

Remedy for the fermion sign problem in the diffusion Monte Carlo method for few fermions with antisymmetric diffusion process

Yuriy Mishchenko

Cold Spring Harbor Laboratory, Cold Spring Harbor, New York 11743, USA

(Received 15 March 2005; published 22 February 2006)

We suggest an exact approach to help remedy the fermion sign problem in diffusion quantum Monte Carlo simulations. The approach is based on an explicit suppression of symmetric modes in the Schrödinger equation by means of a modified stochastic diffusion process (antisymmetric diffusion process). We introduce this algorithm and illustrate it on potential models in one dimension (1D) and show that there it solves the fermion sign problem exactly and converges to the lowest antisymmetric state of the system. Then, we discuss extensions of this approach to many-dimensional systems on examples of quantum oscillator in 2D–20D and a toy model of three and four fermions on harmonic strings in 2D and 3D. We show that in all these cases our method shows a performance comparable to that of a fixed-node approximation with an exact node.

DOI: 10.1103/PhysRevE.73.026706

PACS number(s): 02.70.Ss, 67.90.+z, 71.45.Gm, 82.90.+j

I. INTRODUCTION

The Schrödinger equation is an accepted way of description of nonrelativistic microscopic systems:

$$i\hbar \frac{\partial}{\partial t} \Psi(\mathbf{R}, t) = (H - E_B) \Psi(\mathbf{R}, t); \quad (1)$$

here, t is a real-time variable, $\mathbf{R} = \{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N\}$ is a $3N$ -dimensional vector of positions of N particles, $H = -(\hbar^2/2m)\nabla^2 + V(\mathbf{R})$ is the system's Hamiltonian, and E_B is an energy offset. The Schrödinger equation yields the description of the quantum system in terms of a set of eigenenergies and wave functions. Naturally, exact solutions of Eq. (1) may be obtained only in a few special cases; in general, one needs to use an approximate scheme to find solutions. In recent years increasing attention has been drawn to the random-walk approach for solving the Schrödinger equation, the so-called quantum Monte Carlo (QMC) method, the attractiveness of which lies in the fact that it treats the many-body problem exactly. The QMC method is a projection method based on the combination of the imaginary-time Schrödinger equation, generalized stochastic diffusion process, and Monte Carlo integration. The solutions it yields have only statistical error which can be properly estimated and, in principle, made as small as desired [1,2].

The imaginary-time Schrödinger equation is Eq. (1) in which the time variable t is replaced with its imaginary analog $\tau = it$:

$$-\hbar \partial_\tau \Psi(\mathbf{R}, \tau) = (H - E_B) \Psi(\mathbf{R}, \tau). \quad (2)$$

It is known that for any initial condition $\Psi(\mathbf{R}, 0)$ the solution of Eq. (2) can be written in terms of the eigenfunctions Ψ_n and the eigenvalues E_n of the stationary Schrödinger equation as follows:

$$\Psi(\mathbf{R}, \tau) = \sum_n a_n e^{-(E_n - E_B)\tau} \Psi_n(\mathbf{R}), \quad (3)$$

$$E_n \Psi_n(\mathbf{R}) = H \Psi_n(\mathbf{R}). \quad (4)$$

Here a_n are fixed by $\Psi(\mathbf{R}, \tau) = \Psi(\mathbf{R}, 0)$ at $\tau = 0$. When $\tau \rightarrow \infty$, only the smallest E_n term in expansion (3) will survive, thus projecting an arbitrary initial condition onto the ground state of the Schrödinger equation $\Psi_0(\mathbf{R})$. This feature is widely used to numerically obtain the ground state by solving Eq. (2) with $\tau \rightarrow \infty$. Still, solving numerically Eq. (2) in a large number of dimensions is a challenging task: quantum-chemistry or condensed-matter applications may require solving Eq. (2) with hundreds of coordinates.

