to stabilize the total population. This population contains on average a certain number of copies of the persistent configuration. Since persistent configurations must necessarily have an energy that is lower than the true fixed-node energy, this results in a negatively biased energy estimate. The persistent configuration may disappear because of fluctuations, but the more likely occurrence is that it is replaced by another configuration that is even more strongly persistent, *i.e.*, one that has an even larger value of $Q = \langle q \Delta w \rangle$. This process produces a cascade of configurations of ever decreasing energies. Both sorts of occurrences are demonstrated in Fig. 2. Persistent configurations are most likely to occur near nodes, or near nuclei if $\psi_{\text{T}}$ does not obey the cusp conditions. Improvements to the approximate Green function in these regions, as discussed in the next section, help to reduce greatly the probability of encountering persistent configurations to the point that they are never encountered even in long Monte Carlo runs.

Despite the fact that the modifications described in the next section eliminate persistent configurations for the systems we have studied, it is clearly desirable to have an algorithm that cannot display this pathology even in principle. One possible method is to replace $\tau_{\text{eff}}$ in Eq. (96) by $\tau$ for an accepted move and by zero for a rejected move. This ensures that $\Delta w$ never exceeds unity for rejected moves, hence eliminating the possibility of persistent configurations. Further, this has the advantage that it is not necessary to determine $\tau_{\text{eff}}$.

35

Other possible ways to eliminate the possibility of encountering persistent configurations are discussed in Ref. 50.

### 3. Singularities

The number of iterations of Eq. (74) required for the power method to converge to the ground state grows inversely with the time step $\tau$. Thus, the statement made above, *viz.* that the Green function of Eq. (79) is in error only to $\mathcal{O}(\tau^2)$, would seem to imply that the errors in the electron distribution and the averages calculated from the short-time Green function are of $\mathcal{O}(\tau)$. However, the presence of non-analyticities and divergences in the local energy and the velocity may invalidate this argument: the short-time Green function may lack uniform convergence in $\tau$ over $3n$-dimensional configuration space. Consequently, an approximation that is designed to be better behaved in the vicinity of singularities and therefore behaves more uniformly over space may outperform an approximation that is correct to a higher order in $\tau$ for generic points in configuration space, but ignores these singularities.

Next we discuss some important singularities that one may encounter and their implications for the diffusion Monte Carlo algorithm. The behavior of the local energy $E_L$ and velocity $\mathbf{V}$ near nodes of $\psi_T$ are described in Table I. Although the true wave function $\psi_0$ has a constant local energy $E_0$ at all points in configuration space, the local energy of $\psi_T$ diverges at most points of the nodal surface of $\psi_T$ for almost all the $\psi_T$ that are used in practice, and it diverges at particle coincidences (either electron-nucleus or electron-electron) for wave functions that fail to obey cusp conditions[44]. The velocity $\mathbf{V}$ diverges at the nodes and for the Coulomb potential has a discontinuity at particle coincidences both for approximate and for the true wavefunction. For the nodeless wavefunction of Lennard-Jones particles in their ground state,[5] $\mathbf{V}$ diverges as $r^{-6}$ in the inter-particle distance $r$. Since the only other problem for these bosonic systems is the divergence of $E_L$ at particle coincidences, we continue our discussion for electronic systems and refer the interested reader to the literature[5] for details.

TABLE I. Behavior of the local energy $E_L$ and velocity $v$ as a function of the distance $R_\perp$ of an electron to the nearest singularity. The behavior of various quantities is shown for an electron approaching a node or another particle, either a nucleus or an electron. The singularity in the local energy at particle overlap is only present for a $\psi_T$ that fails to satisfy the cusp conditions.

| Region | Local energy | Velocity |
|---|---|---|
| Nodes | $E_L \sim \pm\frac{1}{R_\perp}$ for $\psi_T$ <br> $E_L = E_0$ for $\psi_0$ | $v \sim \frac{1}{R_\perp}$ |
| Electron-nucleus/electron | $E_L \sim \frac{1}{x}$ for some $\psi_T$ <br> $E_L = E_0$ for $\psi_0$ | $v$ has a discontinuity <br> for both $\psi_T$ and $\psi_0$ |

