

1993

A diffusion Monte Carlo algorithm with very small time-step errors

C. J. Umrigar

M. P. Nightingale

University of Rhode Island, nightingale@uri.edu

K. J. Runge

Follow this and additional works at: https://digitalcommons.uri.edu/phys_facpubs

Citation/Publisher Attribution

Umrigar, C. J., Nightingale, M. P., & Runge, K. J. (1993). A diffusion Monte Carlo algorithm with very small time-step errors. *The Journal of Chemical Physics*, 99(4), 2865-2890. doi: 10.1063/1.465195

Available at: <http://dx.doi.org/10.1063/1.465195>

This Article is brought to you for free and open access by the Physics at DigitalCommons@URI. It has been accepted for inclusion in Physics Faculty Publications by an authorized administrator of DigitalCommons@URI. For more information, please contact digitalcommons-group@uri.edu.

A diffusion Monte Carlo algorithm with very small time-step errors

Terms of Use

All rights reserved under copyright.

A diffusion Monte Carlo algorithm with very small time-step errors

C. J. Umrigar, M. P. Nightingale, and K. J. Runge

Citation: *The Journal of Chemical Physics* **99**, 2865 (1993); doi: 10.1063/1.465195

View online: <https://doi.org/10.1063/1.465195>

View Table of Contents: <http://aip.scitation.org/toc/jcp/99/4>

Published by the *American Institute of Physics*

Articles you may be interested in

[Fixed-node quantum Monte Carlo for molecules^{a\)} b\)](#)

The Journal of Chemical Physics **77**, 5593 (1982); 10.1063/1.443766

[Fermion Monte Carlo without fixed nodes: A game of life, death, and annihilation in Slater determinant space](#)

The Journal of Chemical Physics **131**, 054106 (2009); 10.1063/1.3193710

[Optimization of quantum Monte Carlo wave functions by energy minimization](#)

The Journal of Chemical Physics **126**, 084102 (2007); 10.1063/1.2437215

[Quantum chemistry by random walk. \$H^2P\$, \$H^+_3D_{3h}\$, \$^1A'_1\$, \$H_2^3\Sigma^+_u\$, \$H_4^1\Sigma^+_g\$, \$Be^1S\$](#)

The Journal of Chemical Physics **65**, 4121 (1976); 10.1063/1.432868

[A random-walk simulation of the Schrödinger equation: \$H^+_3\$](#)

The Journal of Chemical Physics **63**, 1499 (1975); 10.1063/1.431514

[Quantum Monte Carlo for molecules: Green's function and nodal release](#)

The Journal of Chemical Physics **81**, 5833 (1984); 10.1063/1.447637

PHYSICS TODAY

WHITEPAPERS

ADVANCED LIGHT CURE ADHESIVES

Take a closer look at what these environmentally friendly adhesive systems can do

READ NOW

PRESENTED BY
 **MASTERBOND**
ADHESIVES | SEALANTS | COATINGS

A diffusion Monte Carlo algorithm with very small time-step errors

C. J. Umrigar

Cornell Theory Center and Laboratory of Atomic and Solid State Physics, Cornell University, Ithaca, New York 14853

M. P. Nightingale

Department of Physics, University of Rhode Island, Kingston, Rhode Island 02881

K. J. Runge

Lawrence Livermore National Laboratory, University of California, Livermore, California 94551

(Received 25 January 1993; accepted 15 April 1993)

We propose modifications to the simple diffusion Monte Carlo algorithm that greatly reduce the time-step error. The improved algorithm has a time-step error smaller by a factor of 70 to 300 in the energy of Be, Li₂ and Ne. For other observables the improvement is yet larger. The effective time step possible with the improved algorithm is typically a factor of a few hundred larger than the time step used in domain Green function Monte Carlo. We also present an optimized 109 parameter trial wave function for Be which, used in combination with our algorithm, yields an exceedingly accurate ground state energy. A simple solution to the population control bias in diffusion Monte Carlo is also discussed.

I. INTRODUCTION

Ground state energies of fermion systems such as atoms and molecules can be computed within the fixed-node approximation¹⁻⁵ by quantum Monte Carlo methods. These methods can also be employed to compute other quantities and apply to a wide variety of systems. Mostly two basic algorithms have been used: the Green function Monte Carlo method developed by Kalos and co-workers,^{6,7} and extended by Ceperley and co-workers,^{8,9} and the diffusion Monte Carlo method,^{2,3,10,11} also sometimes referred to as the short-time approximation to Green function Monte Carlo.

In its implementation the diffusion Monte Carlo method is simpler than Green function Monte Carlo,^{4,6,7} but it suffers from a systematic bias, the so-called time-step error. That is, both Green function Monte Carlo and diffusion Monte Carlo calculations start from an optimized trial wave function for the ground state and improve upon variational Monte Carlo by a stochastic implementation of the power method, a standard algorithm for computing the dominant eigenvalue and eigenvector of an operator. In Green function Monte Carlo the ground state is obtained by iteratively acting on an initial wave function, i.e., the optimized trial wave function ψ_T , with an operator proportional to $(\mathcal{H} + E_0)^{-1}$, where \mathcal{H} is the Hamiltonian and E_0 is a shift that renders the operator positive definite. In diffusion Monte Carlo one uses an operator proportional to $e^{-\tau\mathcal{H}}$, defined in terms of a time step τ . In the previous expression and throughout the rest of this paper we use atomic units.

In Green function Monte Carlo the stochastic multiplication by $(\mathcal{H} + E_0)^{-1}$ (see Ref. 4) is exact, but in the case of diffusion Monte Carlo an approximation to the operator $e^{-\tau\mathcal{H}}$ is used. This involves an expression for the coordinate space representation of the diffusion Monte

Carlo operator which is correct only to some order of the time step τ in the limit $\tau \rightarrow 0$. In practice, the calculations are therefore performed for several values of τ and the results are extrapolated to give the $\tau \rightarrow 0$ limit. These values of τ have to be small enough to permit an accurate extrapolation, but configurations realized in successive iterations of a Monte Carlo run become more correlated as τ is reduced, which increases statistical errors for a run of a given number of iterations, i.e., a given amount of computer time. Hence, the values of τ used in a practical computation represent a compromise between the systematic time-step error and the statistical error.

Because of this trade-off, several attempts have been made to design quadratic algorithms, which have a time-step error in the energy of $\mathcal{O}(\tau^2)$, rather than $\mathcal{O}(\tau)$.¹²⁻¹⁶ However, in practice these algorithms are not second order, except possibly in the special case of nodeless two-electron wave functions,¹⁶ because the local energy and the velocity which play an important role in the diffusion Monte Carlo algorithm are not uniformly well-behaved in configuration space; in particular they diverge at the nodes. In this paper we propose, instead, an algorithm with very small time-step errors to all orders in τ . For the range of time steps of interest the error is far smaller than that of the "quadratic" algorithms, even in the case of nodeless wave functions. Although we have no *a priori* mathematical reason to believe that our algorithm has strictly quadratic errors, they appear to be quadratic to within the statistical accuracy for the systems we have investigated. Some of the preliminary results of this work, obtained with an earlier version of the algorithm presented here, were reported previously.¹⁷

As mentioned, the Green function Monte Carlo method does not have a time-step error, but the effective time steps used in Green function Monte Carlo are considerably smaller than the ones commonly used in diffusion

Monte Carlo. Moreover, Green function Monte Carlo is more complex algorithmically than diffusion Monte Carlo. Consequently, Green function Monte Carlo is computationally much more expensive, even if diffusion Monte Carlo requires calculations for a few different values of τ for accurate $\tau \rightarrow 0$ extrapolation.

It should be noted that the approach discussed in this paper, apart from having a time-step bias, is approximate in principle even in the $\tau \rightarrow 0$ limit because it uses the fixed-node approximation to yield antisymmetric wave functions, as required by the fermionic nature of electrons. For completeness we mention that there are algorithms that yield the exact ground state without making the fixed-node approximation.^{9,18,19} However, the statistical accuracy one can obtain with these algorithms is limited.

The layout of this paper is as follows. In Section II A we review the basics of diffusion Monte Carlo and introduce what we call the *simple algorithm*. This algorithm is typical of one class of algorithms used in the literature^{2,20,21} and will serve to provide a benchmark to evaluate the performance of the improved algorithm, introduced in Sections III A and III C. Those sections contain a systematic discussion of two types of contributions to the time-step error. These are, firstly, the failure of the simple algorithm to sample the exact probability distribution (the square of the ground state wave function) even in the ideal limit in which the trial function is the exact ground state wave function. In other words, this algorithm can, for sufficiently large time steps, yield a distribution and expectation values that are even inferior to those obtained from variational Monte Carlo. The solution to this problem is the inclusion of an accept/reject step as was proposed by Reynolds *et al.*³ This is discussed in Section III A.

A problem caused by the introduction of the accept/reject step is that it can give rise to unphysical persistent configurations, a phenomenon discussed in Section III B. This problem can be alleviated by addressing the second source of large time-step errors, *viz.* Monte Carlo instabilities resulting from nonanalyticity of the velocity and the local energy in the transformed Schrödinger equation governing the diffusion Monte Carlo process. The related improvements of the algorithm are discussed in Section III C. Section II B contains a discussion of more technical aspects of the diffusion Monte Carlo algorithm, such as the estimators employed and systematic errors due to population control, which is an essential part of the algorithm.

In Section IV A we compare the time-step errors of the simple and improved algorithms, for the energy, kinetic energy and moments of the charge distribution of Be, Li₂, and Ne, using simple trial wave functions. These time-step errors are compared to those in the literature in Section IV B. In Section IV C we present the time-step error in the energy of Be using a good wave function.

Finally, in Appendix A we present a summary of the algorithm and in Appendix B we present the form of the trial wave functions and their parameters. These two appendices are provided in order to encourage others to program and test the algorithm for themselves. In Appendix C we present an alternative to the generalized Metropolis

algorithm and in Appendix D we estimate how the errors in the simple diffusion Monte Carlo algorithm scale with the time step.

II. REVIEW OF DIFFUSION MONTE CARLO

A. The simple algorithm

A transformed version of the Schrödinger equation in imaginary time serves as the starting point of the diffusion Monte Carlo method to compute ground state properties of fermion systems within the fixed-node approximation. In principle, the discussion below is applicable also to excited states, except for a minor complication due to the existence of inequivalent nodal pockets for excited state wave functions. Here we will limit ourselves to ground states.

We shall consider systems with n electrons, the spatial coordinates of which are given by a $3n$ -dimensional vector \mathbf{R} . It will be convenient to introduce vectors specifying coordinates of one electron only. For example, we write $\mathbf{R} = (\mathbf{r}_1, \dots, \mathbf{r}_n)$, where \mathbf{r}_i is a three-dimensional vector specifying the coordinates of electron number i . More generally, we shall use upper case symbols to denote n -electron vectors and lower case symbols for the corresponding single electron vectors.

Given a *guiding function* $\psi_G(\mathbf{R})$ one can introduce a distribution $f(\mathbf{R}, t) = \psi_G(\mathbf{R})\psi(\mathbf{R}, t)$ which, if $\psi(\mathbf{R}, t)$ satisfies the Schrödinger equation, can be shown to be a solution of the equation^{2,3}

$$\begin{aligned} \psi_G(\mathbf{R})(\mathcal{H} - E_T)\psi_G(\mathbf{R})^{-1}f(\mathbf{R}, t) \\ = -\frac{1}{2}\nabla^2 f(\mathbf{R}, t) + \nabla \cdot [\mathbf{V}(\mathbf{R})f(\mathbf{R}, t)] - S(\mathbf{R})f(\mathbf{R}, t) \\ = -\frac{\partial f(\mathbf{R}, t)}{\partial t}. \end{aligned} \quad (1)$$

Here the *velocity* \mathbf{V} —we deviate from the usage of referring to \mathbf{V} as the (quantum) force—is given by

$$\mathbf{V}(\mathbf{R}) = (\mathbf{v}_1, \dots, \mathbf{v}_n) = \frac{\nabla \psi_G(\mathbf{R})}{\psi_G(\mathbf{R})} = \frac{(\partial_1, \dots, \partial_n)\psi_G(\mathbf{R})}{\psi_G(\mathbf{R})} \quad (2)$$

and the coefficient of the *branching* term is

$$S(\mathbf{R}) = E_T - E_L(\mathbf{R}), \quad (3)$$

which is defined in terms of the *local energy*

$$E_L(\mathbf{R}) = \frac{\mathcal{H}\psi_G(\mathbf{R})}{\psi_G(\mathbf{R})} = -\frac{\nabla^2 \psi_G(\mathbf{R})}{2\psi_G(\mathbf{R})} + \mathcal{V}(\mathbf{R}), \quad (4)$$

where $\mathcal{V}(\mathbf{R})$ is the potential energy. E_T is a shift in energy such that $E_0 - E_T \approx 0$, where E_0 is the ground state energy. In the special case $\psi_G = 1$ the velocity vanishes and the local energy equals the potential energy, so that Eq. (1) reduces to the standard Schrödinger equation in imaginary time.

The modified Schrödinger equation (1) is equivalent to the standard one only for wave functions that vanish at the nodes of $\psi_G(\mathbf{R})$. Thus a fermionic state may be treated by prescribing a nodal surface on which the Green function and $\psi_G(\mathbf{R})$ vanish. This surface divides space into positive and negative regions that obey exchange antisymmetry. As

is usual, $\psi_G(\mathbf{R})$ is chosen to be some optimized antisymmetric trial wave function $\psi_T(\mathbf{R})$ within the nodal pocket under consideration, and we will not distinguish between $\psi_G(\mathbf{R})$ and $\psi_T(\mathbf{R})$ in the rest of this paper.

For long times the distribution $f(\mathbf{R}, t)$ approaches $\psi_T(\mathbf{R})\psi_0(\mathbf{R})$ up to a normalization constant, where, in each nodal region, $\psi_0(\mathbf{R})$ is the ground state eigenfunction consistent with the boundary condition that it vanish at the nodes of $\psi_T(\mathbf{R})$. This implies that the long-time limit is the true fermion ground state if and only if the nodes of ψ_T happen to be the exact nodes of the ground state wave function. In general there is no procedure for choosing a $\psi_T(\mathbf{R})$ that has the exact nodes. The *fixed-node energy* obtained by extrapolation to $\tau=0$ is an upper bound to the exact fermion ground state energy.

An integral equation equivalent to Eq. (1) is

$$f(\mathbf{R}', t+\tau) = e^{\tau E_T(t+\tau)} \int d\mathbf{R} \tilde{G}(\mathbf{R}', \mathbf{R}, \tau) f(\mathbf{R}, t), \quad (5)$$

where \tilde{G} is the Green function for the case $E_T = 0$. The power method consists of iterating this equation and this is done in an average sense as described below. The energy shift $E_T(t + \tau)$ plays the role of an arbitrary time-dependent normalization constant,²² chosen in such a way that the electron density f remains finite and nonvanishing in the limit $t \rightarrow \infty$. Note that by convention, E_T on the right and the electron distribution f on the left-hand side of Eq. (5) depend on the same parameter, $t + \tau$. In the Monte Carlo iteration of Eq. (5) $E_T(t + \tau)$ depends on the specific random realization of $f(\mathbf{R}, t)$ but not that of $f(\mathbf{R}', t + \tau)$.

The three terms on the left-hand side of Eq. (1) describe diffusion, drift and growth or decay. An approximate Green function, with an error of $\mathcal{O}(\tau^2)$ for small τ , is given by the product of the individual diffusion, drift and growth/decay Green functions,

$$\begin{aligned} \tilde{G}(\mathbf{R}', \mathbf{R}, \tau) = & \frac{1}{(2\pi\tau)^{3n/2}} \int d\mathbf{R}'' e^{-[(\mathbf{R}' - \mathbf{R}'')^2/2\tau]} \\ & \times \delta(\mathbf{R}'' - \mathbf{R} - \mathbf{V}(\mathbf{R})\tau) \\ & \times e^{-(1/2)[E_L(\mathbf{R}') + E_L(\mathbf{R})]\tau + \mathcal{O}(\tau^2)}. \end{aligned} \quad (6)$$

To compute the ground state energy and other expectation values, the n -electron distribution $f(\mathbf{R}, t)$ is represented in diffusion Monte Carlo by an average over a time series of generations of walkers each of which consists of a fluctuating number of N walkers. To be precise, a walker is a pair $(\mathbf{R}_\alpha, w_\alpha)$, $\alpha = 1, \dots, N$, with \mathbf{R}_α a $3n$ -dimensional electron configuration with statistical weight w_α .²³ At time t , the walkers represent a random realization of the n -electron distribution, *viz.*

$$f(\mathbf{R}, t) = \sum_{\alpha=1}^N w_\alpha \delta(\mathbf{R} - \mathbf{R}_\alpha). \quad (7)$$

Each walker executes a branching random walk constructed so as to satisfy on average the power method iteration relation, Eq. (5), where the average is over all realizations of walkers at time $t + \tau$ given a fixed, yet arbitrary

realization of walkers at time t . This means that we are iterating Eq. (5) with $f(\mathbf{R}, t)$ on the right-hand side of the form of Eq. (7). The exact equation does not conserve this form, but its stochastic implementation does.

In the first step, a given walker (\mathbf{R}, w) drifts to $\mathbf{R}' = \mathbf{R} + \mathbf{V}(\mathbf{R})\tau$, according to the δ function in Eq. (6). Then in the second step, the walker diffuses to \mathbf{R}'' , where \mathbf{R}'' is sampled from the diffusion Green function $(2\pi\tau)^{-3n/2} \exp\{-(\mathbf{R}' - \mathbf{R}'')^2/2\tau\}$. Finally, the third step, resulting from the growth/decay (or branching) term of Eq. (1), is to multiply the weight of the walker by the remaining exponential factors in Eqs. (5) and (6), i.e., w is replaced by $w\Delta w$, with

$$\Delta w = \exp\left\{\frac{1}{2}[S(\mathbf{R}', t+\tau) + S(\mathbf{R}, t+\tau)]\right\}. \quad (8)$$

Note that the position dependence in this expression comes from the local energy E_L , and the time dependence from the trial energy E_T . We shall usually suppress explicit reference to the time dependence.

For reasons of efficiency, walkers with big statistical weights are duplicated, while walkers with low weights are combined pairwise. That is, in the duplication an old walker is replaced by two identical new walkers, each with weight equal to half of the old weight. In combining two walkers $(\mathbf{R}_{\alpha_1}, w_{\alpha_1})$ and $(\mathbf{R}_{\alpha_2}, w_{\alpha_2})$ they are replaced by one walker $(\mathbf{R}_\alpha, w_\alpha)$ with weight $w_\alpha = w_{\alpha_1} + w_{\alpha_2}$ and configuration $\mathbf{R}_\alpha = \mathbf{R}_{\alpha_i}$ sampled with probability w_{α_i}/w_α ($i=1$ or $i=2$). We note that in both cases the total weight is kept constant,²³ while the more frequently used alternative approach^{4,5} of creating $\text{int}(w_\alpha + \xi)$ walkers of unit weight, where ξ is a random number uniformly distributed on $[0, 1]$, leads to an unnecessary random fluctuation of the total weight of the walkers or equivalently the number of walkers.