As is known, help is found in the theory of stochastic processes. Consider, e.g., Eq. (2) with $V(\mathbf{R}) = 0$. Then it reduces to the diffusion equation in $3N$ dimensions,

$$-\partial_\tau \Psi(\mathbf{R}, \tau) = -\frac{1}{2} \sum_{i=1}^N \nabla_i^2 \Psi(\mathbf{R}, \tau). \quad (5)$$

Here we used Planck units $\hbar = c = 1$ and unit mass $m = 1$. This is the *master equation for a stochastic diffusion process* [3] that describes the evolution of the population density $\Psi(\mathbf{R}, \tau)$ for an ensemble of particles subject to Brownian motion—i.e., the probability for each particle to move from position \mathbf{x} to position \mathbf{x}' in time τ is proportional to $e^{-(\mathbf{x} - \mathbf{x}')^2/2\tau}$. Since the population density for such an ensemble satisfies

$$\Psi(\mathbf{R}, \tau + \Delta\tau) = \int \frac{d\mathbf{R}'}{(2\pi\Delta\tau)^{-3N/2}} \exp\left(-\frac{(\mathbf{R}' - \mathbf{R})^2}{2\Delta\tau}\right) \Psi(\mathbf{R}', \tau), \quad (6)$$

which is the solution of Eq. (5) in terms of the Green's function

$$G_d(\mathbf{R}' \rightarrow \mathbf{R}; \tau) = (2\pi\tau)^{-3N/2} \exp\left(-\frac{(\mathbf{R}' - \mathbf{R})^2}{2\tau}\right), \quad (7)$$

the solution of the imaginary-time Schrödinger equation (5) can be represented by a diffusive ensemble of particles (walkers). If this ensemble is let to evolve for a sufficiently long time, eventually it will sample the ground-state wave

function $\Psi_0(\mathbf{R})$. This sample can be used to estimate, for example, an integral of the ground-state wave function with an arbitrary weight $O(\mathbf{R})$ via Monte Carlo integration $\int d\mathbf{R} O(\mathbf{R}) \Psi_0(\mathbf{R}) = \sum O(\mathbf{R}_i) / \sum 1$.

In the case of a Hamiltonian with $V(\mathbf{R}) \neq 0$, Eq. (2) still may be interpreted as the master equation for a *generalized diffusion process* governed by the Green's function of the full equation (2). Naturally, except for a few special cases, the exact Green's function can not be known and one has to use an approximation. In the diffusion Monte Carlo (DMC) method commonly used is the *small- τ* approximation [4], i.e., for $\tau \rightarrow 0$,

$$G(\mathbf{R}' \rightarrow \mathbf{R}; \tau) \approx e^{-\tau[V(\mathbf{R})-E_B]/2} G_d(\mathbf{R}' \rightarrow \mathbf{R}; \tau) e^{-\tau[V(\mathbf{R}')-E_B]/2} \quad (8)$$

accurate up to $O(\tau^3)$. Then, the evolution of the ensemble can be broken into a sequence of short time steps, each described by Eq. (8). The factor

$$P = \exp\{-\tau[V(\mathbf{R}') + V(\mathbf{R}) - 2E_B]/2\} \quad (9)$$

acts as a position-time-dependent renormalization of the diffusion Green's function. This normalization is usually accounted for by a branching birth-death algorithm: if $P < 1$, the walker is destroyed with probability $1-P$; if $P > 1$, an additional walker is created at the same position with probability $P-1$. Both cases can be conveniently coded with the number of walkers allowed to continue evolving at position \mathbf{R} given by $\text{INT}(P + \text{rand}(0, 1))$.

Since in the DMC method the wave function has to be a population density, the DMC method can only describe the constant-sign solutions of the Schrödinger equation. This presents a serious problem if one is interested in the ground state of a fermion system where the wave function is antisymmetric (i.e., both positive and negative), the situation known as the “fermion sign problem.” At present, the most successful approach here is the so-called fixed-node approximation. Here one assumes *a priori* knowledge of the nodal hypersurface $\Psi_0(\mathbf{R})=0$ [5,6]. In the volume embraced by the nodal surface the wave function has a constant sign and can be found to high accuracy using the usual DMC method. The fixed-node approximation is known to give the above (variational) estimate for the fermion ground-state energy. Very remarkable results have been achieved in this way for systems with as many as few hundreds of electrons [1]. Yet the proper choice of the nodal surface presents a very nontrivial challenge and exact approaches to the fermion sign problem are still of great interest.