The divergences in the local energies cause large fluctuations in the population size: negative divergences lead to large local population densities and positive divergences lead to
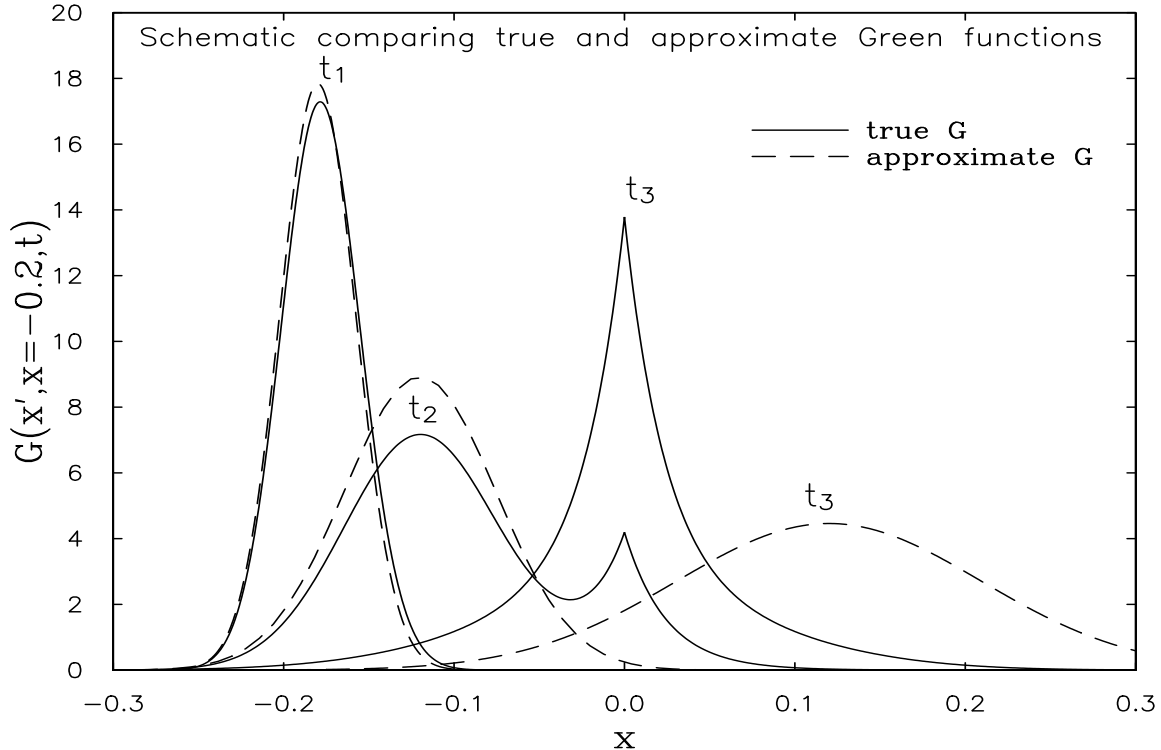
FIG. 3. Schematic comparing the qualitative behaviors of the true $G$ with the approximate $G$ of Eq. (86) of an electron that is located at $x = -0.2$ at time $t = 0$ and evolves in the presence of a nearby nucleus located at $x = 0$. The Green functions are plotted for three times: $t_1 < t_2 < t_3$.

small ones. The divergence of $\mathbf{V}$ at the nodes typically leads to a proposed next state of a walker in a very unlikely region of configuration space and is therefore likely to be rejected. The three-dimensional velocity of an electron which is close to a nucleus is directed towards the nucleus. Hence the true Green function, for sufficiently long time, exhibits a peak at the nucleus, but the approximate Green function of Eq. (86) cause electrons to overshoot the nucleus. This is illustrated in Fig. 3, where we show the qualitative behavior of the true Green function and of the approximate Green function for an electron which starts at $x = -0.2$ in the presence of a nearby nucleus at $x = 0$. At short time $t_1$, the approximate Green function of Eq. (86) agrees very well with the true Green function. At a longer time $t_2$, the true Green function begins to develop a peak at the nucleus which is absent in the approximate Green function, wheras at a yet longer time $t_3$, the true Green function is peaked at the nucleus while the approximate Green function has overshot the nucleus.