For the time being we shall assume that the equilibrium distribution of the Monte Carlo process as described above is $\psi_T(\mathbf{R})\psi_0(\mathbf{R})$. This is only approximately correct, because of population control exercised by having a time dependent $E_T(t + \tau)$ in Eq. (5), as discussed in more detail in Section II B.

This simple diffusion Monte Carlo algorithm is not viable for a wave function with nodes for the following two reasons. Firstly, in the vicinity of the nodes the local energy of the trial function ψ_T diverges inversely proportional to the distance to the nodal surface. For nonzero τ , 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 τ . 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 commonly remedied by introducing cut-offs

in the values of the local energy and the velocity.^{15,21} For example, DePasquale, Rothstein, and Vrbik¹⁵ use the cut-offs

$$E_L(\mathbf{R}) \rightarrow E_{\text{var}} + \frac{2}{\sqrt{\tau}} \text{sgn}\{E_L(\mathbf{R}) - E_{\text{var}}\} \quad \text{for } |E_L(\mathbf{R}) - E_{\text{var}}| > \frac{2}{\sqrt{\tau}}, \quad (9)$$

$$V_i(\mathbf{R}) \rightarrow \frac{1}{\tau} \text{sgn}\{V_i(\mathbf{R})\} \quad \text{for } |V_i(\mathbf{R})| > 1/\tau, \quad (10)$$

where E_{var} is the variational energy associated with ψ_T , and where V_i is component i of the velocity (see Ref. 24). Since the cut-offs have no effect in the $\tau \rightarrow 0$ limit, the results extrapolated to $\tau = 0$ are correct.

We will refer to this algorithm as the *simple diffusion Monte Carlo algorithm*²⁴ and will contrast it with the *improved diffusion Monte Carlo algorithm* discussed below. There is one more feature of the simple algorithm that should be mentioned. For finite τ a walker can cross a node and move into a region where ψ_T is negative. If this happens the walker is killed. This is the source of large time-step errors, especially in the *growth estimator* of the energy, as discussed later.

B. Expectation values

In this section we define the *mixed* and the *growth* estimators of the energy. We show that the distribution of walkers suffers from a population control bias and we present a method for projecting away this bias without having to perform calculations for more than one population size.^{23,25}

The basic power method iteration relation, Eq. (5) has a normalization constant, which contains a running estimate of the ground state energy. In practice, one also has to include a term that insures that the total weight of all walkers remains approximately stationary by correcting for random drifts in the average. In actual calculations, time t is a discrete variable and it is convenient to introduce an integral valued variable $\hat{t} = t/\tau$, but we do not introduce a different notation for functions of t and \hat{t} . We choose

$$E_T(\hat{t}+1) = E_{\text{est}}(\hat{t}) - \frac{1}{g\tau} \log \frac{W_{\hat{t}}}{W_0}. \quad (11)$$

Here $E_{\text{est}}(\hat{t})$ is an estimate of the energy at time \hat{t} , which we have chosen it to be the current cumulative value of the mixed estimator [see Eq. (19)], but other choices are equally valid. This term serves to suppress an exponential population explosion (or implosion) as would occur for a constant $E_T > E_0$ (or $E_T < E_0$). The second term redresses fluctuations and attempts to reset some g generations later the total weight of the population, $W_{\hat{t}}$, to a target, W_0 . In the calculations reported in this paper we set $g=1/\tau$, which is of the order of the correlation time of $e^{-\tau\mathcal{H}}$.

Even though each generation of walkers on the average evolves by construction from the previous one according to

Eq. (5), the feedback of the number of walkers into $E_T(\hat{t})$ produces a systematic bias, the population control bias. To understand that the equilibrium distribution of the stochastic process is only approximately equal to the ground state distribution $\psi_T\psi_0$, consider the branching term in Eq. (1) with $E_T = E_0$. If a fluctuation increases the fraction of walkers in a region where $E_L(\mathbf{R}) - E_0 < 0$ the population size increases. The second term in Eq. (11) moderates this trend, decreasing the equilibrium distribution relative to $\psi_T(\mathbf{R})\psi_0(\mathbf{R})$. In other words, the diffusion Monte Carlo equilibrium distribution $f(\mathbf{R})$ is too small for low $E_L(\mathbf{R})$. Similarly, $f(\mathbf{R})$ is too large in regions of high $E_L(\mathbf{R})$. Both effects increase the time average of the energy. This reasoning suggests that the expectation value of the energy will be biased by an amount proportional to the covariance of the fluctuations in $\exp(E_T(\hat{t})\tau)$ and the fluctuations in the average energy of a generation of walkers. Since the former fluctuation is also proportional to the fluctuations in the average energy of a generation of walkers and since both have fluctuations proportional to $1/\sqrt{W_0}$, we expect a bias linear in $1/W_0$, i.e., proportional to the inverse of the average population size. In order to reduce the covariance, it is desirable to make g , in Eq. (11), greater than the auto correlation time of the local energy. However, a large value of g results in large fluctuation in the population size, so there is a trade-off. It is possible to eliminate the population control error by performing calculations for different population sizes W_0 and extrapolating to infinite W_0 . However, a better method exists^{23,25} for determining the population control error, one that does not require calculations for different sized populations, as we discuss next.

Repeated operation by \tilde{G} on an arbitrary distribution f results in an equilibrium distribution proportional to $\psi_T(\mathbf{R})\psi_0(\mathbf{R})$. However, in order to maintain an approximately constant population of walkers, the operator used in the Monte Carlo process is \tilde{G} multiplied by a time-dependent renormalization factor $e^{\tau E_T(\hat{t})}$. In order to recover a distribution proportional to $\psi_T(\mathbf{R})\psi_0(\mathbf{R})$ it is necessary to undo the effect of the time-dependent normalization factors. This is done by a reweighting of the averages. A complete undoing of the normalization would involve products of an ever-increasing number of factors which would introduce fluctuations of ever-increasing magnitude. Fortunately, the number of factors used for reweighting can be kept rather small in practice. The number required to reduce the population-control error to a given value is proportional to the autocorrelation time (in units of τ) of the energy, with a proportionality constant that decreases with increasing population size and increases with increasing rms fluctuation of the local energy. From a more formal point of view, the diffusion Monte Carlo process has been constructed so that

$$\overline{e^{-\tau E_T(\hat{t})} |f(\hat{t})\rangle} = \tilde{G} |f(\hat{t}-1)\rangle, \quad (12)$$

where the bar on the left indicates an average in which the distribution corresponding to $|f(\hat{t}-1)\rangle$ is kept fixed in a state of the form given in Eq. (7). That is, if $|f(\hat{t}-1)\rangle$ is any sum of δ -functions in configuration space, then the

sum of δ -functions resulting from one step of the algorithm has expectation value $\bar{G}|f(\hat{i}-1)\rangle/e^{-\tau E_T(\hat{i})}$. By iteration one obtains

$$\overline{\Pi(\hat{i}, T_p)}|f(\hat{i})\rangle = \tilde{G}_p|f(\hat{i}-T_p)\rangle, \quad (13)$$

where

$$\Pi(\hat{i}, T_p) = \prod_{m=0}^{\tilde{T}_p-1} e^{-\tau E_T(\hat{i}-m)}, \quad (14)$$

with $\Pi(\hat{i}, 0) = 1$, $\Pi(0, T_p) = 1$, where $\tilde{T}_p = \min(\hat{i}, T_p)$. (Note that $T_p = \tilde{T}_p$ except at the start of the equilibration Monte Carlo moves.) In practice, in order to avoid overflow/underflow errors on the computer, most of the constant part of $e^{-\tau E_T(\hat{i})}$ is absorbed into the definition of \tilde{G} , so that the remaining part in Eq. (14) fluctuates about a value close to one.

It follows from Eq. (13) that one can extract the following estimator of the equilibrium distribution $\psi_T\psi_0$ of the transformed Schrödinger, Eq. (1), from a Monte Carlo run with realizations of the electron distribution $f(\mathbf{R}, \hat{i})$ as defined in Eq. (7):

$$\psi_T(\mathbf{R})\psi_0(\mathbf{R}) \approx \frac{1}{T} \sum_{i=1}^T \Pi(\hat{i}, T_p) f(\mathbf{R}, \hat{i}). \quad (15)$$

In principle, this estimator is unbiased only for infinite T_p but, its variance increases with increasing T_p . The energy autocorrelation time sets the scale for the values of T_p required to project out the population control bias. In practice, for systems we studied, this time is small enough that the increase in the variance is negligible. Almost all the calculations presented in the literature have been performed omitting the factor $\Pi(\hat{i}, T_p)$ above. This is justified in most cases, since the population control errors tend to be smaller than the statistical errors for sufficiently large sized populations. However, there is the risk that this may not be true at times, especially when highly accurate results are required. The above expression and the explicit estimators given below allow one to estimate the bias due to population control and correct for it at virtually no computational cost without performing several runs with increasing numbers of walkers.

Expression (15) is of practical use only upon integration over \mathbf{R} , i.e., for the computation of expectation values of observables. For example, one can estimate mixed expectation values of the form

$$\frac{\langle \psi_T | \mathcal{A} | \psi_0 \rangle}{\langle \psi_T | \psi_0 \rangle} = \frac{\int d\mathbf{R} \psi_T(\mathbf{R}) \mathcal{A}(\mathbf{R}) \psi_0(\mathbf{R})}{\int d\mathbf{R} \psi_T(\mathbf{R}) \psi_0(\mathbf{R})}, \quad (16)$$

where \mathcal{A} is an arbitrary operator diagonal in the position representation. Substitution of Eq. (15) and Eq. (7) yields

$$\frac{\langle \psi_T | \mathcal{A} | \psi_0 \rangle}{\langle \psi_T | \psi_0 \rangle} \approx \frac{\sum_{i=1}^T \Pi(\hat{i}, T_p) \sum_{\alpha=1}^{N(\hat{i})} w_{\alpha}(\hat{i}) \mathcal{A}\{\mathbf{R}_{\alpha}(\hat{i})\}}{\sum_{i=1}^T \Pi(\hat{i}, T_p) W(\hat{i})}, \quad (17)$$

where $W(\hat{i})$ denotes the total weight of all $N(\hat{i})$ walkers of generation \hat{i} . We note that if \mathcal{A} commutes with \mathcal{H}

$$\frac{\langle \psi_T | \mathcal{A} | \psi_0 \rangle}{\langle \psi_T | \psi_0 \rangle} = \frac{\langle \psi_0 | \mathcal{A} | \psi_0 \rangle}{\langle \psi_0 | \psi_0 \rangle}, \quad (18)$$

i.e., the mixed expectation value equals the true quantum mechanical expectation value. For other operators, if reasonably accurate trial functions are used, the mixed expectation value closely approximates the true expectation value and is likely to have a comparable time-step bias, which is all that counts for the purpose of this paper.

We have used two different estimators for the ground state energy E_0 and a single one for the expectation values of other operators.⁴ The *mixed estimator* estimator for E_0 is an immediate application of Eq. (17) with $\mathcal{A} = \mathcal{H}\psi_T/\psi_T$. It is defined as

$$E_0 \approx E_{\text{mix}}(T, T_p) = \frac{\sum_{i=1}^T \Pi(\hat{i}, T_p) \sum_{\alpha=1}^{N(\hat{i})} w_{\alpha}(\hat{i}) \frac{\mathcal{H}\psi_T\{\mathbf{R}_{\alpha}(\hat{i})\}}{\psi_T\{\mathbf{R}_{\alpha}(\hat{i})\}}}{\sum_{i=1}^T \Pi(\hat{i}, T_p) W(\hat{i})}. \quad (19)$$

Another useful estimator is the *growth estimator*, which reads

$$E_0 \approx E_{\text{gr}}(T, T_p) = -\frac{1}{\tau} \log \frac{\sum_{i=1}^T \Pi(\hat{i}+1, T_p+1) W(\hat{i}+1)}{\sum_{i=1}^T \Pi(\hat{i}, T_p) W(\hat{i})}. \quad (20)$$

In this expression the argument of the logarithm is an estimator of $\langle \psi_T | G(\tau) | \psi_0 \rangle / \langle \psi_T | \psi_0 \rangle$.

It should be noted that the statistical error of the growth estimator calculated with the above method is no larger than that of the mixed estimator, as will be apparent from the results. An alternative method,^{4,5} mentioned above, is to construct $\text{int}(w_{\alpha} + \xi)$ unit-weight walkers from each of the original walkers labeled by α . This operation does not conserve the total weight of all walkers and hence if the integral weights obtained after this operation are used in the numerator of Eq. (20), additional large fluctuations are introduced in the growth estimator, which are not present in the mixed estimator. These unnecessary fluctuations can be avoided by simply choosing $W(\hat{i}+1)$ in the numerator of Eq. (20) to be the sum of weights prior to performing this operation.

The method described in this section for evaluating the population control error can be extended to computations on parallel computers to yield unbiased expectation values while at the same time performing load-balanced computations.

III. IMPROVED DIFFUSION MONTE CARLO ALGORITHM

A. The limit of perfect importance sampling

In the limit of perfect importance sampling, that is if $\psi_T(\mathbf{R}) = \psi_0(\mathbf{R})$, the energy shift E_T can be chosen such that the branching term in Eq. (1) vanishes identically for all \mathbf{R} . It is interesting to note that, in that case, the energy estimators have zero variance, but the important observation is that the branching term can be considered small for good wave functions. Yet, the simple diffusion Monte

Carlo algorithm discussed in Section II A yields an equilibrium distribution of walkers with a time-step error even in the ideal limit. If for the moment we ignore the branching term in Eq. (1), then we have the equation

$$-\frac{1}{2}\nabla^2 f + \nabla \cdot (\mathbf{V}f) = -\frac{\partial f}{\partial t} \quad (21)$$

This equation has the steady-state solution $f = |\psi_T|^2$ for any ψ_T . In the limit of perfect importance sampling this is the desired distribution, but for any finite time step τ it is not the equilibrium distribution of the drift-diffusion Green function, i.e., the Green function in Eq. (6) without the branching factor. Following Reynolds *et al.*,³ we can sample $|\psi_T|^2$ with no time-step error by using a generalized²⁶⁻²⁸ Metropolis algorithm.²⁹ The approximate drift-diffusion Green function is used to propose moves, which are then accepted with probability

$$p = \min\left(\frac{|\psi_T(\mathbf{R}')|^2 \tilde{G}(\mathbf{R}, \mathbf{R}', \tau)}{|\psi_T(\mathbf{R})|^2 \tilde{G}(\mathbf{R}', \mathbf{R}, \tau)}, 1\right) \equiv 1 - q, \quad (22)$$

as prescribed by the detailed balance condition.

Within the fixed-node approximation, the number of walkers that move across nodal surfaces of the trial wave function vanishes as $\tau \rightarrow 0$. The fixed-node method can be implemented with the requirement that $G(\mathbf{R}', \mathbf{R}, \tau)$ vanish if \mathbf{R}' and \mathbf{R} are in different nodal pockets for all τ . Yet, for any finite τ , moves across the nodes will be proposed because of the nature of the approximation of the Green function. Hence we always reject moves that attempt to cross nodes, even though it is possible to satisfy the detailed balance condition without so doing.

It is worth noting that the common practice of killing walkers that stray across nodes^{1-3,5,30-32} results in a large time-step error, particularly in the growth estimator. In the $\psi_T = \psi_0$ limit, the true Green function is normalized to one, i.e., $\int d\mathbf{R}' G(\mathbf{R}', \mathbf{R}, \tau) = 1$. On the other hand, if walkers that cross nodes are killed then the approximate $G(\mathbf{R}', \mathbf{R}, \tau)$ has a normalization smaller than one, and consequently the growth estimator, which is directly related to the norm, fails to satisfy a zero-variance principle. Even some algorithms that contain an accept/reject step kill walkers that cross nodes.^{3,5,31} In this case they fail to sample $|\psi_0|^2$ in the ideal $\psi_T = \psi_0$ limit since detailed balance is violated. Unlike the improved algorithm, the simple one has a growth estimator for the energy which always has nonzero variance, since the algorithm fails to conserve probability even in the $\psi_T = \psi_0$ limit.

It is shown in Appendix D that killing walkers that cross nodes results in a $\sqrt{\tau}$ dependence of the growth estimator. This makes accurate extrapolation to $\tau=0$ exceedingly difficult since $\sqrt{\tau}$ has infinite slope.

If we stopped here we would have an exact and efficient variational Monte Carlo algorithm to sample from $|\psi_T|^2$. Now, we reintroduce the branching term to convert the steady-state distribution from $|\psi_T|^2$ to $\psi_T \psi_0$. This is accomplished by reweighting the walkers with the branching factor [see Eq. (6)]

Δw

$$= \begin{cases} \exp\{\frac{1}{2}[S(\mathbf{R}') + S(\mathbf{R})]\tau_{\text{eff}}\} & \text{for an accepted move,} \\ \exp[S(\mathbf{R})\tau_{\text{eff}}] & \text{for a rejected move,} \end{cases} \quad (23)$$

where S is defined in Eq. (3). Following Reynolds *et al.*,³ an effective time step τ_{eff} was introduced to account for the changed rate of diffusion. We set

$$\tau_{\text{eff}} = \tau \frac{\langle p \Delta R^2 \rangle}{\langle \Delta R^2 \rangle}, \quad (24)$$

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

An alternative to expression (23) is obtained by replacing the two reweighting factors by a single, average expression, where the average is over accepting and rejecting the proposed move with the appropriate weights. Subsequent reweighting factors contribute multiplicatively and thus it is natural to use the expression obtained by averaging the exponent, which of course gives the same result as averaging the exponential for $\tau \rightarrow 0$. This yields the reweighting factor

$$\Delta w = \exp\left[\left\{\frac{p}{2}(S(\mathbf{R}') + S(\mathbf{R})) + qS(\mathbf{R})\right\}\tau_{\text{eff}}\right] \quad (25)$$

for all moves.

In our computations this expression was found to yield somewhat smaller fluctuations and time-step error than expression (23).