One of the steps in this direction is to represent the fermion wave function as a difference of two positive contributions, e.g.,

$$\Psi(\mathbf{R}) = \Psi^+(\mathbf{R}) - \Psi^-(\mathbf{R}),$$

and use two sets of “positive-sign” and “negative-sign” walkers to sample Ψ^+ and Ψ^- separately. If one starts with an appropriately chosen initial “antisymmetric” sample, theoretically, Ψ^+ and Ψ^- will converge to the fermion ground state $\Psi = \Psi^+ - \Psi^-$ [7]. In such an approach, known as the transient estimator method, it can be seen rather immediately

that both subsets of walkers independently converge to a symmetric solution. Information about the antisymmetric contribution is only encoded as an $e^{-(E_1-E_0)\tau}$ variation on top of a large symmetric component [see Eq. (3)]. Even if the coefficient a_0 in Eq. (3) was originally exactly zero, if left without control, noise results in $a_0 \neq 0$ and washes out the fermionic signal, the situation known as the “exponential signal-to-noise ratio problem.” One further possibility would be to suppress the symmetric component of noise explicitly at each DMC step using certain cancellation criteria between the positive and negative walkers [8–10]. For that one needs to realize that, strictly speaking, contributions coming to a point at each moment of time from Ψ^+ and Ψ^- shall cancel each other. Then one attempts to implement this cancellation by making walkers of positive and negative signs to annihilate whenever they end up at the same position \mathbf{R} . Since it is practically impossible for two pointlike walkers to end up at the same position, implementation of this process makes use of “smearing” for which a certain smearing scale shall be introduced. Normally, the annihilation of positive and negative walkers is performed via cancellation between the Green's functions for some “+–” pair of walkers, e.g.,

$$G(\mathbf{R}; \mathbf{R}^+, \mathbf{R}^-) = G(\mathbf{R}, \mathbf{R}^+) - G(\mathbf{R}, \mathbf{R}^-), \quad (10)$$

and, thus, the scale is $\sqrt{\tau}$. Here \mathbf{R}^+ and \mathbf{R}^- are positions of some positive- and negative-sign walkers in the sample. Although impressive results have been achieved with this approach for smaller molecules [11], it was realized soon that for the cancellation to work efficiently the average distance between the walkers should be comparable to the annihilation scale. For example, if the diffusion scale $\sqrt{\tau}$ becomes much smaller than the average distance between the walkers, the overlap in Eq. (10) will almost surely vanish and the two sets of positive and negative walkers will evolve independently. The necessity to maintain a rather large density of walkers makes the computational cost grow exponentially with the number of dimensions and renders solution of the problems with more than about 30 dimensions practically unfeasible [8,12].

In this paper we propose a different exact approach to the fermion sign problem which we call the antisymmetric diffusion Monte Carlo (ADMC) method. It is based on modification of the stochastic diffusion process as is described by

$$H_D = -\frac{1}{2}\nabla^2 \rightarrow H_a = -\frac{1}{2}\nabla^2 + \mathcal{A}P_+, \quad (11)$$

where P_+ is a projector on the linear space of all symmetric quantum states and \mathcal{A} is a large positive constant. As we show in this paper, such a modification leads to a suppression of symmetric modes in the DMC dynamics so that the lowest antisymmetric state becomes the true ground state of the stochastic process.

In the next section, Sec. II, we introduce the *stochastic antisymmetric diffusion process* and show how it can be adopted for use with the DMC method. In Sec. III we test this method numerically on a few potential models in one dimension (1D), harmonic oscillators in 1D–20D, and the problem of three and four fermions bound by harmonic

strings to a fixed center in 2D and 3D (which can be seen as a crude model for three- and four-electron atoms). All of these models have known analytic solutions. As we shall see, our method gives results comparable to that of a fixed-node approximation with an exact node in all of these examples. In Sec. III we also discuss in detail ADMC implementation in many dimensions. Conclusions follow in Sec. IV. In the Appendix we present the source code (MATLAB) for the ADMC method.