The combined effect of these nonanalyticities is a large time-step error, which can be of either sign in the energy, and large statistical uncertainty in the computed expectation values. We now give a brief description of how these nonanalyticities are treated. The divergence of the local energy at particle coincidences is cured simply by employing wavefunctions that obey cusp conditions[44]. The other nonanalyticities are addressed by employing a modification of the Green function of Eq. (86) such that it incorporates the divergence of $E_{\mathrm{L}}$ and $\mathbf{V}$ at nodes and the discontinuity in $\mathbf{V}$ at particle coincidences but smoothly reduces to Eq. (86) in the short-time limit or in the limit that the nearest nonanalyticity is far away. The details can be found in Ref. 50. The modified algorithm has a time-step error which

is two to three orders of magnitude smaller[50] than the simple algorithm corresponding to Eq. (86) with cutoffs imposed on $E_\mathrm{L}$ and $\mathbf{V}$.

We have used the application to all-electron electronic structure calculations to illustrate the sort of problems that can lead to large time-step errors and their solution. Other systems may exhibit only a subset of these problems or a modified version of them. For example, in calculations of bosonic clusters[5] there are no nodes to contend with, while, in electronic structure calculations employing pseudopotentials[51−54], or pseudo-Hamiltonians[55] the potential need not diverge at electron-nucleus coincidences. We note, in passing, that the use of the latter methods has greatly extended the practical applicability of quantum Monte Carlo methods to relatively large systems of practical interest[53,56] but at the price of introducing an additional approximation[57].

## VII. CLOSING COMMENTS

The material presented above was selected to describe from a unified point of view Monte Carlo algorithms as employed in seemingly unrelated areas in quantum and statistical mechanics. Details of applications were given only to explain general ideas or important technical problems, such as encountered in diffusion Monte Carlo. We ignored a whole body of literature, but we wish to just mention a few topics. Domain Green function Monte Carlo[25−28] is one that comes very close to topics that were covered. In this method the Green function is sampled exactly by iterating upon an approximate Green function. Infinite iteration, which the Monte Carlo method does in principle, produces the exact Green function. Consequently, this method lacks a time-step error, and in this sense has the advantage of being exact. In practice, there are other reasons, besides the time-step error that force the algorithm to move slowly through state space, and currently the available algorithms seem to be less efficient than diffusion Monte Carlo, even when one accounts for the effort required to perform the extrapolation to vanishing time step.

Another area that we just touched upon in passing is path integral Monte Carlo.[58] Here we remind the reader that path integral Monte Carlo is a particularly appealing alternative for the evaluation of matrix elements such as such as $X_{\alpha\beta}^{(p',p)}$ in Eq. (15). The advantage of this method is that no products of weights appear, but the disadvantage is that it seems to be more difficult to move rapidly through state space. This is a consequence of the fact that branching algorithms propagate just a single time slice through state space, whereas path integral methods deal with a whole stack of slices, which for sampling purposes tends to produce a more rigid object. Finally, it should be mentioned that we ignored the vast literature on quantum lattice systems[59].

In splitting the evolution operator into weights and probabilities [see Eq. (33)] we assumed that the weights were non-negative. To satisfy this requirement, the fixed-node approximation was employed for fermionic systems. An approximation in the same vein is the fixed-phase approximation,[60] which allows one to deal with systems in which the wavefunction is necessarily complex valued. The basic idea here is analogous to that underlying the fixed-node approximation. In the latter, a trial function is used to approximate the nodes while diffusion Monte Carlo recovers the magnitude of the wavefunction. In the fixed phase approximation, the trial function is responsible for the phase and Monte Carlo produces the magnitude of the wavefunction.

## ACKNOWLEDGMENTS

[1] H.A. Kramers and G.H. Wannier, Phys. Rev. **60**, 252 (1941).

[2] M.P. Nightingale, in *Finite-Size Scaling and Simulation of Statistical Mechanical Systems,* edited by Privman, (World Scientific, Singapore 1990), p.287-351.