An analogous modification is made in the computation of mixed estimators. Equation (17) is modified to read

$$\frac{\langle \psi_T | \mathcal{A} | \psi_0 \rangle}{\langle \psi_T | \psi_0 \rangle} \approx \frac{\sum_{t=1}^T \Pi(\hat{t}, T_p) \sum_{\alpha=1}^{N(\hat{t})} w_{\alpha}(\hat{t}) (q \mathcal{A}\{\mathbf{R}_{\alpha}(\hat{t}-1)\} + p \mathcal{A}\{\mathbf{R}'_{\alpha}(\hat{t})\})}{\sum_{t=1}^T \Pi(\hat{t}, T_p) W(\hat{t})}, \quad (26)$$

where $\mathbf{R}'_a(\hat{t})$ is the proposed move at time \hat{t} which has probability p of being accepted. This is motivated by the fact that in a variational Monte Carlo calculation [i.e., all weights and $\Pi(\hat{t}, T_p)$ set to one], the above expression yields the same expectation values as Eq. (17), but with a somewhat smaller variance.

A requirement to maintain a correct algorithm is that the ratio $\tau_{\text{eff}}/\tau \rightarrow 1$ for $\tau \rightarrow 0$. This is fulfilled since the acceptance defined in Eq. (22) approaches 1 in this limit. A stronger requirement is that inclusion of the accept/reject step does not alter the first two moments of the effective Green function to leading order in τ . This condition is met since the acceptance p , defined in Eq. (22) can be shown, by Taylor expansion, to be $1 + \mathcal{O}(\tau)$. Hence, τ_{eff} can be expanded as $\tau_{\text{eff}} = \tau(1 + c_1\tau + c_2\tau^{3/2} + \dots)$. In principle, there always exists a "true" value of τ_{eff} in the range between 0 and τ that yields the true $\tau \rightarrow 0$ extrapolated value of the fixed-node energy. Empirically we find that the time-step error in the energy is nearly quadratic. Since the order of the error in the energy is one lower than in the Green function, this implies that the procedure outlined above for determining τ_{eff} gets values of c_1 and c_2 that are very close to the "true" values.

An estimate of τ_{eff} is readily obtained iteratively from sets of equilibration runs. During the initial run, τ_{eff} is set equal to τ . For the next runs, the value of τ_{eff} is obtained from the values of τ_{eff} computed with Eq. (24) 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 τ_{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 τ_{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 τ increases, τ_{eff} has a maximum as a function of τ . However, since the time-step error increases with τ , the largest values of τ that we have used were always smaller than this "optimum."

Algorithms that do not exactly simulate the equilibrium distribution of the drift-diffusion equation without the branching term, can for sufficiently large τ 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 can be expected to 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.

As mentioned above, the notion of including an accept/reject step was first introduced by Reynolds *et al.*³ Their algorithm differs from ours in that an accept/reject step is performed on each individual electron move separately rather than on the full n -electron move. We will see that the approximate Green function of Eq. (1) is a poor approximation to the true Green function near nuclei.

Consequently, moves of electrons close to nuclei are much more likely to be rejected than moves of electrons far from nuclei, whereas in our algorithm all the electrons either move or do not move. Furthermore, their value of τ_{eff} fluctuates and is different for each walker, whereas ours is a constant for all walkers and all times.

B. 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 will now discuss. We are unaware of any analysis in the literature of this pathology of the algorithm although persistent configurations (or trapped walkers) have been observed by others.^{33,34}

For a given configuration \mathbf{R} , consider the quantity $P = \langle q\Delta w \rangle$, where q and Δw are the rejection probability and the branching factor given by Eqs. (22) and (25). The average in the definition of P is over all possible moves for the configuration \mathbf{R} under consideration. If the local energy at \mathbf{R} is relatively low and τ_{eff} is sufficiently large, P 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_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 $P = \langle q\Delta w \rangle$. This process produces a cascade of configurations of ever decreasing energies. Both sorts of occurrences are demonstrated in Fig. 1. Persistent configurations are most likely to occur near nodes or near nuclei. 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 were never encountered in the longest runs we performed. We note that Fig. 1 was produced by choosing a value of $\tau = 0.1 \text{ H}^{-1}$, which is half the largest time step that we will use in Section IV in the calculations performed with the final version of the improved algorithm. On the other hand, in runs without these modifications, we have observed persistency even for $\tau = 0.025 \text{ H}^{-1}$ though much less frequently than for $\tau = 0.1 \text{ H}^{-1}$. Hence if one employs an algorithm that includes the accept/reject step, but does not include the other modifications that we will describe, then it is necessary to use relatively very small time steps to avoid persistency. It should be noted that in the algorithm of Ref. 3, the accept/reject step is performed on individual electrons rather than on configurations. Hence, that algorithm runs the risk of

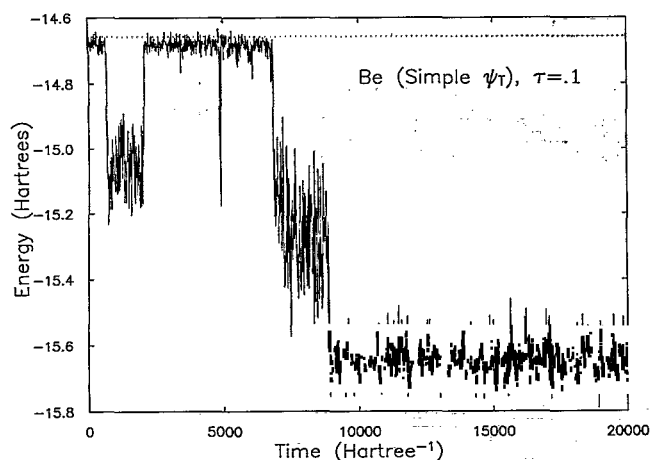


FIG. 1. Illustration of the persistent configuration catastrophe. The dotted horizontal line is the true fixed-node energy for the simple Be wave function extrapolated to $\tau=0$.

having persistent electrons rather than persistent configurations. Since the acceptance for electrons that are close to a nucleus is small, they are particularly likely to be persistent. Hence, that algorithm also suffers from this pathology, but in a less severe form. In practice various authors^{3,5} have made the occurrence of persistent configurations very unlikely by using a sufficiently small time-step, that the average acceptance is close to one (typically 0.99).

Despite the fact that the modifications described in the next section eliminated persistent configurations for the systems we studied, it is clearly desirable to have an algorithm that cannot display this pathology even in principle. We tested three different methods for doing this which we describe next.

The first method we tried was to force walkers to move at each step by eliminating the accept/reject step while maintaining an algorithm that samples $|\psi_0(\mathbf{R})|^2$ exactly in the limit that $\psi_T(\mathbf{R}) = \psi_0(\mathbf{R})$. It is shown in Appendix C that it is possible to sample $|\psi_T(\mathbf{R})|^2$ exactly by replacing the accept/reject step by an additional reweighting

$$\Delta w = \frac{|\psi_T(\mathbf{R}')|^2 \tilde{G}(\mathbf{R}, \mathbf{R}', \tau)}{|\psi_T(\mathbf{R})|^2 G(\mathbf{R}', \mathbf{R}, \tau)}. \quad (27)$$

The usual reweighting by Δw of Eq. (25) then converts an exact sampling of $|\psi_T|^2$ to an approximate sampling of $\psi_T \psi_0$ as before. This method was discarded because it leads to statistical errors that are a few times larger.

The second method was to replace τ_{eff} in Eq. (23) by τ for an accepted move and by zero for a rejected move. This ensures that Δw 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 τ_{eff} . However, this method led to a time-step error that was about a factor of two larger in the case of Be.

The final solution that we adopted was to monitor the *age* a' of each walker, defined as the number of generations for which the walker had persisted at the same position.

TABLE I. Behavior of the local energy E_L and velocity v as a function of the distance R_1 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 ψ_T that fails to satisfy the cusp conditions.

Region	Local energy	Velocity
Nodes	$E_L \sim \pm \frac{1}{R_1}$ for ψ_T	$v \sim \frac{1}{R_1}$
	$E_L = E_0$ for ψ_0	
Electron/ nucleus/electron	$E_L \sim \frac{1}{x}$ for some ψ_T	v has a discontinuity
	$E_L = E_0$ for ψ_0	for both ψ_T and ψ_0

The acceptance probability of walkers with age greater than 50 was multiplied by the exponentially growing factor $1.1^{a'-50}$. Of course, this solution to the persistency problem strictly speaking violates detailed balance but since it is never exercised in the $\tau = 0$ limit, it leaves the limit unchanged. Since the oldest walker that we ever encountered, in the algorithm that incorporates the modifications described in the next section, had $a'=21$ for Be and Li_2 and $a'=40$ for Ne, in runs with as many as 8×10^7 Monte Carlo steps, this solution to the persistent configuration problem was in fact never exercised in obtaining the results presented in this paper. It is presented here merely as an algorithm that cannot even in principle exhibit persistency, even for trial wave functions that are much inferior to those used here.

C. Singularities

The number of iterations of Eq. (5) required for the power method to converge to the ground state grows inversely with the time step τ . Thus, the statement made above, *viz.* that the Green function of Eq. (6) 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 nonanalyticities in the local energy and the velocity may invalidate this argument: the short-time Green function lacks uniform convergence in τ over $3n$ -dimensional configuration space. We have modified the Green function of Eq. (6) to take into account these singularities in a simple and approximate way, such that the Green function reduces to the original form of Eq. (6) far from the nonanalyticities.

More specifically, for a generic approximate trial wave function, the local energy diverges at nodes and at electron–nucleus and electron–electron overlaps. Both for approximate and exact wave functions, the velocity diverges at the nodes and has a discontinuity at the nucleus. The remaining improvements of the simple algorithm focus on these regions of the n -electron configuration space, since they make large contributions to the time-step error. In

particular, we shall systematically address the singularities summarized in Table I.

The local energy diverges at particle overlaps unless ψ_T satisfies the cusp conditions.³⁵ Failure to impose in particular the electron–nucleus cusp condition significantly increases the time-step error. More specifically, if the wave function ψ_T fails to satisfy the electron–nucleus cusp condition, the local energy diverges to $+\infty$ as an electron approaches the nucleus if the magnitude of the cusp is too large and to $-\infty$ in the opposite case. In addition, the electron distribution of the simple algorithm has a considerable time-step error at the nucleus since the discontinuity in the velocity at the nucleus can cause the electrons to overshoot it, resulting in a reduced density of electrons at the nucleus as τ is increased. The combined effect is a very large total time-step dependence. We note that this error can be either positive or negative, depending on whether the magnitude of the cusp is too small or too large, respectively.

In practice, imposing the cusp conditions on the trial wave functions for atoms is straightforward since all orbitals are centered at the same nucleus. For molecules it is necessary to consider the contributions to the value of an orbital at a nucleus coming from basis functions centered at other nuclei. The electron–nucleus cusp condition for molecules is imposed iteratively during the optimization of the trial wave function. When complicated forms of the correlation functions are used, it is sometimes advantageous to use a penalty function to impose the cusp conditions as part of the trial function optimization.^{36,37}

Next we address the divergence at the nodes. The diffusion term is relatively small near the nodes, and thus we may approximate the modified Schrödinger equation, Eq. (1), by

$$\nabla \cdot \{V(\mathbf{R})f(\mathbf{R},t)\} + \{E_L(\mathbf{R}) - E_T\}f(\mathbf{R},t) = -\frac{\partial f(\mathbf{R},t)}{\partial t}. \quad (28)$$

The corresponding exact drift-branching Green function is given by

$$G_0(\mathbf{R}'', \mathbf{R}, \tau) = \delta\{\mathbf{R}'' - \mathbf{R}(\tau)\} e^{\bar{S}(\mathbf{R})\tau}, \quad (29)$$

where $\mathbf{R}(t)$ is the solution to the differential equation $d\mathbf{R}/dt = \mathbf{V}$, satisfying the boundary condition $\mathbf{R}(0) = \mathbf{R}$, and

$$\bar{S}(\mathbf{R}) = \frac{1}{\tau} \int_0^\tau [E_T - E_L\{\mathbf{R}(t)\}] dt. \quad (30)$$

This can be verified by using the expression $\mathbf{R}(\tau + \delta\tau) \approx \mathbf{R}(\tau) + \mathbf{V}\{\mathbf{R}(\tau)\}\delta\tau$ and expanding to linear order in $\delta\tau$ the expression

$$f(\mathbf{R}, \tau + \delta\tau) = \int G_0(\mathbf{R}, \mathbf{R}', \delta\tau) f(\mathbf{R}', \tau) d\mathbf{R}', \quad (31)$$

to show that f satisfies Eq. (28).

Our goal is to find approximations for $\mathbf{R}(t)$ and \bar{S} in Eq. (29). These approximations should be accurate, but simple enough to be expressed in terms of the wave func-

tion and its gradient, or more generally in terms of quantities that are already being computed to execute the simple diffusion Monte Carlo algorithm.

First consider $\mathbf{R}(t)$. The expression $\mathbf{R}(t) \approx \mathbf{R} + \mathbf{V}(\mathbf{R})t$, as used in Eq. (6), derives from the approximation that $\mathbf{V}(\mathbf{R}) = \nabla\psi_T(\mathbf{R})/\psi_T(\mathbf{R})$ is constant over the integration path. However, \mathbf{V} diverges in the vicinity of the nodes of the trial function, and under those circumstances it is more accurate to assume that $\nabla\psi_T$ rather than \mathbf{V} is constant. To be precise, within the linear approximation of the wave function, the velocity is given by $\mathbf{V} \approx \hat{R}_\perp/R_\perp$, where R_\perp is the distance to the node and \hat{R}_\perp is a unit vector orthogonal to the nodal surface pointing from the nodal surface to \mathbf{R} . Integration of the equation of motion gives

$$R_\perp(t) - R_\perp(0) = \sqrt{R_\perp(0)^2 + 2t} - R_\perp(0) = \bar{V}(t)t, \quad (32)$$

by definition of \bar{V} . In other words, in Eq. (6) expression $\mathbf{R} + \mathbf{V}(\mathbf{R})\tau$ is replaced by $\mathbf{R} + \bar{\mathbf{V}}(\mathbf{R})\tau$ with

$$\bar{\mathbf{V}} = \frac{-1 + \sqrt{1 + 2V^2\tau}}{V^2\tau} \mathbf{V} \quad (33)$$

with V evaluated at $\mathbf{R} = \mathbf{R}(0)$. $\bar{\mathbf{V}}$ reduces to \mathbf{V} for small $V^2\tau$ but the magnitude of the drift, $\bar{V}\tau$, is limited to $\sqrt{2\tau}$ for large $V^2\tau$.

One can make an assumption intermediate between $\mathbf{V} = \nabla\psi_T/\psi_T$ and $\nabla\psi_T$ being constant by introducing a parameter a , approximately in the range 0 to 1, and define

$$\bar{\mathbf{V}} = \frac{-1 + \sqrt{1 + 2aV^2\tau}}{aV^2\tau} \mathbf{V}. \quad (34)$$

With this choice, the small $V^2\tau$ limit of the drift is unchanged at $\mathbf{V}\tau$, but the large $V^2\tau$ limit is $\sqrt{2\tau}/a$.

In our actual calculations a more complicated form was used for the curve traced out by the drifting n -electron configuration. Velocities are large near a node but this is also true near a nucleus. The velocity goes to infinity near a node and it is typically as large as Z near a nucleus. The constant a was made position dependent in a way that roughly distinguishes largeness of V in the proximity of a node and in proximity of a nucleus: a should be close to 1 near a node and close to 0 near a nucleus. For this purpose, first of all the drift of the whole n -electron configuration is broken up into contributions of individually drifting electrons. In terms of single electron coordinates we have

$$\bar{\mathbf{v}} = \frac{-1 + \sqrt{1 + 2a(\mathbf{r})v^2\tau}}{a(\mathbf{r})v^2\tau} \mathbf{v}. \quad (35)$$

Secondly, the cross-over parameter a introduced in Eq. (34) is made dependent on position as indicated above

$$a(\mathbf{r}) = \frac{1}{2}(1 + \hat{v} \cdot \hat{z}) + \frac{Z^2 z^2}{10(4 + Z^2 z^2)}, \quad (36)$$

where \hat{z} is a unit vector from the nearest nucleus to the electron, which is at distance z , while \hat{v} is a unit vector in the direction of the single electron velocity (see Fig. 2); only the order of magnitude of the various constants in this expression has physical significance. The motivation for

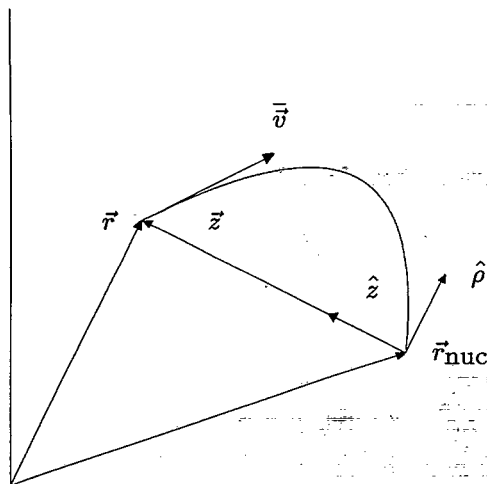


FIG. 2. An electron initially at \mathbf{r} starts to drift in the direction $\bar{\mathbf{v}}$ and at sufficiently long times comes to rest at the position of the nearest nucleus \mathbf{r}_{nuc} . Also shown is the cylindrical coordinate system with unit vectors $\hat{\rho}$ and \hat{z} , both of which are coplanar with \mathbf{v} .

these two terms is that the first term is close to one if the velocity of the electron is directed away from the nearest nucleus, which is an indication of the proximity of a node. The second term ensures that the value of a is never very small when the electron is not close to any nucleus, thereby limiting the magnitude of the drift far from nuclei to reasonable values. Note that for nonzero τ , Eqs. (34) and (35) give slightly different results even for constant a .

It is illuminating to compare Eq. (35) with the result for the average drift obtained from the exact Green function for the modified Schrödinger equation for a particle in one dimension in the immediate vicinity of a node in the guiding function

$$-\frac{1}{2} \frac{\partial^2 f}{\partial x^2} + \frac{\partial(f/x)}{\partial x} = -\frac{\partial f}{\partial t}. \quad (37)$$

The substitution $f = \psi x$ yields the Schrödinger equation in imaginary time for a free particle subject to the boundary condition that ψ have a node at $x=0$. The Green function of the latter is found with the method of images. The result can then be transformed back to give the Green function of Eq. (37):

$$\tilde{G}(x, x', t) = \frac{x}{x' \sqrt{2\pi t}} \{e^{-(x-x')^2/2t} - e^{-(x+x')^2/2t}\}. \quad (38)$$

This yields a large τ limit of the average drift of $\sqrt{8\tau/\pi}$, corresponding to $a=\pi/4$ in Eq. (35). We note that, in agreement with the exact Green function of Eq. (37), our improved Green function goes to zero quadratically at the nodes (because of the accept/reject step) whereas the simple Green function goes to a nonzero constant. In principle one could modify the algorithm so as to sample from the Green function given in Eq. (38) for the direction perpendicular to a nearby nodal plane, but we have not found that necessary.