II. STOCHASTIC ANTISYMMETRIC DIFFUSION PROCESS

The antisymmetric diffusion Monte Carlo method stems from the notion that the ground state of the Schrödinger equation need not be necessarily symmetric, contrary to widespread opinion. Consider, e.g., the following modification of the Hamiltonian:

$$H = -\frac{1}{2}\nabla^2 + V(x) \rightarrow H_a = -\frac{1}{2}\nabla^2 + V(x) + \mathcal{A}(1 + \hat{\pi}), \quad (12)$$

where $\hat{\pi}\Psi(x) = \Psi(-x)$ and \mathcal{A} is a large positive constant. If $[\hat{\pi}, V(x)] = 0$, it is known that the eigenfunctions of H are also the eigenfunctions of $\hat{\pi}$ with eigenvalues ± 1 ; thus, they are also the eigenfunctions of H_a and vice versa. Then, it is easy to see that the result of transformation (12) is to shift the eigenvalues of parity-even (Ψ^+) and parity-odd (Ψ^-) solutions as follows:

$$\begin{aligned} \Psi^+ : E^+ \Psi^+ &= H \Psi^+ \rightarrow (E^+ + 2\mathcal{A}) \Psi^+ = H_a \Psi^+, \\ \Psi^- : E^- \Psi^- &= H \Psi^- \rightarrow E^- \Psi^- = H_a \Psi^-, \end{aligned}$$

thus making negative-parity eigenstates have lower energy than their positive-parity counterparts.

Note that potential in H_a is nonlocal. While it is difficult to represent a nonlocal potential in the DMC method via the branching process, the term $\mathcal{A}(1 + \hat{\pi})$ can be exactly accommodated at the level of the diffusion Green's function itself by substituting it with the solution of Eq. (13). E.g., in 1D a solution of Eq. (13) can be immediately found:

$$\begin{aligned} -\frac{\partial}{\partial \tau} G_a(x, x'; \tau) &= -\frac{1}{2} \partial_x^2 G_a(x, x'; \tau) + \mathcal{A}(1 + \hat{\pi}_x) G_a(x, x'; \tau), \\ G_a(x, x'; 0) &= \delta(x - x'). \end{aligned} \quad (13)$$

In momentum space we have

$$-\frac{\partial G_a(k, y; \tau)}{\partial \tau} = \frac{k^2 G_a(k, y; \tau)}{2} + \mathcal{A}[G_a(k, y; \tau) + G_a(-k, y; \tau)]. \quad (14)$$

Since parity is conserved in Eq. (14), defining

$$G_a(k, y; \tau) = G_a^+(k, y; \tau) + G_a^-(k, y; \tau),$$

$$G_a^+(k, y; 0) = \cos ky, \quad G_a^-(k, y; 0) = -i \sin ky,$$

we find

$$G_a(k, y; \tau) = e^{-(k^2/2 + \mathcal{A})\tau} \cos ky - i e^{-k^2\tau/2} \sin ky. \quad (15)$$

After an inverse Fourier transform we obtain

$$\begin{aligned} G_a(x, y; \tau) &= \frac{1}{2\sqrt{2\pi\tau}} [e^{-(x-y)^2/2\tau} (1 + e^{-2\mathcal{A}\tau}) \\ &\quad - e^{-(x+y)^2/2\tau} (1 - e^{-2\mathcal{A}\tau})]. \end{aligned} \quad (16)$$

In general, if the projector onto irrelevant states in Eq. (12) $[P_a = (1 + \hat{\pi})/2]$ is such that $[P_a, \nabla^2] = 0$, then the modified diffusion propagator is

$$G_a(\mathbf{R}, \mathbf{R}'; \tau) = (1 - P_a + e^{-2\mathcal{A}\tau} P_a) G_d(\mathbf{R}, \mathbf{R}'; \tau). \quad (17)$$

Note that $e^{-2\mathcal{A}\tau}$ is important to render $G_a(\mathbf{R}, \mathbf{R}'; \tau) \rightarrow \delta(\mathbf{R} - \mathbf{R}')$ as $\tau \rightarrow 0$. However, for any practical purposes this term may be dropped. For the DMC method it is sufficient to use the modified diffusion Green's function

$$G_a(x, y; \tau) = \frac{1}{\sqrt{2\pi\tau}} \frac{e^{-(x-y)^2/2\tau} - e^{-(x+y)^2/2\tau}}{2}. \quad (18)$$