[3] M.P. Nightingale and C.J. Umrigar, in Recent Advances in Quantum Monte Carlo Methods, edited by W.A. Lester, Jr. (World Scientific, Singapore, 1997), p. 201. (See also http://xxx.lanl.gov/abs/chem-ph/9608001.)

[4] Andrei Mushinski and M. P. Nightingale, J. Chem. Phys. **101**, 8831 (1994).

[5] M. Meierovich, A. Mushinski, and M.P. Nightingale, J. Chem. Phys. **105**, 6498 (1996).

[6] M. Suzuki, Commun. math. Phys. **51**, 183 (1976).

[7] M. Suzuki, Prog. theor. Phys. **58**, 1377 (1977).

[8] M. P. Nightingale and H.W.J. Blöte, Phys. Rev. B **54**, 1001 (1996).

[9] M. P. Nightingale and H.W.J. Blöte, Phys. Rev. Lett. **76**, 4548 (1996).

[10] M. P. Nightingale and H.W.J. Blöte, Phys. Rev. Lett. **80**, 1007 (1998). Also see http://xxx.lanl.gov/abs/cond-mat/9708063

[11] C.P. Yang, Proc. Symp. Appl. Math. **15**, 351 (1963).

[12] J.H. Hetherington, Phys. Rev. A **30**, 2713 (1984).

[13] D.M. Ceperley and B. Bernu, J. Chem. Phys. **89**, 6316 (1988).

[14] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.M. Teller and E. Teller, J. Chem. Phys. **21**, 1087 (1953).

[15] P.J. Reynolds, R. N. Barnett, B. L. Hammond and W.A. Lester, Stat. Phys. **43**, 1017 (1986).

[16] C.J. Umrigar, *Int. J. Quant. Chem. Symp.* **23**, 217 (1989).

[17] see *e.g.* D. Ceperley and B. J. Alder, J. Chem. Phys. **81**, 5833 (1984); D. Arnow, M. H. Kalos, M. A. Lee and K. E. Schmidt, J. Chem. Phys. **77**, 5562, (1982); J. Carlson and M. H. Kalos, Phys. Rev. C **32**, 1735, (1985); Shiwei Zhang and M. H. Kalos, Phys. Rev. Lett. **67**, 3074 (1991); J. B. Anderson, in *Understanding Chemical Reactivity*, edited by S. R. Langhoff, Kluwer (1994); R. Bianchi, D. Bressanini, P. Cremaschi and G. Morosi, Chem. Phys. Lett., **184**, 343, (1991); V. Elser, Phys. Rev. A **34**, 2293 (1986); P. L. Silvestrelli, S. Baroni and R. Car, Phys. Rev. Lett. **71**, 1148 (1993)

[18] R. N. Barnett, P. J. Reynolds and W. A. Lester, J. Chem. Phys. **96**, 2141 (1992).

[19] B. Bernu, D.M. Ceperley, and W.A. Lester, Jr., J. Chem. Phys. **93**, 552 (1990).

[20] W.R. Brown, W.A. Glauser, and W.A. Lester, Jr., J. Chem. Phys. **103**, 9721 (1995).

[21] R.L. Coldwell, Int. J. Quant. Chem. Symp. **11**, 215 (1977).

[22] C.J. Umrigar, K.G. Wilson, and J.W. Wilkins, Phys. Rev. Lett. **60**, 1719 (1988); *Computer Simulation Studies in Condensed Matter Physics*, edited by D.P. Landau, K.K. Mon, and H.-B. Schüttler, Springer Proceedings in Physics Vol. 33 (Springer-Verlag, Berlin, 1988), p.185.

[23] M.P. Nightingale, Computer Simulation Studies in Condensed Matter Physics IX, edited by D. P. Landau, K.K. Mon and H. B. Schüttler, Springer Proc. Phys. 82 (Springer, Berlin, 1997).

[24] P.A. Whitlock and M. H. Kalos, J. Comp. Phys. **30,** 361 (1979).

[25] D.M. Ceperley, M.H. Kalos, in *Monte Carlo Methods in Statistical Physics*, ed. by K. Binder, Topics Current Phys. Vol.7 (Springer, Berlin, Heidelberg 1979) Chap.4.