Next we deal with the divergence of the local energy at the nodes. The expression for $\mathbf{R}'(t)$ found above with $\bar{\mathbf{V}}$ given in Eq. (33) can be used to obtain an approximate expression for $S(\mathbf{R})$ as given in Eq. (30). With the approximation $E_L(R_\perp) \approx E_{\text{est}} + (B/R_\perp)$, where E_{est} is the current estimate of the energy, integration over the path yields

$$\bar{S}(\mathbf{R}) = \left\{ (E_T - E_{\text{est}}) + [E_{\text{est}} - E_L(\mathbf{R})] \frac{\bar{V}(\mathbf{R})}{V(\mathbf{R})} \right\}, \quad (39)$$

where all position dependent quantities on the right are evaluated at \mathbf{R} . It should be noted that Eq. (39) was derived assuming a single constant a , while in practice \bar{V}^2 is calculated by summing the squared single electron velocities v_i , each of which has a different value of a . For small τ this expression (39) differs appreciably from the original expression, Eq. (3), only in the vicinity of a node. Thus we always use the branching factor, Eq. (25), with this new definition of S . It should be noted that whereas the S of Eq. (3) can diverge at nodes, the new \bar{S} does not, since both E_L and V diverge as the inverse of the distance to the node but \bar{V} does not.

Finally, we deal with inaccuracies of the short-time Green function near the nucleus. The velocity of an electron close to a nucleus is always directed approximately toward the nucleus. That is, the velocity has a discontinuity and the true Green function has a cusp at the nucleus. The short-time Green function is inaccurate first of all because the drift term can cause an electron to overshoot the nucleus. Secondly, the gaussian cannot reproduce the cusp, although for sufficiently small τ it can approximate one. Hence, the distribution of electrons in the vicinity of the nucleus depends strongly on τ and produces a large time-step error. If the drift is done before the diffusion, as is commonly the case, then the electron density close to the nuclei is diminished as τ is increased. Similar arguments apply to electron-electron overlap, but the effect on the time-step error is negligible because the electrons do not experience the discontinuity in the velocity since the velocity moves the electrons apart and because the probability of a close encounter is small. Clearly, the goal is to modify the diffusion Monte Carlo walk such that the distribution of electrons changes very little from the exact distribution as τ is increased.

The discontinuity in the velocity at the nucleus can be dealt with by working in cylindrical polar coordinates (z, ρ, ϕ) , rather than Cartesian coordinates, with $\mathbf{z} = \mathbf{r} - \mathbf{r}_{\text{nuc}}$ centered on the nearest nucleus (see Fig. 2). The velocity $\bar{\mathbf{v}}$ is resolved into a \hat{z} component \bar{v}_z in the direction of the nearest nucleus and a $\hat{\rho}$ component \bar{v}_ρ , i.e., $\bar{\mathbf{v}} = \bar{v}_z \hat{z} + \bar{v}_\rho \hat{\rho}$. If z is the initial distance to the nearest nucleus, then the final \hat{z} component of the distance after drifting, z'' , is chosen as $\max(z + \bar{v}_z \tau, 0)$. The drift in the $\hat{\rho}$ direction is chosen as $\rho'' = 2\bar{v}_\rho \tau z'' / (z + z'')$. For increasing time the electron traces a curved path that ends at the nucleus as illustrated in Fig. 2. Thus, the electron is prevented from overshooting the nucleus.

The true Green function has a cusp at the nucleus. The approximate Green function used in the Monte Carlo pro-

cess is smooth at the nucleus if the drift is done before the diffusion. On the other hand, if the order of the drift and diffusion were reversed, and the electrons are prevented from overshooting the nucleus, then the approximate Green function has a δ function at the nucleus, which is not the right behavior either. In order to have a more accurate Green function in the vicinity of the nucleus we use a form with one-particle factors that interpolate between the short-time Gaussian diffusion kernel and the hydrogenic ground state wave function to which for long times they would evolve in the absence of other electrons or nuclei. The crossover between the two behaviors is expected to occur on a time scale such that the electron would move beyond the nucleus by pure diffusion with appreciable probability. Hence, we sample from a Gaussian $(2\pi\tau)^{-3/2}e^{-r'^2/2\tau}$ with probability \tilde{p} , and from an exponential $(\xi^3/\pi)e^{-2\xi r'}$ with probability $\tilde{q}=1-\tilde{p}$. The Gaussian is always centered at the position of the electron after it has drifted, whereas the exponential is always centered at the nearest nucleus prior to drifting. Note that if $z+\bar{v}_z\tau < 0$ then both the Gaussian and the exponential are centered at the same point. The value of \tilde{q} is chosen to equal the probability that the electron diffuses across a plane through the nearest nucleus, perpendicular to the line from the electron to the nucleus, i.e.,

$$\tilde{q} = \frac{1}{2} \operatorname{erfc} \left(\frac{z + \bar{v}_z\tau}{\sqrt{2\tau}} \right), \quad (40)$$

z being the distance to the nearest nucleus prior to drifting. The inverse length ξ in the exponent is chosen to be $\sqrt{Z^2 + 1/\tau}$. For small τ , $\xi = 1/\sqrt{\tau}$, i.e., the second moment of the exponential equals that of the Gaussian. In this way, we maintain a Green function correct to $\mathcal{O}(\tau)$. On the other hand, for large τ , $\xi = Z$, so that the Green function has the correct cusp

$$\left. \frac{\partial \tilde{G}(\mathbf{R}', \mathbf{R}, \tau) / \partial r'}{G(\mathbf{R}', \mathbf{R}, \tau)} \right|_{r'=0} = -2Z,$$

where r' is the distance of any one of the electrons to a nucleus.

In summary, in this section we have proposed several simple modifications to the Green function that take into account its nonanalyticities. These improvements, not only reduce the time-step errors but also increase the acceptance probability because if the true importance-sampled Green function $G(\mathbf{R}', \mathbf{R}, \tau)$ were used to generate moves, then all proposed moves would be accepted. This follows from the fact that the Green function of the original Schrödinger equation, prior to the importance-sampling transformation of Eq. (1), is symmetric in its arguments and the definition of the importance sampled Green function in terms of the original Green function. So, the improvements in the Green function not only reduce the time-step error but also enhance the acceptance and consequently the efficiency of the algorithm and reduce greatly the chance of encountering persistent configurations. Furthermore, if the accept/reject were done for each electron, as in the algorithm of Reynolds *et al.*³ then these improvements to $G(\mathbf{R}', \mathbf{R}, \tau)$

would have served to make the acceptance probability larger and more uniform for all the electrons, whereas the algorithm of Ref. 3 has a considerably lower acceptance for electrons close to a nucleus, than for electrons far from the nuclei.

IV. RESULTS

In this section we present a numerical comparison of the simple and improved diffusion Monte Carlo algorithms described above. Results are presented for Be, Li₂, and Ne, these being representative of a light atom, a molecule and a heavy atom. Ne is the heaviest atom that has been treated to date by diffusion Monte Carlo without the use of methods that treat core electrons approximately. Hence, this constitutes a severe test of the method. We also present results both for a simple and a very good wave function for Be in order to study the behavior of the algorithm as the quality of the wave function is improved. We have applied the algorithm to other wave functions, not presented here, with equally satisfactory results.

In employing a particular algorithm for a given atom or molecule a practical complication in making a comparative evaluation is that the time-step error depends strongly on the trial wave function. In the limit that the trial wave function approaches an eigenfunction the simple algorithm has time-step errors that vanish for the mixed estimator for the energy, but not for the growth estimator or the mixed estimators of quantities that do not commute with the Hamiltonian. (The nonvanishing of the error of the growth estimator is directly related to killing walkers that cross nodes.) On the other hand, the improved algorithm has vanishing time-step errors for all quantities. The simple algorithm has a vanishing variance for the mixed estimator of the energy in the $\psi_T \rightarrow \psi_0$ limit, but not for the growth estimator of the energy or the mixed estimators of operators that do not commute with \mathcal{H} . The improved algorithm has vanishing variance for both estimators of the energy, but not for operators that do not commute with \mathcal{H} . In view of the dependence on the trial wave functions, we have tested our algorithm first employing simple wave functions for Be, Li₂, and Ne, consisting of a single configuration determinant multiplied by a simple Jastrow function. These wave functions are of quality roughly comparable to many of the wave functions used in the literature. Then, in one case, we repeated the computation with an accurate, two-configuration wave function, with a complicated Jastrow factor of the form described elsewhere,³⁶⁻³⁸ a wave function much more accurate than any used by other authors. The wave functions are described in detail in Appendix B.

A. Simple wave function comparisons

Figures 3, 4, and 5 are plots of the total energy as a function of the time step for simple Be, Ne, and Li₂ wave functions. The triangles and crosses are the results obtained with the simple diffusion Monte Carlo algorithm, and, respectively, display data for the mixed and growth estimators. The various curves are polynomial fits in integral powers of $\sqrt{\tau}$. We have reason to believe (see Appen-

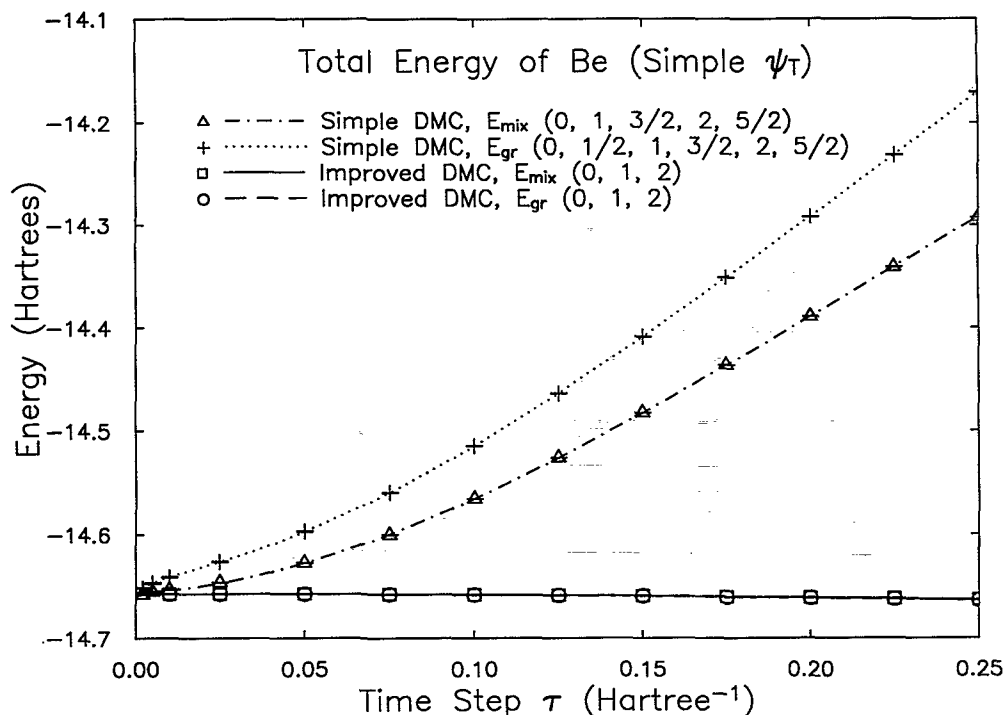


FIG. 3. Time-step dependence of the total energy of Be using a simple wave function. The triangles (crosses) are the energies from the mixed (growth) estimator in simple diffusion Monte Carlo and the dashed-dotted (dotted) line is a polynomial fit in powers of τ . The powers of τ included in the fit are shown in the legend. The squares/circles are the energies from the mixed (growth) estimator in the improved diffusion Monte Carlo algorithm and the solid (dashed) line is a fit. The two curves are almost indistinguishable. The error bars are plotted, but appear as horizontal ticks since the errors are small.

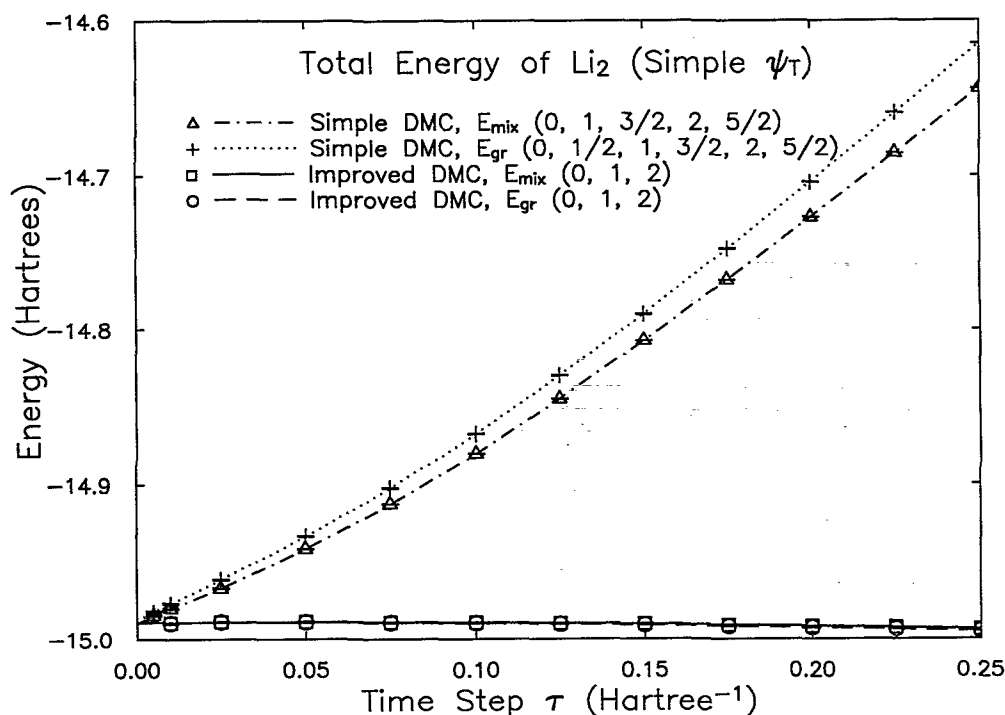


FIG. 4. Same as Fig. 3 but for a simple Li₂ wave function.

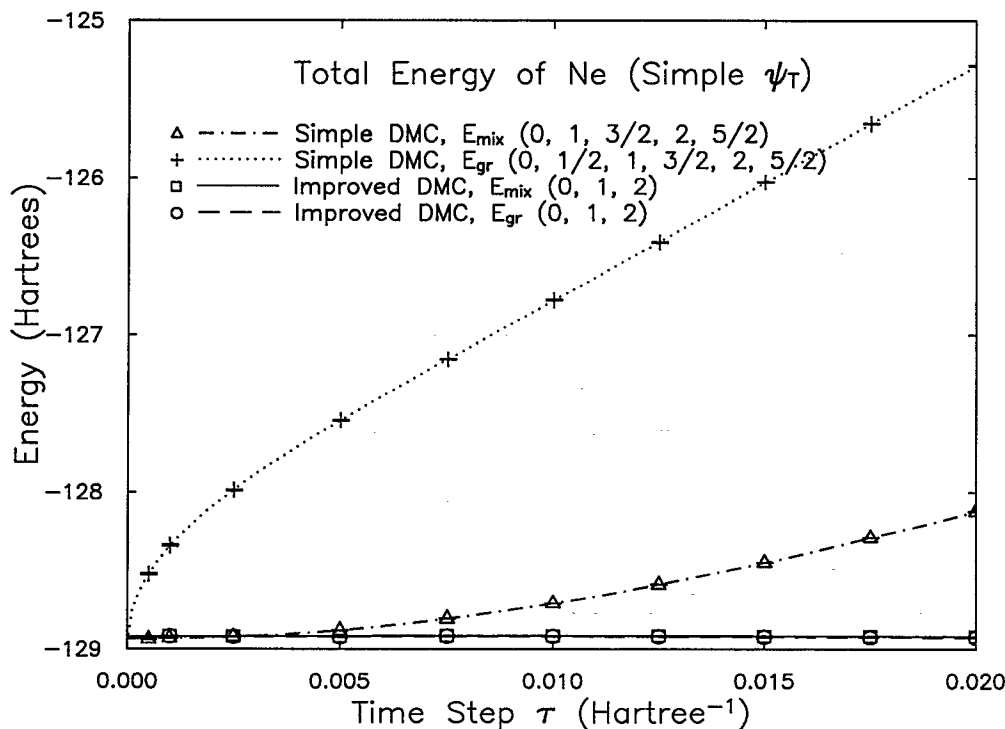


FIG. 5. Same as Fig. 3 but for a simple Ne wave function.

dix D) that the term proportional to $\sqrt{\tau}$ is missing for the time-step dependence of the mixed estimator, in which fit this term was suppressed. The powers of τ included in the fit are shown in the parentheses in the legends. The squares and circles, respectively, display data for the mixed and growth estimators obtained from the improved algorithm. The two estimators are in all cases sufficiently close that it is difficult to tell them apart.

Table II shows the time-step errors for the two algorithms at $\tau=0.2 \text{ H}^{-1}$ for Be and Li_2 and $\tau=0.015 \text{ H}^{-1}$ for Ne. The last column shows that the improved algorithm has from 70 to 300 times smaller errors in the mixed estimator and from 70 to 1100 smaller errors in the growth estimator. The improvements are even greater at the smaller time steps.

Figures 6 and 7 are the same as Figs. 3 and 4 but with an expanded energy scale to illustrate the functional form of time-step dependence of the improved algorithm. Although the mixed and growth estimators of the energy agree very well here, the agreement in our earlier version¹⁷ of the improved algorithm was another 2–3 orders of magnitude better, in all cases. The deterioration in the agreement of the two estimators can be traced to our using Eqs. (25), (26), and (39) instead of Eqs. (23), (17), and (3) in the earlier version. We do not see a big benefit to having the two estimators agree to orders of magnitude better than the statistical errors, so we have elected to make these changes which slightly degrade the agreement of the two estimators, but reduce both the time-step errors and the statistical errors.

We notice that although we have not designed the al-

gorithm to have quadratic time-step errors, in practice the errors are quadratic to within the error bars. (The time-step error for Ne are sufficiently small that we cannot discern a clear trend above the error bars; hence an expanded energy scale plot is not presented for Ne.) This implies that the procedure used for determining τ_{eff} is particularly accurate in determining the lowest two orders of correction to τ as discussed in Section III A.

The improvement in quantities other than the total energy is in many cases even more dramatic, as can be seen from Table II. For some quantities it is impossible to see any time-step error above the statistical noise, so we give upper bounds for the error and lower bounds for the factors by which the time-step errors are reduced. The improvements range from a factor of more than 80 to a factor of more than 1000 for the simple wave functions, with yet larger improvements for the good wave function. The kinetic energy and negative moments of the charge distribution have particularly large time-step errors in the simple algorithm because the electron distribution of the simple algorithm is significantly in error near the nucleus—a region where these quantities diverge.

To study the detailed dependence of the time-step errors we present as examples, in Figs. 8–10, the mixed estimators of the kinetic energy E_{kin} and moments of the charge distribution $\langle r^2 \rangle$ and $\langle r^{-1} \rangle$ for the simple Be wave function. We have included a $\tau^{5/4}$ term in the fit of E_{kin} and $\langle r^{-1} \rangle$ from the simple algorithm because it significantly improved the quality of the fit. Chin¹⁶ has presented an argument for why this term could occur because of the discontinuity in the velocity at the nucleus. We have not

TABLE II. Comparison of the time-step errors for various quantities computed by simple and improved diffusion Monte Carlo at $\tau=0.2 \text{ H}^{-1}$ for Be and Li_2 and $\tau=0.015 \text{ H}^{-1}$ for Ne. Energies are in hartrees and lengths in Bohr radii. Statistical uncertainty in the last digit is given in parentheses. The last column gives the absolute value of the ratio of the simple diffusion Monte Carlo error to the improved diffusion Monte Carlo error.

Wave function	Observable	Extrapolated value at $\tau=0$	Error at largest τ		Error ratio
			Simple DMC	Improved DMC	
Be Simple	E_{mix}	-14.6568(2)	+0.268	-0.0038	71
	E_{gr}	-14.6568(2)	+0.366	-0.0042	87
	E_{kin}	14.708(4)	-15.53	+0.07	190
	$\langle r^2 \rangle$	3.956(4)	+1.12	<0.006	>185
	$\langle r^{-1} \rangle$	2.1120(3)	-1.08	+0.0043	250
Be Good	E_{mix}	-14.66719(3)	+0.052	-0.00017	310
	E_{gr}	-14.66718(3)	+0.146	-0.00013	1100
	E_{kin}	14.674(3)	-15.85	<0.004	>4000
	$\langle r^2 \rangle$	4.020(3)	+1.10	<0.005	>220
	$\langle r^{-1} \rangle$	2.1076(2)	-0.98	<0.0003	>3300
Li_2 Simple	E_{mix}	-14.9890(2)	+0.262	-0.0030	87
	E_{gr}	-14.9890(2)	+0.285	-0.0039	73
	E_{kin}	-14.897(2)	-12.4	+0.041	300
Ne Simple	E_{mix}	-128.919(3)	+0.48	<0.004	>125
	E_{gr}	-128.919(3)	+2.90	<0.004	>720
	E_{kin}	129.44(2)	-90.7	<0.05	>1800
	$\langle r^2 \rangle$	0.882(1)	+0.159	<0.002	>80
	$\langle r^{-1} \rangle$	3.1247(5)	+0.99	<0.001	>1000

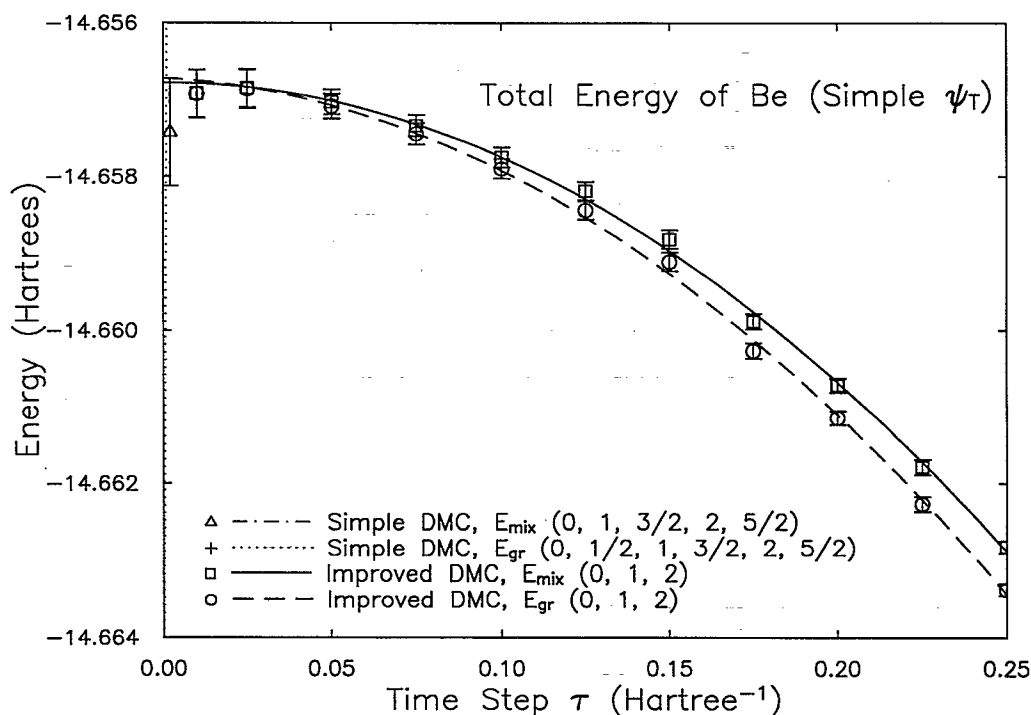


FIG. 6. Same as Fig. 3 but with an expanded energy scale, which makes the error bars visible, to show the time-step dependence of the improved algorithm. Note that the results for the simple diffusion Monte Carlo algorithm are almost completely off-scale. The fits for the improved algorithm include terms only up to 2nd order. The fitted value of the linear coefficient is zero within statistical errors.

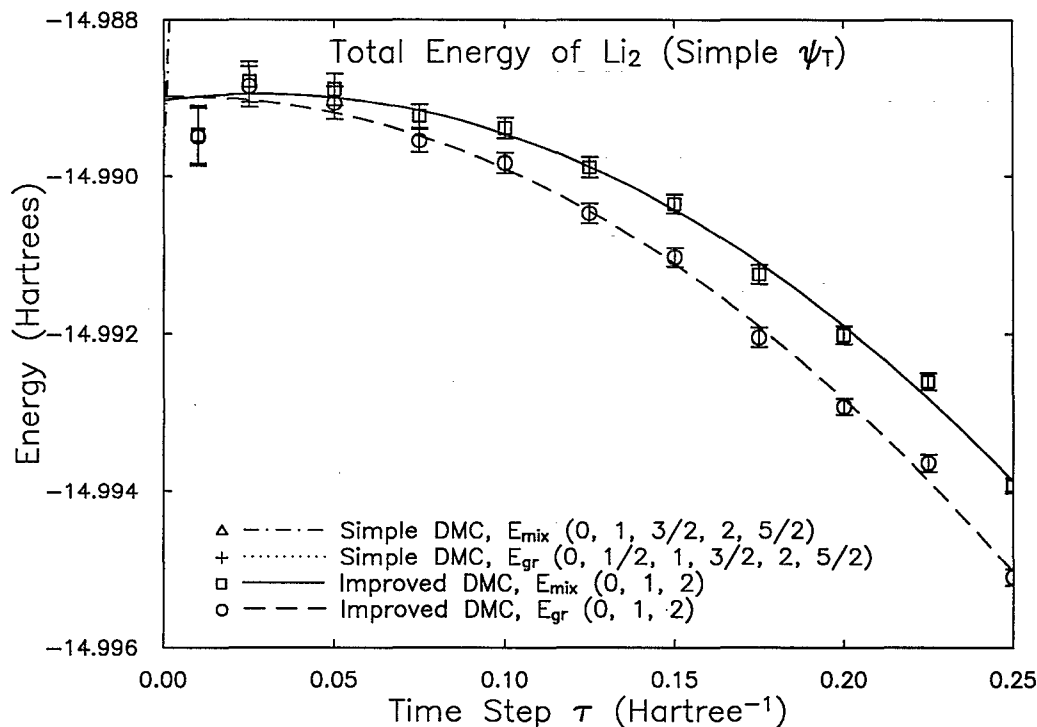
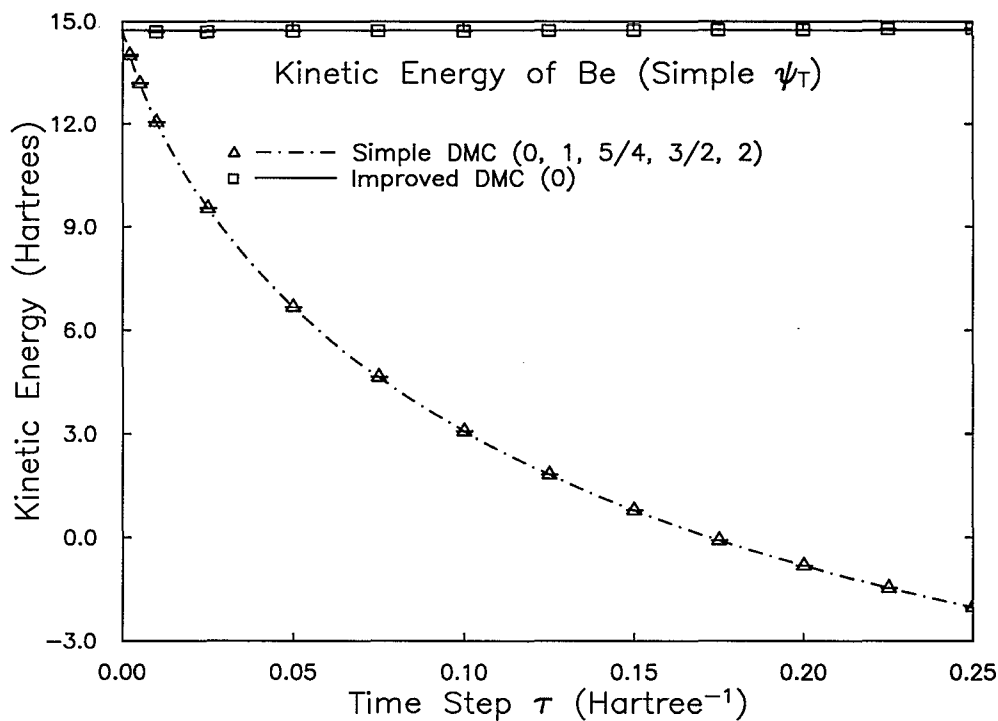
FIG. 7. Same as Fig. 6 but for Li_2 .

FIG. 8. Time-step dependence of the mixed estimator of the kinetic energy of Be using a simple wave function. The triangles are the kinetic energies from the mixed estimator in simple diffusion Monte Carlo and the dashed-dotted line is a fit. The squares are the energies from the mixed estimator using the improved diffusion Monte Carlo algorithm and the solid line is a fit. The powers of τ included in the fit are shown in the legend. The error bars are plotted, but appear as horizontal ticks since the errors are small.

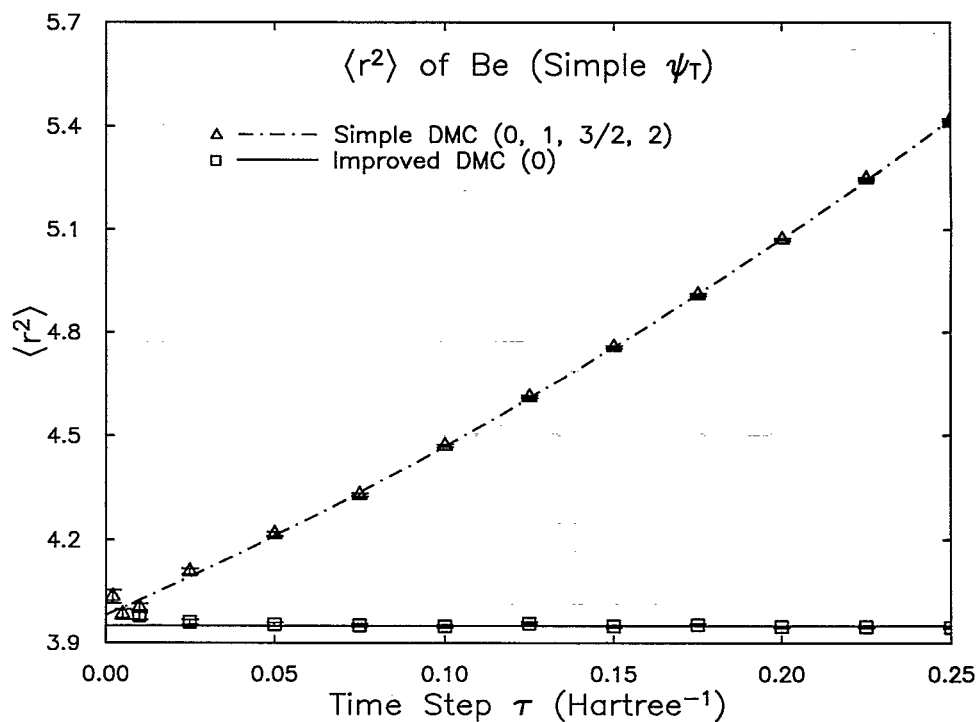


FIG. 9. Same as Fig. 8 but for the mixed estimator of $\langle r^2 \rangle$. The error bars in this case are large enough to be seen clearly at the shorter time-step values.

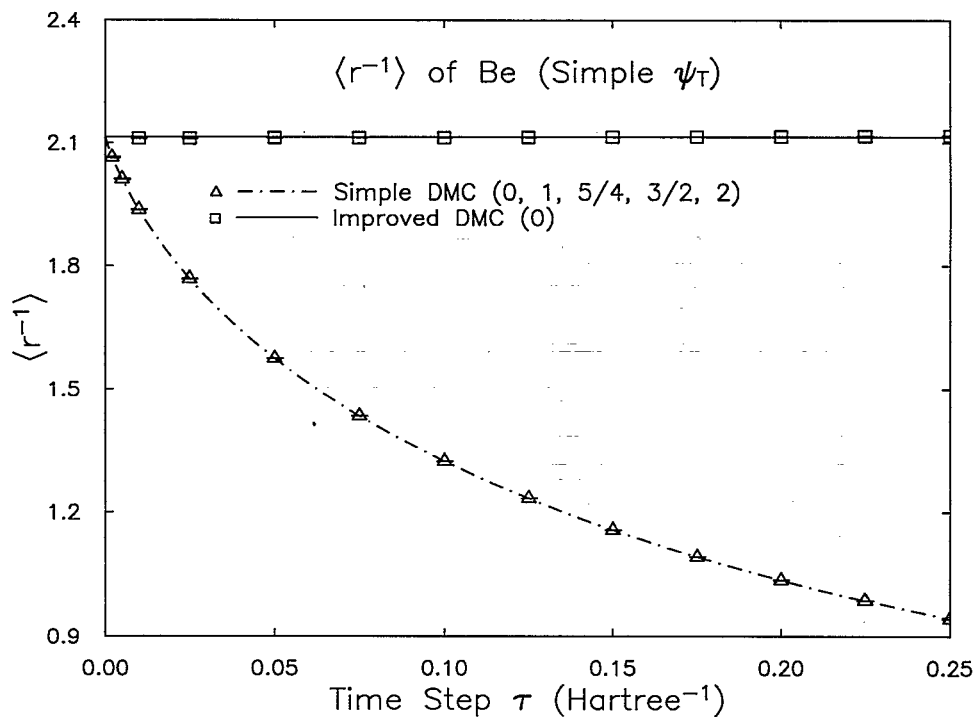


FIG. 10. Same as Fig. 8 but for the mixed estimator of $\langle r^{-1} \rangle$.

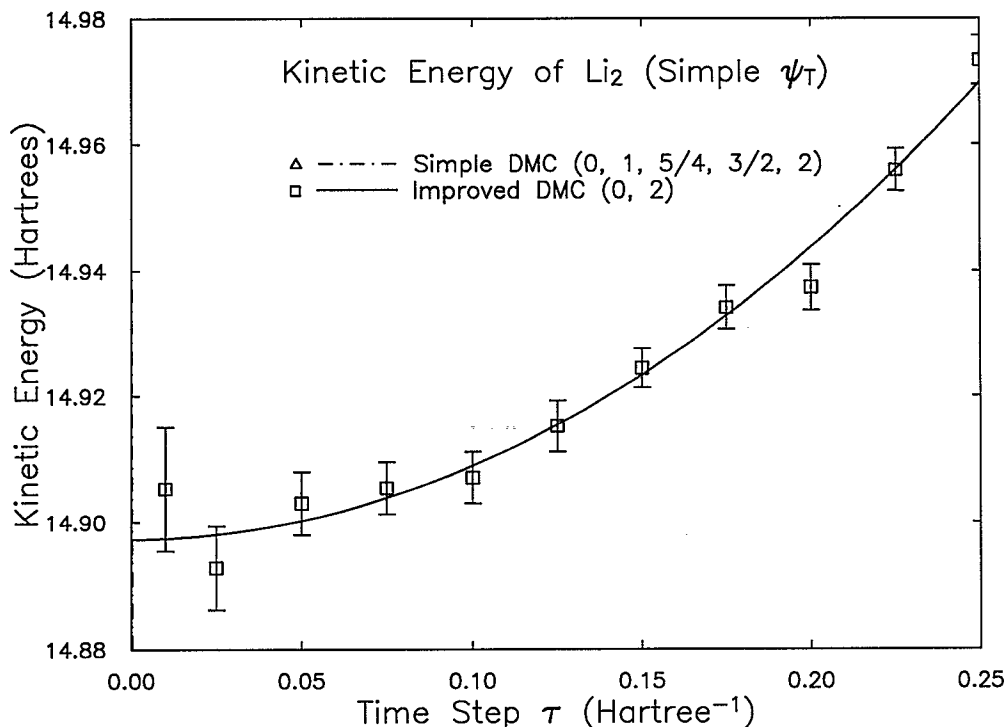


FIG. 11. Mixed estimator of the kinetic energy of Li_2 . Note the time-step error for simple diffusion Monte Carlo is so big, that the curve is almost indistinguishable from the y axis. The fits for the improved algorithm include terms only up to 2nd order. The fitted value of the linear coefficient is zero within statistical errors.

verified in detail the applicability of his arguments to the simple algorithm, but the empirical evidence for the existence of the $\tau^{5/4}$ term is strong. For example, replacing the $\tau^{5/4}$ term by a $\tau^{1/2}$ term in the fit for the kinetic energy increased the value of χ^2 from 1.3 to 29.3 and yielded an extrapolated value that is in error by 0.9 H, whereas leaving out the $\tau^{5/4}$ term all together resulted in a χ^2 of 547 and an extrapolated value that is in error by -0.8 H. Our preliminary work¹⁷ has plots of the kinetic energy for 3 of the 4 wave functions used in this paper which clearly show that the correct extrapolated value is not obtained in the simple algorithm if the $\tau^{5/4}$ term is omitted.