[26] K.E. Schmidt and J.W. Moskowitz, J. Stat. Phys. **43**, 1027 (1986); J.W. Moskowitz and K.E. Schmidt, J. Chem. Phys. **85**, 2868 (1985).

[27] M. Kalos, D. Leveque and L. Verlet, Phys. Rev. **A9**, 2178 (1974).

[28] D. M. Ceperley, J. Comp. Phys. **51**, 404 (1983).

[29] J.H. Hetherington, Phys. Rev. A **30**, 2713 (1984).

[30] M.P. Nightingale and H.W.J. Blöte, Phys. Rev. B **33**, 659 (1986).

[31] C.J. Umrigar, M.P. Nightingale, and K.J. Runge, J. Chem. Phys. **99**, 2865 (1993).

[32] M.P. Nightingale and H.W.J. Blöte, Phys. Rev. Lett. **60**, 1662 (1988).

[33] K.J. Runge, Phys. Rev. B **45**, 12292 (1992).

[34] M.H. Kalos, J. Comput. Phys. **1**, 257 (1966); the original idea of "forward walking" predates this paper [M.H. Kalos (private communication)]. For further references see Ref. 11 of Ref. 35.

[35] K.J. Runge, Phys. Rev. B **45**, 7229 (1992).

[36] R. Grimm and R.G. Storer, J. Comp. Phys. **4**, 230 (1969); 7, 134, (1971); 9, 538, (1972).

[37] J.B. Anderson, J. Chem. Phys. **63**, 1499 (1975); J. Chem. Phys. **65**, 4121 (1976).

[38] D.M. Ceperley and B. J. Alder, Phys. Rev. Lett. **45**, 566 (1980).

[39] P.J. Reynolds, D.M. Ceperley, B.J. Alder and W.A. Lester, J. Chem. Phys. **77**, 5593 (1982).

[40] J.W. Moskowitz, K.E. Schmidt, M.A. Lee and M.H. Kalos, J. Chem. Phys. **77**, 349 (1982).

[41] Ceperley, D. M. and L. Mitas, in "New Methods in Computational Quantum Mechanics" Advances in Chemical Physics, XCIII, eds. I. Prigogine and S. A. Rice, 1996.

[42] James B. Anderson, Int. Rev. Phys. Chem., **14**, 85 (1995)

[43] B.L. Hammond, W.A. Lester and P.J. Reynolds, *Monte Carlo Methods in Ab Initio Quantum Chemistry*, (World Scientific 1994)

[44] T. Kato, Comm. Pure Appl. Math. **10** , 151 (1957).

[45] M.F. DePasquale, S.M. Rothstein and J. Vrbik, J. Chem. Phys. **89**, 3629 (1988).

[46] D.R. Garmer and J.B. Anderson, J. Chem. Phys. **89**, 3050 (1988).

[47] W.K. Hastings, Biometrika **57**, 97 (1970).

[48] D. Ceperley, G.V. Chester and M.H. Kalos, Phys. Rev. B **16**, 3081 (1977).

[49] M.H. Kalos and P.A. Whitlock, *Monte Carlo Methods,* Vol. 1, (Wiley, 1986).

[50] C.J. Umrigar, M.P. Nightingale and K.J. Runge, J. Chem. Phys. **99**, 2865 (1993).

[51] M. M. Hurley and P.A. Christiansen, J. Chem. Phys. **86**, 1069 (1987); P.A. Christiansen, J. Chem. Phys. **88**, 4867 (1988); P.A. Christiansen, J. Chem. Phys. **95**, 361 (1991).

[52] B. L. Hammond, P.J. Reynolds, and W.A. Lester, J. Chem. Phys. **87**, 1130 (1987).

[53] S. Fahy, X. W. Wang and Steven G. Louie, Phys. Rev. Lett. **61**, 1631 (1988); Phys. Rev. B **42**, 3503 (1990).

[54] H.-J. Flad, A. Savin and H. Preuss, J. Chem. Phys. **97**, 459 (1992).

[55] G.B. Bachelet, D.M. Ceperley and M.G.B. Chiocchetti, M.G.B. Phys. Rev. Lett. **62**, 2088 (1989). A. Bosin, V. Fiorentini, A. Lastri, and G.B. Bachelet, in *Materials Theory and Modeling Symposium*, edited by J. Broughton, P. Bristowe and J. Newsam, (Materials Research Society, Pittsburgh, 1993).

[56] L. Mitas, Phys. Rev. A 49, 4411 (1994); L. Mitas, in Electronic Properties of Solids Using Cluster

Methods, ed. T.A. Kaplan and S.D. Mahanti, Plenum, New York, (1994); J. C. Grossman and L. Mitas, Phys. Rev. Lett. **74**, 1323 (1995); J. C. Grossman, L. Mitas, and K. Raghavachari, Phys. Rev. Lett. **75**, 3870 (1995); J. C. Grossman and L. Mitas, Phys. Rev. B 52, 16735 (1995).

[57] P.A. Christiansen and L. A. Lajohn, Chem. Phys. Lett. **146**, 162 (1988); M. Menchi, A. Bosin, F. Meloni, G. B. Bachelet, in *Materials Theory and Modeling Symposium*, edited by J. Broughton, P. Bristowe and J. Newsam, (Materials Research Society, Pittsburgh, 1993).

[58] D.M. Ceperley, Rev. Mod Phys. **67**, 279 (1995); D.L. Freeman and J.D. Doll, Adv. Chem. Phys. **B70**, 139 (1988); J.D. Doll, D.L. Freeman and T.L. Beck, Adv. Chem. Phys., **78**, 61 (1990); B.J. Berne and D. Thirumalai, Ann. Rev. Phys. Chem. **37**, 401 (1986).

[59] J. Hirsch, Phys. Rev. B **28**, 4059 (1983); G. Sugiyama and S.E. Koonin, Ann. Phys. **168**, 1 (1986); Masuo Suzuki, J. Stat. Phys. **43**, 883 (1986); S. Sorella, S. Baroni, R. Car, and M. Parrinello, Europhys. Lett. **8**, 663 (1989); S. Sorella, E. Tosatti, S. Baroni, R. Car, and M. Parrinello, Int. J. Mod. Phys. **B1**, 993 (1989); R. Blankenbecler and R.L. Sugar, Phys. Rev. D **27,** 1304, (1983). S.R. White, D.J. Scalapino, R.L. Sugar, E.Y. Loh, Jr., J.E. Gubernatis, and R.T. Scalettar, Phys. Rev. **B40**, 506 (1989); N. Trivedi and D. M. Ceperley, Phys. Rev. B **41**, 4552 (1990); D. F. B. ten Haaf, J. M. van Bemmel, J. M. J. van Leeuwen and W. van Saarloos and D. M. Ceperley, Phys. Rev. B, **51**, 13039 (1995); A. Muramatsu, R. Preuss, W. von der Linden, P. Dietrich, F. F. Assaad and W. Hanke in *Computer Simulation Studies in Condensed Matter Physics*, edited by D.P. Landau, K.K. Mon, and H.-B. Schüttler, Springer Proceedings in Physics (Springer-Verlag, Berlin, 1994); H. Betsuyaku, in *Quantum Monte Carlo Methods in Equilibrium and Nonequilibrium Systems*, edited by M.Suzuki, Proceedings of the Ninth Taniguchi International Symposium, (Springer-Verlag; Berlin 1987) pgs. 50-61; M. Takahashi, Phys. Rev. Lett. **62**, 2313 (1989); Hans De Raedt and Ad Lagendijk, Phys. Reports **127**, 233 (1985); W. von der Linden, Phys. Reports **55**, 220, 53 (1992); Shiwei Zhang, J. Carlson and J. E. Gubernatis, Phys. Rev. Lett. **74**, 3652 (1995). See also Refs. 30 and 35.

[60] G. Ortiz, D. M. Ceperley, and R. M. Martin, Phys. Rev. Lett. **71**, 2777 (1993); G. Ortiz and D. M. Ceperley, Phys. Rev. Lett. **75**, 4642 (1995).

[61] M.P. Nightingale, Y. Ozeki, Y. Ye, unpublished.