The time-step errors for quantities other than the energy (in those cases where it is large enough to be clearly discernible beyond the statistical errors) also appears to be quadratic. An example of this is shown in Fig. 11 where the mixed estimator for the kinetic energy of Li_2 is plotted. This is in contrast to Chin's algorithm¹⁶ which has quadratic time-step errors only for the total energy and only for nodeless wave functions.

In Table III we present values of σ , the fluctuation of the local energy, T_{corr} , the autocorrelation time of the energy, and $\sigma^2 T_{\text{corr}}$ for both algorithms and τ_{eff}/τ and the average acceptance for the improved algorithm. These quantities were measured in runs with a target population size of 100 walkers that were propagated for 4×10^5 (Be and Li_2) or 8×10^5 (Ne) MC generations, making for a total of 4×10^7 or 8×10^7 Monte Carlo moves. In order to calculate T_{corr} , the Monte Carlo generations were divided into 100 blocks, each consisting of 4×10^3 or 8

$\times 10^3$ Monte Carlo moves. Note that the number of Monte Carlo moves in a block is in all cases much larger than T_{corr} ; else the value of T_{corr} would have been systematically underestimated. By repeating runs with different random numbers, we have determined that the errors in σ are in the third significant digit and those in T_{corr} in the second significant digit.

If we ignore fluctuations, mostly due to statistical errors in the estimation of T_{corr} , the trends in Table III are very clear. $\sigma^2 T_{\text{corr}}$ is a measure of the number of Monte Carlo moves needed to reduce the statistical error to a given value. For a given value of τ , the value of T_{corr} is larger for the improved algorithm than the simple one because some of the proposed Monte Carlo moves are rejected. This is reflected in τ_{eff} being smaller than τ . However, the simple algorithm has a considerably larger σ than the improved algorithm at the larger time steps because as τ is increased it has an increasing density of walkers at the nodes—a region where the local energy diverges. The value of σ actually goes down a little with increasing τ in the improved algorithm because of the averaging present in Eqs. (25) and (26). In our earlier version of the algorithm,¹⁷ where we used Eqs. (23) and (17) rather than Eqs. (25) and (26) the values of σ were independent of τ , as expected. The product $\sigma^2 T_{\text{corr}}$ is always considerably smaller for the improved algorithm than for the simple one at the larger time-steps. Hence, not only does the improved algorithm have a much smaller time-step error and a higher efficiency due to the possibility of using larger val-

TABLE III. Comparison of σ , the fluctuation of the local energy, and T_{corr} , the autocorrelation time of the energy, for the simple and improved diffusion Monte Carlo algorithms. The product $\sigma^2 T_{\text{corr}}$ is a measure of the number of Monte Carlo iterations required to reduce the statistical error to a given value. The time step τ is in inverse hartrees; E and σ are in hartrees; and T_{corr} are in units of τ . The numbers in parentheses are the statistical errors in the last digits.

Wave function	Simple diffusion Monte Carlo				Improved diffusion Monte Carlo				
	τ	σ	T_{corr}	$\sigma^2 T_{\text{corr}}$	σ	T_{corr}	$\sigma^2 T_{\text{corr}}$	τ_{eff}/τ	Acceptance
Be	0.010	0.384	36.0	5.30	0.385	25.9	3.85	0.954	0.963
Simple	0.050	0.373	10.5	1.46	0.368	8.5	1.15	0.828	0.861
	0.100	0.413	3.4	0.58	0.361	5.4	0.70	0.773	0.822
$E_{\text{VMC}} = -14.6275(1)$	0.150	0.465	2.4	0.53	0.359	4.7	0.61	0.758	0.813
$E_{\text{DMC}} = -14.6569(2)$	0.200	0.521	2.1	0.57	0.358	2.9	0.37	0.754	0.809
	0.250	0.574	1.8	0.59	0.358	2.2	0.28	0.754	0.805
Be	0.010	0.125	13.5	0.213	0.085	28.0	0.200	0.954	0.963
Good	0.050	0.166	2.3	0.064	0.078	6.5	0.040	0.828	0.861
	0.100	0.214	1.8	0.083	0.075	3.3	0.019	0.774	0.823
$E_{\text{VMC}} = -14.66648(1)$	0.150	0.262	1.6	0.108	0.072	3.3	0.017	0.759	0.814
$E_{\text{DMC}} = -14.66717(3)$	0.200	0.316	1.6	0.163	0.071	2.3	0.011	0.754	0.808
	0.250	0.380	1.4	0.197	0.070	2.1	0.010	0.751	0.802
Li_2	0.010	0.405	34.6	5.68	0.408	33.4	5.58	0.963	0.968
Simple	0.050	0.394	9.4	1.46	0.390	11.5	1.75	0.836	0.858
	0.100	0.410	4.9	0.82	0.378	5.1	0.72	0.751	0.786
$E_{\text{VMC}} = -14.9472(2)$	0.150	0.442	3.1	0.61	0.371	4.5	0.62	0.710	0.753
$E_{\text{DMC}} = -14.9890(2)$	0.200	0.482	2.8	0.66	0.370	3.6	0.49	0.689	0.740
	0.250	0.530	2.6	0.72	0.368	3.3	0.45	0.679	0.734
Ne	0.0010	2.084	134.6	584.0	1.861	139.9	485.0	0.968	0.971
Simple	0.0050	2.205	25.6	124.4	1.805	37.1	120.7	0.843	0.856
	0.0100	2.319	13.0	70.3	1.774	17.6	55.2	0.738	0.759
$E_{\text{VMC}} = -128.713(2)$	0.0150	2.485	11.3	69.8	1.760	17.6	54.6	0.667	0.693
$E_{\text{DMC}} = -128.922(4)$	0.0200	2.676	6.9	49.4	1.751	15.5	47.6	0.615	0.643

ues of τ , but even for a given large value of τ it has a higher efficiency.

At present, variational Monte Carlo is often resorted to when diffusion Monte Carlo is deemed too computationally expensive. However, the improved algorithm permits one to use large values of τ , so that the values of T_{corr} are actually smaller than that in most variational Monte Carlo algorithms in use, except for the recent accelerated Metropolis algorithm.³⁹ Hence the improved algorithm should be applicable to any system that hitherto could only be treated by variational Monte Carlo.

B. Comparison with the literature

Thus far, we have compared the results of our improved algorithm with those of the simple algorithm which we claim to represent fairly some fraction of the algorithms found in the literature. To convince the reader that we have done more than just defeat a poor caricature, we proceed to compare with published results. As mentioned, this is hampered by the dependence of the time-step error on the trial wave function. Nevertheless, it is possible to roughly compare the time-step errors by using as a gauge the complexity of the wave functions or the value of the variational energy. Three groups of algorithms can be distinguished. The first group consists of algorithms that are very similar to the simple algorithm of this paper. The

second is the algorithm of Reynolds *et al.*³ which incorporates the accept/reject step, but does not contain the other improvements of our improved algorithm. The third group attempts to achieve quadratic time-step error.

First, we compare with algorithms similar to the simple diffusion Monte Carlo algorithm described above, possibly differing from it only in detail, e.g., the detailed form of the cutoffs used for the local energy and the velocity, or whether a switch is made to a smaller time-step when the local energy or the velocity is large.^{21,40} These methods should have time-step errors very similar to those of the simple diffusion Monte Carlo algorithm used here.

Garmer and Anderson²¹ used an extended basis set wave function for the F atom and report³⁰ a time-step error of 0.33 H at $\tau=0.012 \text{ H}^{-1}$. By way of comparison, the time-step error of our improved algorithm, used with an almost minimal basis set wave function, for a slightly heavier atom, Ne, is $<0.003 \text{ H}$ at the same τ , a reduction by a factor of more than 100. Garmer and Anderson²⁰ present time-step errors for another 10-electron system, methane. Their single and double ξ wave functions had a time-step error of 0.2 H and 0.09 H, respectively, at $\tau=0.005 \text{ H}^{-1}$. These errors are 500 and 225 times larger than our error for Ne at the same time-step, in spite of the fact that, for a given total nuclear charge, time-step errors

are usually smaller for molecules than for a heavy atom with the same combined nuclear charge.

A second group of results to compare with was obtained by the algorithm of Reynolds *et al.*³ This algorithm, which includes an accept/reject step, probably has the smallest time-step error of the three groups of algorithms used up to the present time. Barnett *et al.*⁴¹ use double ζ wave functions for F and F⁻ and find time-step errors of -0.019 H and -0.087 H, respectively, at $\tau=0.005$ H⁻¹. Our time-step error for Ne (<0.004 H at $\tau=0.015$ H⁻¹) is more than 43 and 195 times smaller than their error for F and F⁻, respectively, assuming that our time-step error scales quadratically in τ for Ne, as it does for Be and Li₂. Even if we make the more conservative assumption that our time-step error scales linearly in τ , our time-step error is more than 14 and 65 times smaller than their errors.

Recently, Sun *et al.*⁴² have used optimized wave functions and find time-step errors of 0.004 H at $\tau=0.1$ H⁻¹ for Li₂ and 0.04 H at $\tau=0.01$ H⁻¹ for water. Our time-step error for Li₂ at $\tau=0.1$ H⁻¹ is 0.00075 H, a factor of 5 smaller than theirs, and for Ne <0.002 H at $\tau=0.01$ H⁻¹, a factor of more than 20 smaller than theirs. In making the comparison for Li₂, it should also be noted that they have the advantage of having used a slightly better wave function: theirs has 12 basis functions and recovers 68% of the correlation energy in a variational calculation, whereas ours has 8 basis functions and recovers 62% of the correlation energy variationally. Also, in all of the comparisons with this second group of algorithms, it should be borne in mind that they perform the accept/reject step on each electron rather than on the full n -electron move, so that their algorithm is at least a factor of two more time consuming.

Finally, there have been attempts to design algorithms to achieve a small, quadratic time-step error¹²⁻¹⁶ for the energy. Vrbik¹² was the first to propose an algorithm that would have a quadratic time-step error provided that there are no discontinuities in the velocity and local energy. Vrbik and Rothstein¹³ used this algorithm along with a modified velocity to obtain what appeared to be an $\mathcal{O}(\tau^2)$ time-step error in the energy for a H₂ molecule. Rothstein, Patil, and Vrbik¹⁴ presented several versions of their algorithm, each of which has a quadratic time-step error provided that there are no nonanalyticities in the velocity or the energy. However, these algorithms when applied to H₂ and LiH did not have a quadratic time-step error even in the nodeless H₂ case, because the trial wave functions did not satisfy the correct cusp-conditions exactly. Furthermore, these authors found that algorithms with a relatively small error for one molecule have large errors for the other. DePasquale *et al.*¹⁵ have tried several $\mathcal{O}(\tau)$ modifications of the usual drift and diffusion terms in an attempt to reduce the time-step error. Their procedure lacks general validity because a different modification of the algorithm was constructed by trial and error for each of the systems studied, and in spite of this their time-step errors are large. In practice, quadratic time-step errors have only been achieved for the special case of nodeless

2-electron systems¹⁶ and only for the total energy. More importantly, the time-step error, though somewhat smaller than that of simple diffusion Monte Carlo, is nevertheless very large. For example, the preferred algorithm of Ref. 16 has a time-step error of 0.1 H at $\tau=0.4$ H⁻¹ for He—a very large value for a light atom in a nodeless 2-electron state.

The domain Green function Monte Carlo method^{4,6,7} has no time-step error. The time step in Green function Monte Carlo is not fixed but there is an average time step which is controlled by the shift in the potential. As the average time step is increased, a growing number of iterations are required to sum the Neumann series, which reconstructs the exact Green function from an approximate Green function. There is an optimal average time step, approximately given by the time step which maximizes the ratio of the average time step to the average computer time required to propagate the walkers to the next generation. The optimal average time steps for Be, Li₂, and Ne are approximately 0.0006 H⁻¹, 0.0008 H⁻¹, and 0.00007 H⁻¹, respectively. These time steps are factors of 400, 300, and 300 smaller than the largest time steps used in the improved diffusion Monte Carlo algorithm. Furthermore, there is much greater branching in Green function Monte Carlo resulting in the inefficiency of having many closely related, i.e., statistically dependent, walkers. For the systems studied here, this resulted in an additional factor of 1.5 to 2 greater statistical error, or a factor of 3 to 4 greater computer time for a given statistical error. This factor by itself is compensated for by the fact that in diffusion Monte Carlo one needs to perform calculations at several values of τ to get an accurate extrapolation to zero time step. In other words, owing to the increased time step, the improved diffusion Monte Carlo algorithm is a few hundred times more efficient than Green function Monte Carlo. If we took into account the additional overhead of the more complicated Green function Monte Carlo algorithm, coming from the fact that in Green function Monte Carlo intermediate points are sampled (at which wave functions and their gradients need to be evaluated), that do not contribute to the expectation values, the balance would tip even further in favor of the improved diffusion Monte Carlo algorithm. This is not to say that the domain Green function Monte Carlo algorithm could not be made more competitive by using a better approximate Green function from which to construct the exact Green function. The comparison is merely for the present state of the art.

Ceperley⁸ has developed a more general form of the Green function Monte Carlo method that does not require the motion of the electrons to be within their 'domains' and has used a better approximate Green function from which to construct the exact Green function. Hence it is possible to take larger time steps. However, in this method the electrons may attempt to cross nodes. Hence the time-step error is not totally eliminated, though it is probably very small. The effective time step used in Ref. 8 for Li₂ is 0.043 H⁻¹ which is considerably larger than the optimal time step in domain GFMC but a factor of 6 smaller than the largest time step we used in the improved diffusion Monte

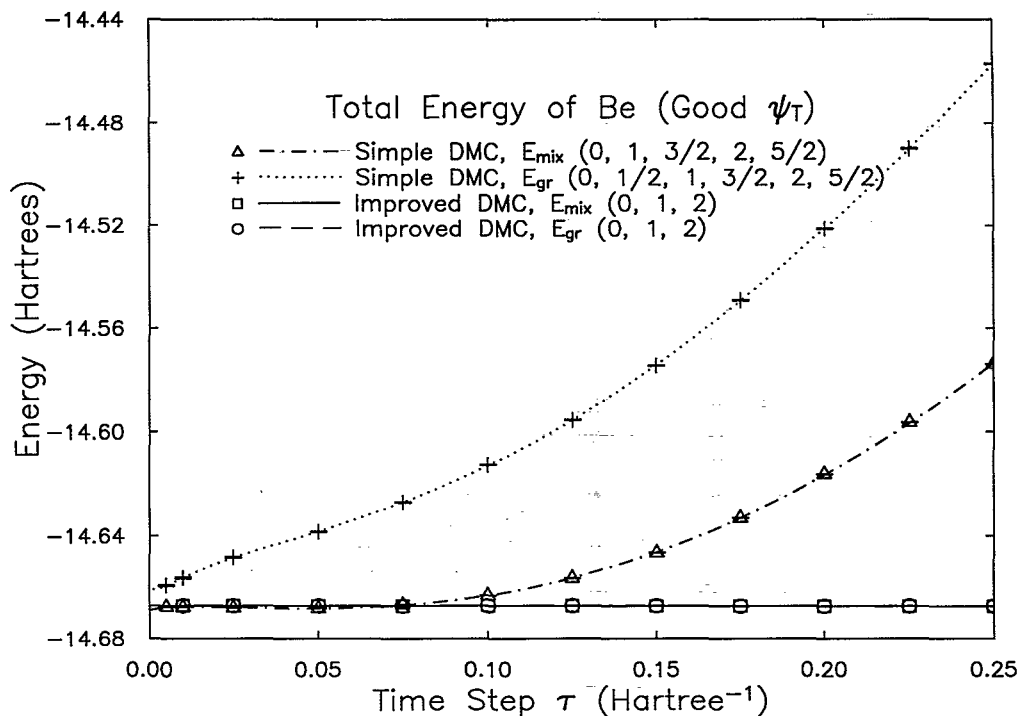


FIG. 12. Same as Fig. 3 but for the good Be wave function.

Carlo method. In common with domain Green function Monte Carlo, Ceperley's method also suffers from the enhanced branching and the overhead of constructing an exact Green function from an approximate one.

C. Good wave function comparisons

So far we have presented results for simple wave functions in order that the advantage of using a better wave function should, if at all, lie with the other algorithms to which we compare ours. We now show that when the improved algorithm is used with good wave functions it is possible to obtain some of the most accurate results to date. Figures 12 and 13 are plots of the total energy as a function of the time step for an accurate Be wave function. The time step error for either algorithm is, as expected, much smaller than for the same algorithm using the simple wave function. As shown in Table II, at $\tau = 0.2 \text{ H}^{-1}$, the time-step error of E_{mix} and E_{gr} in the improved algorithm is a factor of 310 and 1100 smaller than that of simple diffusion Monte Carlo. It should be noted that the simple algorithm has a time-step error in the growth estimator that is larger than that in the mixed estimator. This becomes increasingly noticeable as ψ_T is improved and is a direct reflection of the fact that the growth estimator lacks a zero-variance principle in the case of the simple algorithm because walkers that cross nodes are killed. Table II also shows that the errors in E_{kin} , $\langle r^2 \rangle$ and $\langle r^{-1} \rangle$ change very little for the simple algorithm in going from the simple to the good wave function, but the errors of the improved algorithm are greatly reduced. This is a reflection of the fact that the improved algorithm yields the correct distribution in the

limit that the trial wave function approaches the true wave function, but the simple algorithm does not.

We note from Table II that the value of the energy, extrapolated to $\tau=0$, is 0.0103 H lower for the good wave function than for the simple one. This demonstrates that the nodes of single configuration simple wave functions for Be are significantly in error. Those of the two-configuration good wave function, which takes into account the near-degeneracy correlation of the electrons, are very good. The $\tau=0$ value of the energy for the good wave function is $-14.66718 \pm 0.00003 \text{ H}$ which is $0.00021 \pm 0.00003 \text{ H}$ lower than the best energy calculated directly to date using a state-of-the-art 650 000 determinant multiconfiguration Hartree-Fock wave function,⁴³ and only $0.00019 \pm 0.00006 \text{ H}$ higher than Olsen and Sundholm's⁴³ extrapolated energy, which is obtained by a double extrapolation to infinite single-particle and multiparticle bases size. This result is remarkable, and is a consequence both of the improved diffusion Monte Carlo algorithm and of the high quality of the wave function, despite its compactness.

V. DISCUSSION

To summarize, we have identified four important ingredients that make an algorithm with a small time step error. First, it is necessary to construct trial wave functions that satisfy the cusp conditions so as to avoid divergences in the local energy. Second, by incorporating an accept/reject step the algorithm is constructed so as to sample an electron distribution at least as good as the trial wave function $|\psi_T|^2$. Third, we have identified the nonanalyticities in the local energy and the velocity that result in large time-

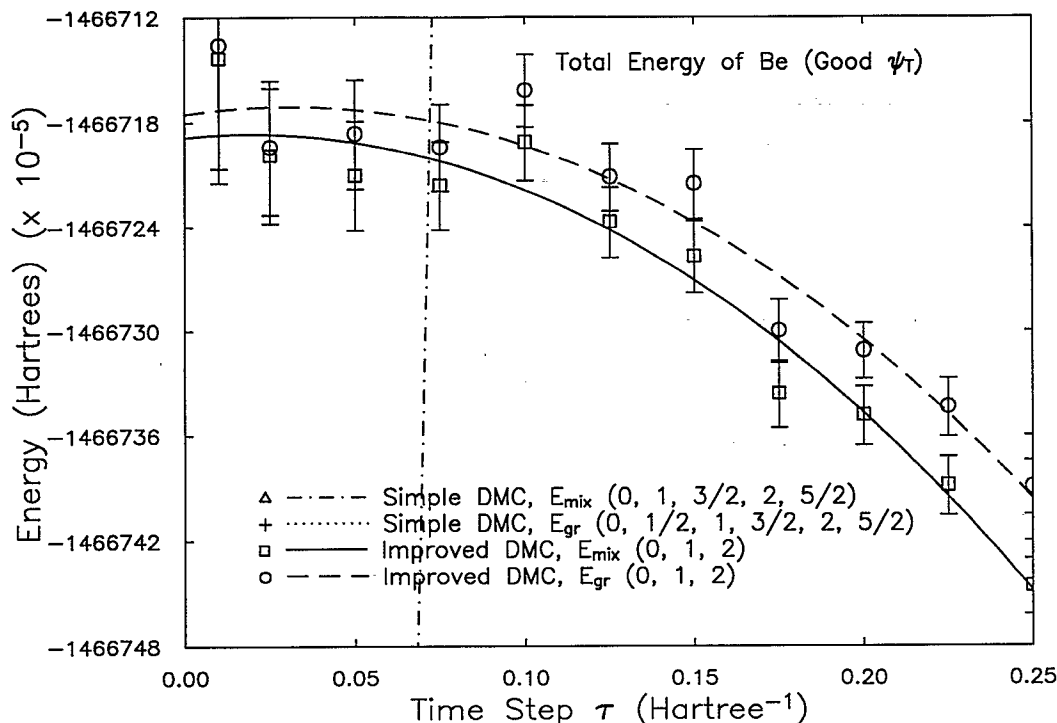


FIG. 13. Same as Fig. 6 but for the good Be wave function.

step errors and we have proposed simple modifications to the approximate Green function that take these into account. These improvements to the Green function not only reduce the time-step error but also enhance the acceptance and consequently the efficiency of the algorithm and reduce greatly the chance of encountering persistent configurations. Fourth, although we find that the improvements to the Green function render the likelihood on encountering persistent configurations negligible, we propose simple modifications of the algorithm that render them impossible.

Further improvements to the algorithm are possible. First, since the acceptance probability for one-electron moves is greater than for multielectron moves, and since updating the wave function after one-electron moves takes less time than after multielectron moves, it may be advantageous, for systems with many electrons, to revert to the original Reynolds *et al.*³ method of doing the accept/reject on each electron, while preserving all the other modifications proposed. Another possibility is to use the Green function of this paper as the approximate known Green function from which the exact Green function is constructed by iterating the Neumann series.⁸ Since the approximate Green function of this paper is a good approximation to the true Green function, it may be possible to both use large time steps and have a rapidly convergent Neumann series. Such an algorithm would still have a non-zero time-step error, as does Ceperley's algorithm,⁸ because some walkers would attempt to cross nodes, but they would be even smaller than the errors presented in this paper.

Although the examples we have chosen are from the electronic structure of atoms and molecules, algorithms of other applications of diffusion Monte Carlo can be improved similarly. In fact in some cases, these improvements will be even more important than in the applications described here. For example, in simulations of Lennard-Jones systems, because of the hard-core repulsion of the Lennard-Jones particles, the velocity is very large if two particles approach each other, while the local energy diverges strongly. Building these nonanalyticities into the Green function and into the short-distance behavior of the trial function⁴⁴ would greatly improve the efficiency of the simulation.

ACKNOWLEDGMENTS

We thank Malvin Kalos and David Ceperley for useful discussions, Gerard Barkema, David Freeman, and Peter Reynolds for useful comments on the manuscript and Kevin Schmidt for generously making his domain Green function Monte Carlo program available to us. The computations were performed on IBM RS-6000 workstations provided by the Cornell Theory Center and by the Cornell-IBM Computing for Scientific Research Joint Study. This work was supported by the Office of Naval Research. The work at the University of Rhode Island was also supported by NSF Grants Nos. DMR-9214669 and CHE-9203498.

APPENDIX A: SUMMARY OF ALGORITHM

In this appendix we summarize the improved algorithm to facilitate its implementation by others. Comments in the algorithm will be indicated as follows: (* ... comment ... *).

1. For $\alpha=1$ to N_0 generate by variational Monte Carlo independent walkers $(\mathbf{R}_\alpha, w_\alpha)$ with $w_\alpha=1$ and $\Psi_T(\mathbf{R}) > 0$.
2. Use the algorithm given below to obtain τ_{eff} and to equilibrate the walkers. This defines the walkers at $t=0$, $E_T(0)$, and $S(0)$.
3. Reset to 0 values of the cumulative variables used to compute expectation values. (Do not reset weights w or Π .)
 $\mathcal{A}_{\text{sum}} = 0$ and $a' = 0$.
4. For times $\hat{t}=1$ to T do (* Do a run of length T . *)
5. ($g=1/\tau$)
 $E_T(\hat{t}) = E_{\text{mix}}(\hat{t}-1, T_p) - \frac{1}{g\tau_{\text{eff}}} \log \frac{W_{\hat{t}-1}}{N_0}$.
[* cf. Eq. (11). $E_{\text{mix}}(0, T_p)$ is taken from equilibration runs *].
6. $\Pi(\hat{t}, T_p) = \prod_{m=0}^{T_p-1} e^{-\tau_{\text{eff}} E_T(\hat{t}-m)}$. [* cf. Eq. (13). Values of E_T for $\hat{t} \leq m$ are taken from equilibration runs *].
7. For each walker $\alpha=1$ to N_t do
8. $G(\mathbf{R}, \mathbf{R}') = G(\mathbf{R}', \mathbf{R}) = 1$.
9. [* Evaluate $G(\mathbf{R}', \mathbf{R})$ for forward move. *]
For each electron $i=1$ to n do
10. (* Drift and diffuse. Each symbol below referring to electron i should have an index i . For simplicity of notation such indices have been suppressed. *)
 $\mathbf{v} = \frac{\partial \psi_T(\mathbf{R})}{\psi_T(\mathbf{R})}$.
11. Find \mathbf{r}_{nuc} the position of the nucleus nearest to \mathbf{r} , and the nuclear charge Z .
12. $\xi = \sqrt{Z^2 + \frac{1}{\tau}}$.
13. (* Evaluate $\bar{\mathbf{v}}$ and decompose $\bar{\mathbf{v}}$ using cylindrical coordinates. *)
 $\mathbf{z} = \mathbf{r} - \mathbf{r}_{\text{nuc}}$; $\hat{\mathbf{z}} = \frac{\mathbf{z}}{z}$
 $a = \frac{1}{2}(1 + \hat{\mathbf{v}} \cdot \hat{\mathbf{z}}) + \frac{Z^2 z^2}{10(4 + Z^2 z^2)}$;
 $\bar{\mathbf{v}} = (-1 + \sqrt{1 + 2av^2\tau})\bar{\mathbf{v}}/(av^2\tau)$
[* cf. Eq. (36) *].
15. $\bar{\mathbf{v}} = \bar{v}_z \hat{\mathbf{z}} + \bar{v}_\rho \hat{\rho}$.
16. $z'' = \max(z + \bar{v}_z \tau, 0)$.
17. $\rho'' = \frac{2\bar{v}_\rho \tau z''}{z + z''}$.

18. [* The previous seven steps define a function $\mathbf{d}(\mathbf{r})$, the position drifted to from an arbitrary initial position \mathbf{r} . *]
 $\mathbf{d}(\mathbf{r}) = \mathbf{r}'' = \mathbf{r}_{\text{nuc}} + \rho'' \hat{\rho} + z'' \hat{\mathbf{z}}$.
19. $\tilde{q} = 1 - \tilde{p} = \frac{1}{2} \operatorname{erfc} \left\{ \frac{z + \bar{v}_z \tau}{\sqrt{2\tau}} \right\}$
[* cf. Eq. (40) *].
20. With probability \tilde{p} do
21. Sample ξ from $g_1(\xi) = (2\pi\tau)^{-3/2} e^{-\xi^2/2\tau}$ (* e.g., with the Box-Muller method *).
22. $\mathbf{r}' = \mathbf{r}'' + \xi$.
23. Else
24. Sample ξ from $g_2(\xi) = \frac{\xi^3}{\pi} e^{-2\xi|\xi|}$ (* for instance see Ref. 45 *).
25. $\mathbf{r}' = \mathbf{r}_{\text{nuc}} + \xi$.
26. End do.
27. $G(\mathbf{R}', \mathbf{R}) := G(\mathbf{R}', \mathbf{R})$
 $\times [\tilde{p} g_1(\mathbf{r}' - \mathbf{d}(\mathbf{r})) + \tilde{q} g_2(\mathbf{r}' - \mathbf{r}_{\text{nuc}})]$.
28. End loop over electrons i started at step 9.
29. Evaluate $\psi_T(\mathbf{R}')$ and $E_L(\mathbf{R}')$ and $\mathbf{V}(\mathbf{R}')$. If $\psi_T(\mathbf{R}) < 0$ set $p=1-q=0$ and go to step 36.
30. [* Evaluate $G(\mathbf{R}, \mathbf{R}')$ for reverse move. *]
For each electron $i=1$ to n do
31. Perform steps 10 through 18 starting from \mathbf{r}' rather than \mathbf{r} to evaluate \mathbf{r}'_{nuc} (the position of the nucleus nearest to \mathbf{r}') and $\mathbf{d}(\mathbf{r}')$.
32. $1 - \tilde{p}' = \tilde{q}' = \frac{1}{2} \operatorname{erfc} \left\{ \frac{z' + \bar{v}'_z \tau}{\sqrt{2\tau}} \right\}$.
33. $G(\mathbf{R}, \mathbf{R}') := G(\mathbf{R}, \mathbf{R}') [\tilde{p}' g_1(\mathbf{r} - \mathbf{d}(\mathbf{r}')) + \tilde{q}' g_2(\mathbf{r} - \mathbf{r}'_{\text{nuc}})]$.
34. End loop over electrons i started at 30.
35. (* Compute accept/reject probability. *)
 $p = 1 - q$
 $= \min \left(1, 1^{\max(0, a' - 50)} \frac{|\psi_T(\mathbf{R}')|^2 \tilde{G}(\mathbf{R}, \mathbf{R}', \tau)}{|\psi_T(\mathbf{R})|^2 \tilde{G}(\mathbf{R}', \mathbf{R}, \tau)}, 1 \right)$.
36. (* Reweight walker. *)
 $\bar{S}(\mathbf{R}', \hat{t}) = \left\{ (E_T(\hat{t}) - E_{\text{mix}}) + [E_{\text{mix}} - E_L(\mathbf{R}')] \right.$
 $\left. \times \frac{\bar{V}(\mathbf{R}')}{V(\mathbf{R}')} \right\}$
 $\bar{S}(\mathbf{R}, \hat{t}) = \left\{ (E_T(\hat{t}) - E_{\text{mix}}) + [E_{\text{mix}} - E_L(\mathbf{R})] \right.$
 $\left. \times \frac{\bar{V}(\mathbf{R})}{V(\mathbf{R})} \right\}$ [* cf. Eq. (39) *].

$$37. \quad \Delta w = \exp \left[\left[\frac{p}{2} \{ \bar{S}(\mathbf{R}', \hat{t}) + \bar{S}(\mathbf{R}, \hat{t}) \} + q \bar{S}(\mathbf{R}, \hat{t}) \right] \tau_{\text{eff}} \right] \quad [* \text{cf. Eq. (25)} *].$$

$$38. \quad w := w \Delta w.$$

$$39. \quad (* \text{ Update expectation values of all observables. } *) \\ \mathcal{A}_{\text{sum}} := \mathcal{A}_{\text{sum}} + \Pi(\hat{t}, T_p) w \{ p \mathcal{A}(\mathbf{R}') + q \mathcal{A}(\mathbf{R}) \}.$$

$$40. \quad (* \text{ Accept or reject move. } *) \\ a' := a' + 1, \quad \mathbf{R}(\hat{t}) = \mathbf{R}'(\hat{t}), \quad \mathbf{V}(\hat{t}) = \mathbf{V}'(\hat{t}), \quad \text{etc.,} \\ \text{with probability } p. \\ a' = 0, \quad \mathbf{R}(\hat{t}) = \mathbf{R}(\hat{t}-1), \quad \mathbf{V}(\hat{t}) = \mathbf{V}(\hat{t}-1), \quad \text{etc.,} \\ \text{with probability } q.$$

$$41. \quad \text{End loop over walkers } \alpha \text{ started at step 7.}$$

$$42. \quad \text{Split/join walkers as described in text.}$$

$$43. \quad \text{End loop over times } t \text{ started at step 4.}$$

APPENDIX B: TRIAL WAVE FUNCTIONS

The quality of the trial wave functions plays an important role in comparing the performance of the algorithms discussed in this paper. For completeness we therefore give in this appendix the explicit form of the trial wave functions used. For detailed justification of the functional forms we refer the reader to earlier work.³⁶⁻³⁸

The simple n -electron wave functions we used are of the form commonly employed in the literature:

$$\psi = \sum_{n=1}^{N_{\text{det}}} (d_n D_n^\dagger D_n) \prod_{i < j} J(r_{ij}),$$

where D^\dagger (D) are the up (down) spin Slater determinants, $J(r_{ij}) = \{\exp(a_{ij} r_{ij} / (1 + b r_{ij}))\}$ and r_{ij} is the inter-electron distance of electrons i and j . This form consists of a product of an antisymmetric part (consisting of a sum of N_{det} determinants) and a symmetric Jastrow part (involving a product over all pairs of electrons); although the notation does not make this explicit the Jastrow part is different for like and unlike spins. The N_{orb} orbitals in the determinants are themselves linear combinations of products of N_{basis} Slater functions and normalized real spherical harmonics

$$\phi_\alpha(\mathbf{r}) = \sum_{\beta=1}^{N_{\text{basis}}} C_{\alpha\beta} N_\beta r^{n_\beta-1} e^{-\zeta_\beta r} Y_{l_\beta m_\beta}(\hat{r}), \quad (\text{B1})$$

TABLE IV. Simple Be wave function. $N_{\text{det}} = 1$, $N_{\text{basis}} = 3$, and $N_{\text{orb}} = 2$. Jastrow parameter $b = 1.0383$.

	Basis functions		
	1s	1s	2s
ζ_β	4.743989	3.365966	1.096756
$C_{1\beta}$	0.509325	1	0
$C_{2\beta}$	0.094609	0	1

where $N_\beta = ((2\zeta_\beta)^{2n_\beta+1} / (2n_\beta)!)^{1/2}$ is the normalization constant of the radial part of the basis function. The basis functions, N_{basis} in number, are identified by their n_β and l_β values (e.g., $n_\beta = 1$, $l_\beta = 0$ in agreement with standard usage is denoted as a 1s function); complete identification requires in addition ζ_β and m_β . The electron-electron cusp condition³⁵ implies that the Jastrow a_{ij} equal 1/2 for pairs of electrons with antiparallel spins and 1/4 for parallel-spin pairs. The determinants D^\dagger (D) are functionally identical and are of order $n/2$. The values of the parameters ζ_β , $C_{\alpha\beta}$, d_n and b are given in Tables IV–VII.

The Jastrow function of the simple wave functions correlates pairs of electrons whereas the Jastrow function of the good wave function correlates pairs of electrons and a nucleus.³⁶⁻³⁸ It is a function of three variables, r_i , r_j , r_{ij} , the distance to the nucleus of electrons i and j , and the distance between these electrons. The function is written in terms of scaled variables $R_i = r_i(1 - e^{-\kappa r_i}) / \kappa r_i$ and additional variables S , T , U , and $R = \sqrt{R_i^2 + R_j^2}$ where $S = R_i + R_j$, $T = R_i - R_j$, $U = r_{ij}(1 - e^{-\kappa r_{ij}}) / \kappa r_{ij}$. It has the form

$$J(r_i, r_j, r_{ij}) = \exp \left(\frac{P(\{a\}, S, T, U) + Q(\{a\}, S, T, U)}{1 + P(\{b\}, S, T, U) + Q(\{b\}, S, T, U)} \right), \quad (\text{B2})$$

where again the spin-dependence is understood and made explicit in Table VIII. Here $P(\{a\}, S, T, U)$ is a complete 4th order polynomial in S , T , U and

$$Q(\{a\}, S, T, U) = (a_{35}U + a_{36}S)R + a_{37}U^3/R + (a_{38} + a_{39}U) \times (R^2 - U^2) \log R^2. \quad (\text{B3})$$

The terms in Q are motivated by the Fock expansion⁴⁶ and serve to reduce the magnitude of the finite discontinuity in

TABLE V. Simple Li_2 wave function. $N_{\text{det}} = 1$, $N_{\text{basis}} = 8$, and $N_{\text{orb}} = 3$. The first four basis functions are centered on the first atom and the second four on the second atom. Internuclear separation is $5.051 a_0$. Jastrow parameter $b = 0.821683$.

	Basis functions							
	1s	1s	2s	2p _z	1s	1s	2s	2p _z
ζ_β	3.579103	2.338523	0.707563	0.532615	3.579103	2.338523	0.707563	0.532615
$C_{1\beta}$	0.606630	1	0	0.061592	0.606630	1	0	-0.061592
$C_{2\beta}$	0.603086	1	0	0.002946	-0.603086	-1	0	0.002946
$C_{3\beta}$	0.104957	0	1	0.305729	0.104957	0	1	-0.305729

TABLE VI. Simple Ne wave function. $N_{\text{det}} = 1$, $N_{\text{basis}} = 10$, and $N_{\text{orb}} = 5$. Jastrow parameter $b = 2.454995$. Basis functions 7 and 8, and orbital 4 are the same as basis functions 5 and 6, and orbital 3 with p_x replaced by p_y . The same holds for basis functions 9 and 10, and orbital 5 with p_x replaced by p_z . Parameters not given explicitly in the table are defined by $\zeta_5 = \zeta_7 = \zeta_9$ and $\zeta_6 = \zeta_8 = \zeta_{10}$; all nontabulated C_{ab} vanish, except $C_{35} = C_{47} = C_{59}$ and $C_{36} = C_{48} = C_{5,10}$.

	Basis functions					
	1s	1s	2s	2s	2p _x	2p _x
ζ_j	10.694072	8.410602	4.368380	2.820366	5.459124	2.413540
C_{1j}	1	0.628185	-0.003506	0	0	0
C_{2j}	0	-0.545029	0.582661	1	0	0
C_{3j}	0	0	0	0	0.430262	1

TABLE VII. Antisymmetric part of the good 2-configuration (4-determinant) Be wave function. $N_{\text{det}} = 4$, $N_{\text{basis}} = 6$, and $N_{\text{orb}} = 8$. Orbitals 1 and 2 comprise the first determinant, orbitals 3 and 4 comprise the second determinant. The third and fourth determinants are identical to the second determinant with p_x replaced by p_y and p_z , respectively. Determinant 1 has a coefficient of 1 and determinants 2, 3, 4 have coefficient -0.13854052. (Note that the coefficients are for unnormalized determinants.) The Jastrow part of the wave function is given in Table VIII.

	Basis functions			
	1s	1s	2s	2p _x
ζ_j	3.19558419	2.98632518	1.01884335	1.00267375
C_{1j}	1	0.00394241	0	0
C_{2j}	0	-0.56216281	1	0
C_{3j}	1	0.00007078	0.00688704	0
C_{4j}	0	0	0	1

TABLE VIII. Jastrow part of the good 2-configuration (4-determinant) Be wave function. $\kappa = 0.34975$. Coefficients that are zero by symmetry have been omitted; hence the missing rows.

	Term	a^{11}	$a^{1\bar{1}}$	b^{11}	$b^{1\bar{1}}$
1	U	0.5	0.25	0.76601265	0.15740663
2	S	-0.32174049	-0.16167748	1.19899131	2.52528576
4	U^2	-0.21045945	-0.52058647	-1.33114784	-1.35904505
5	S^2	-0.47543025	-0.03961320	-2.23269739	-2.99137524
6	T^2	0.35599892	0.31816662	0.66142506	1.10603549
7	US	0.79354593	0.76705142	3.12932039	2.68709399
10	U^3	-0.58102451	-0.64373415	-0.70800536	-0.69092478
11	S^3	0.27774564	0.02890483	0.57532420	0.51324051
13	U^2S	1.15309740	0.65819572	2.05533492	2.44485355
14	US^2	-0.97233708	-0.26631618	-0.81533750	-0.65662838
16	UT^2	-0.22800272	-0.04673077	0.22890273	-0.07731331
18	ST^2	0.20655924	0.12141523	0.42943719	0.73491736
20	U^4	-0.23278323	-0.03592145	-0.62929393	-0.21125656
21	S^4	-0.09443317	-0.03069166	-0.02204179	-0.01084392
22	T^4	0.09478121	0.06408161	0.10065836	0.13288769
23	U^3S	0.71679511	0.43474543	0.80843246	0.87773934
24	US^3	0.44601457	0.18215904	-0.00843236	0.12032247
31	UST^2	0.32921730	0.11925596	-0.32792418	-0.61325175
32	U^2S^2	-0.85492061	-0.47289798	-0.34844499	-0.70648420
33	U^2T^2	-0.12507010	-0.12123822	0.20593344	0.12914268
34	S^2T^2	-0.18247064	0.02862570	0.13438414	0.14765475
35	UR	-0.62590436	-0.22546956	-2.71459521	-3.58148361
36	SR	0.44919268	-0.26189060	1.01996374	2.72970787
37	U^3/R	0.13707051	0.21943996	-0.05980694	-0.94453732
38	$(R^2 - U^2)\log R^2$	-0.08096158	-0.05173070	0.00131255	0.02686620
39	$U(R^2 - U^2)\log R^2$	-0.05582019	-0.00268818	-0.09655479	0.10040772

the local energy in the limit that two electrons approach a nucleus.⁴⁷ Symmetry demands that terms containing odd powers of T have zero coefficients; hence these terms are omitted in Table VIII.

The good Be wave function has 114 parameters, $1d_n$, $4c_{\alpha\beta}$'s, $4\xi_\beta$'s, 1κ , $52a$'s, and $52b$'s. In order to satisfy the cusp conditions these coefficients must satisfy 50 equations in the coefficients $\{a\}, \{b\}$. Four of these equations depend on the $c_{\alpha\beta}$'s, and ξ_β 's. Although these 50 conditions can be imposed exactly, in the wave function presented here only five are satisfied exactly and the remaining are imposed approximately by a penalty function. Hence 109 parameters are varied in the optimization. We have also constructed several wave functions, that have only one-sixth to one-third as many parameters, that are almost but not quite as good as the wave function presented here.

The tables below give the values of the parameters of the wave functions, namely $C_{\alpha\beta}$, ξ_β and b for the simple wave functions or $C_{\alpha\beta}$, ξ_β , d_n , κ , $\{a\}$ and $\{b\}$ for the good wave function. d_1 can without loss of generality be chosen to be unity.

APPENDIX C: ALTERNATIVE TO GENERALIZED METROPOLIS

In this appendix we present an alternative to the generalized Metropolis algorithm for exactly sampling any known distribution. This alternative is useful in the context of the diffusion Monte Carlo algorithm, where the walkers carry statistical weights. In that case the accept/reject step can be replaced by a reweighting step.

Let $f(\mathbf{R})$ be any known distribution that we wish to sample. One possibility is provided by the generalized²⁶⁻²⁸ Metropolis algorithm. Here we discuss an alternative in which the usual accept/reject step is replaced by a weighted, unconditional acceptance.

Let $G(\mathbf{R}', \mathbf{R})$ be a stochastic kernel that is used to propose moves from \mathbf{R} to \mathbf{R}' . (A stochastic kernel is one that is non-negative everywhere and whose integral over the first argument is one.) Let $A(\mathbf{R}', \mathbf{R})$ be the reweighting factor for moves from \mathbf{R} to \mathbf{R}' . We wish to find $A(\mathbf{R}', \mathbf{R})$ such that the equilibrium distribution of this process is $f(\mathbf{R})$, i.e., $f(\mathbf{R}') = \int d\mathbf{R} A(\mathbf{R}', \mathbf{R}) G(\mathbf{R}', \mathbf{R}) f(\mathbf{R})$. From the fact that $G(\mathbf{R}', \mathbf{R})$ is a stochastic kernel it follows that a possible choice is

$$A(\mathbf{R}', \mathbf{R}) = \frac{f(\mathbf{R}') G(\mathbf{R}, \mathbf{R}')}{f(\mathbf{R}) G(\mathbf{R}', \mathbf{R})} \quad (\text{C1})$$

Note that $A(\mathbf{R}', \mathbf{R})$ in Eq. (C1) will exceed one for some choices of \mathbf{R} and \mathbf{R}' , it can only be interpreted as a reweighting factor, not as a probability. This alternative to the generalized Metropolis method has the advantage all moves are accepted, but the disadvantage that there is an additional source of fluctuations of the weights of the walkers.

It is interesting to note that although the total kernel $A(\mathbf{R}', \mathbf{R}) G(\mathbf{R}', \mathbf{R})$ yields the correct distribution $f(\mathbf{R})$, it is not stochastic and does not satisfy detailed balance. On the other hand, if we choose $A(\mathbf{R}', \mathbf{R})$ to be the square root of

the right hand side of Eq. (C1) then detailed balance is satisfied but this kernel does not yield the desired distribution $f(\mathbf{R})$. In case the reader is surprised by this result, we note that to generate a given distribution detailed balance is not a necessary condition, and it is a sufficient condition only when the kernel is stochastic.

APPENDIX D: TIME-STEP ERRORS IN SIMPLE DIFFUSION MONTE CARLO

We argue in this appendix that the act of killing walkers that cross nodes in the simple diffusion Monte Carlo algorithm leads to a $\tau^{1/2}$ term in the time-step error of the growth estimator of the energy E_{gr} . We also show that the nonanalyticity of the local energy and the velocity near nodes does not alter the order of the time-step error in either the growth or the mixed estimator [the leading term is $\mathcal{O}(\tau)$] if the energy is cut off as in Eq. (9).

As in Section III C we let R_1 denote the distance in configuration space to the nearest node of ψ_T . Let f_τ be the distribution of the simple diffusion Monte Carlo random walk with time step τ , so that f_0 is the limit of no time-step error. One suspects that the error $\delta f_\tau = f_\tau - f_0$ induced by the nonanalyticities of the local energy and of the velocity near a node is unimportant beyond some healing distance $\lambda \sim \sqrt{\tau}$ away from the node. Although the exact distribution behaves quadratically near the node, $f_0(R_1) \approx c_0 R_1^2 + \dots$, the simple diffusion Monte Carlo distribution has a constant density of walkers at the node: $f_\tau(R_1) \approx a_\tau + b_\tau R_1 + c_\tau R_1^2 + \dots$. The exact distribution behaves quadratically near the node so that the coefficients a_τ and b_τ must vanish as $\tau \rightarrow 0$ to retrieve the exact distribution. We note in passing that for the improved algorithm a_τ and b_τ are rigorously zero for all τ as a result of the accept/reject step.

How might we model this τ dependence of the coefficients for the simple algorithm? One way is to argue that the $a_\tau + b_\tau R_1$ contribution must somehow "match onto" the correct $c_0 R_1^2$ behavior in the vicinity of the healing distance. Thus $a_\tau + b_\tau \lambda \sim c_0 \lambda^2$, from which one reads off $a_\tau \sim \lambda^2 \sim \tau$ and $b_\tau \sim \lambda \sim \sqrt{\tau}$. Numerically exact results obtained for a one-dimensional lattice model indicate that the simple arguments presented here are correct so long as the local energy does not diverge in the vicinity of the nodes.⁴⁸

Let us first consider the error induced by the node for the mixed energy estimator. In Eq. (8) the divergence of the local energy E_L is cut off to $2/\sqrt{\tau}$, thus

$$\delta E_{\text{mix}} \sim \int_0^\lambda \frac{2}{\sqrt{\tau}} \delta f_\tau(R_1) dR_1 \sim \lambda^2 \sim \tau. \quad (\text{D1})$$

Hence the nonanalyticities in the nodal region do not decrease the order of time-step error in E_{mix} from the $\mathcal{O}(\tau)$ error already present in the simple diffusion Monte Carlo algorithm. Similar arguments also apply to the growth estimator.

We note that on the other hand, if Eq. (8) is not used to cut off the divergence in the local energy, the above estimate is modified to

$$\delta E_{\text{mix}} \sim \int_0^\lambda \frac{\delta f_\tau(R_\perp)}{R_\perp} dR_\perp, \quad (\text{D2})$$

which is logarithmically divergent at any nonzero τ . If the local energy at a point infinitesimally on one side of a nodal surface diverges to $+\infty$, the local energy at a point infinitesimally on the other side diverges to $-\infty$. Hence the logarithmic divergences cancel, but the infinite variance renders the algorithm unusable.

Next we show that if walkers that cross nodes are killed then the growth estimator E_{gr} has an $\mathcal{O}(\sqrt{\tau})$ time-step error. The population at time $t+1$ is in error due to killing walkers that cross nodes in the simple algorithm. The number of such walkers near the node is estimated as

$$\delta N_{t+1} \sim N_t \int_0^\lambda \delta f_\tau(R_\perp) dR_\perp \sim N_t \lambda^3 \sim N_t \tau^{3/2}. \quad (\text{D3})$$

The growth estimator is basically

$$E_{\text{gr}} \approx \frac{1}{\tau} \ln(N_{t+1}/N_t) \quad (\text{D4})$$

(somewhat more precisely, it is the ratio of the sum of the walker weights w_α , but this form is correct to leading order). So if we now say N_{t+1} is "in error" by the above estimate for δN_{t+1} , we find, after expanding the logarithm, that:

$$\delta E_{\text{gr}} \sim -\frac{1}{\tau} \frac{\delta N_{t+1}}{N_t} \sim \tau^{1/2}. \quad (\text{D5})$$

Thus the convergence of the growth estimator is slower than the mixed. Our empirical observations discussed in Section IV confirm the two power laws for the two estimators. The $\tau^{1/2}$ convergence of E_{gr} makes accurate extrapolation to $\tau=0$ very difficult.

¹J.B. Anderson, J. Chem. Phys. **65**, 4121 (1976).

²J.W. Moskowitz, K.E. Schmidt, M.A. Lee, and M.H. Kalos, J. Chem. Phys. **77**, 349 (1982).

³P.J. Reynolds, D.M. Ceperley, B.J. Alder, and W.A. Lester, J. Chem. Phys. **77**, 5593 (1982).

⁴D.M. Ceperley and M.H. Kalos, in *Monte Carlo Method in Statistical Physics*, edited by K. Binder, Topics in Current Physics, Vol. 7 (Springer, Berlin, Heidelberg, 1979), Chap. 4; K.E. Schmidt and M.H. Kalos, in *Applications of the Monte Carlo Method in Statistical Physics*, edited by K. Binder, Topics in Current Physics, Vol. 36 (Springer, Berlin, Heidelberg, 1984), Chap. 4.

⁵B.H. Wells, in *Methods of Computational Chemistry*, Vol. 1, edited by S. Wilson (Wiley, New York, 1987).

⁶M. Kalos, D. Leveque, and L. Verlet, Phys. Rev. A **9**, 2178 (1974).

⁷K.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).

⁸D. Ceperley, J. Comp. Phys. **51**, 404 (1983).

⁹D.M. Ceperley and B.J. Alder, J. Chem. Phys. **81**, 5833 (1984).

¹⁰R. Grimm and R.G. Storer, J. Comp. Phys. **4**, 230 (1969); **7**, 134 (1971); **9**, 538 (1972).

¹¹J.B. Anderson, J. Chem. Phys. **63**, 1499 (1975).

¹²J. Vrbik, J. Phys. A **18**, 1327 (1985).

¹³J. Vrbik and S.M. Rothstein, J. Comp. Phys. **63**, 130 (1986).

¹⁴S.M. Rothstein, N. Patil, and J. Vrbik, J. Comp. Chem. **8**, 412 (1987).

¹⁵M.F. DePasquale, S.M. Rothstein, and J. Vrbik, J. Chem. Phys. **89**, 3629 (1988).

¹⁶S.A. Chin, Phys. Rev. A **42**, 6991 (1990).

¹⁷C.J. Umrigar, K.J. Runge, and M.P. Nightingale, in *Monte Carlo Methods in Theoretical Physics*, edited by S. Caracciolo and A. Fabrocini (Editrice, Pisa, 1990).

¹⁸S. Zhang and M.H. Kalos, Phys. Rev. Lett. **67**, 3074 (1991).

¹⁹J.B. Anderson, C.A. Traynor, and B.M. Boghosian, J. Chem. Phys. **95**, 7418 (1991).

²⁰D.R. Garmer and J.B. Anderson, J. Chem. Phys. **86**, 4025 (1987).

²¹D.R. Garmer and J.B. Anderson, J. Chem. Phys. **89**, 3050 (1988).

²²Note that if \tilde{f} is the solution to Eq. (1) with $E_T=0$ then $f = \exp[\int_0^t E_T(t') dt'] \tilde{f}$ is the solution for time-dependent $E_T(t)$.

²³M.P. Nightingale and H. Blöte, Phys. Rev. B **33**, 659 (1986); Phys. Rev. Lett. **60**, 1562 (1988).

²⁴In the simple diffusion Monte Carlo algorithm used here, we limit the magnitude of the three-dimensional velocity of each electron to $1/\tau$, rather than limiting each component of the velocity. The difference is unimportant.

²⁵J.H. Hetherington, Phys. Rev. A **30**, 2713 (1984).

²⁶W.K. Hastings, Biometrika **57**, 97 (1970).

²⁷D. Ceperley, G.V. Chester, and M.H. Kalos, Phys. Rev. B **16**, 3081 (1977).

²⁸M.H. Kalos and P.A. Whitlock, *Monte Carlo Methods*, Vol. 1 (Wiley, New York, 1986).

²⁹N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.M. Teller, and E. Teller, J. Chem. Phys. **21**, 1087 (1953).

³⁰D.R. Garmer, Ph.D. thesis, 1987.

³¹Although in Refs. 3, 32, and 5 walkers that cross nodes are killed, in later work the moves are rejected without killing the walkers. P.J. Reynolds and D.M. Ceperley (private communication).

³²R.N. Barnett, P.J. Reynolds, and W.A. Lester, J. Chem. Phys. **82**, 2700 (1985).

³³S.M. Rothstein (private communication).

³⁴R.N. Barnett, P.J. Reynolds, and W.A. Lester, J. Chem. Phys. **96**, 2141 (1992); Int. J. Quantum Chem. **42**, 837 (1992).

³⁵T. Kato, Comm. Pure Appl. Math. **10**, 151 (1957).

³⁶C.J. Umrigar, K.G. Wilson, and J.W. Wilkins, Rev. Lett. **60**, 1719 (1988).

³⁷C.J. Umrigar, K.G. Wilson, and J.W. Wilkins, in *Computer Simulation Studies in Condensed Matter Physics: Recent Developments*, edited by D.P. Landau, K.K. Mon, and H.B. Schüttler, Springer Proc. Phys. (Springer, Berlin, 1988).

³⁸C.J. Umrigar, Int. J. Quantum Chem. **23**, 217 (1989).

³⁹C.J. Umrigar (unpublished).

⁴⁰F. Mentch and J.B. Anderson, J. Chem. Phys. **80**, 2675 (1984).

⁴¹R.N. Barnett, P.J. Reynolds, and W.A. Lester, J. Chem. Phys. **84**, 4992 (1986).

⁴²Z. Sun, S. Huang, R.N. Barnett, and W.A. Lester, J. Chem. Phys. **93**, 3326 (1990).

⁴³J. Olsen and D. Sundholm (unpublished), as reported in Ann-Marie Mårtensson-Pendrill *et al.*, Phys. Rev. A **43**, 3355 (1991).

⁴⁴A. Mushinski and M.P. Nightingale (unpublished).

⁴⁵C.J. Everett and E.D. Cashwell, *A Third Monte Carlo Sampler*, Los Alamos Technical Report No LA-9721-MS, 1983.

⁴⁶V. Fock, Izv. Akad. Nauk. SSSR, Ser. Fiz. **18**, 161 (1954); English translation: D. Kngl. Norske Videnskab. Selsk. Forh. **31**, 138, 145 (1958).

⁴⁷C.R. Myers, C.J. Umrigar, J.P. Sethna, and J.D. Morgan, Phys. Rev. A **44**, 5537 (1991).

⁴⁸Claudia Filippi (unpublished).