It is straightforward to check that Eq. (18) is a proper propagation function—i.e.,

$$\int dy G_a(x, y; \tau_1) G_a(y, z; \tau_2) = G_a(x, z; \tau_1 + \tau_2),$$

except for τ_1 or $\tau_2 = 0$. For a many-body system Eq. (18) generalizes,

$$G_a(\mathbf{R}' \rightarrow \mathbf{R}; \tau) = \text{Assym}_{\mathbf{R}} G_d(\mathbf{R}, \mathbf{R}'; \tau) \equiv \sum_{\sigma} \varepsilon_{\sigma} G_d(\sigma \mathbf{R}, \mathbf{R}'; \tau), \quad (19)$$

where the sum is over particle permutations σ and $\varepsilon_{\sigma} = 1$ for even and -1 for odd permutations and antisymmetrization operator *assym* is defined by the second identity in Eq. (19). Then, the fermionic ground state of the Schrödinger equation can be found using the DMC method with diffusion process substituted with antisymmetric diffusion which suppresses symmetric solutions and reduces the exponential signal-to-noise ratio problem.

Note that the idea of suppressing the irrelevant components of the wave function by introducing into the diffusion Hamiltonian a “large” projector onto such states may also be applied to finding excited states by considering, e.g., $H \rightarrow H + \mathcal{A} P_{\psi}$, where $\hat{P}_{\psi} = |\psi\rangle\langle\psi|$ is simply a projector onto the ground state $|\psi\rangle$.

III. NUMERICAL RESULTS

In this section we numerically test our idea on a few potential models in 1D, harmonic oscillator in 1D–20D, and the problem of three and four fermions on harmonic strings in 2D–3D. For the ADMC method we use an implementation in MATLAB designed “on top” of a straightforward realization of the DMC method and described in detail in the Appendix. In the numerical procedure $|G_a(x, y; \tau)|$ defines the probability for a walker to move from position x to position y in time τ

and $\text{sgn}[G_a(x,y;\tau)]$ is used to keep track of the walkers' signs. The local potential $V(x)$ is accounted for via the regular branching process (9). One should also remember that walkers may be destroyed during antisymmetric diffusion itself [$\int dy |G_a(x,y;\tau)| < 1$] which is an important addition to the extinction from the branching process in 1D.

In 1D, since for any values of y the $G_a(x,y;\tau)$ has the node at the same position $x=0$, $\text{sgn}[G_a(x,y;\tau)]$ divides the configuration space into “positive” and “negative” halves. If the initial sample was completely localized in one-half and each walker in this half had, e.g., a positive sign, then whenever such a walker is moved so that it does not cross the node its sign will remain positive. Shall such a walker move into the other half of the space, it will cross the node and its sign will change. If this walker happens to move back into the original half, its sign will change again to the original positive. Thus, the change of signs induced by $\text{sgn}[G_a(x,y;\tau)]$ always occurs coherently in 1D. This simplifies greatly the fermion sign problem because no cancellation is necessary, but this is the exception limited to $D=1$.

We now consider numerically the ADMC method in a finite quantum well, harmonic oscillator, and linear potential in 1D, each of which has a known analytical solution. In each case we found that the ADMC method converges to the fermion ground state. For a finite well with depth $V=10$ units the energy levels are $E^+=2.3$ and $E^-=8.0$ units. Using an implementation in MATLAB we obtain $E^+=2.3\pm 0.1$ and $E^-=8.1\pm 0.1$. For the harmonic oscillator $V(x)=x^2/2$ the energy levels are $E^+=0.5$ and $E^-=1.5$; we obtain $E^+=0.5\pm 0.1$ and $E^-=1.5\pm 0.1$. And for the linear potential $V(x)=|x|$ the energy levels are $E^+=0.8$ and $E^-=1.9$ while we find $E^+=0.81\pm 0.1$ and $E^-=1.85\pm 0.1$. In all cases we find that the population of walkers properly samples fermionic $|\Psi(x)|$ (see Fig. 1). Thus, in 1D the fermion sign problem is solved exactly.

Note that in our calculations we aimed at verifying that the ADMC approximation converges to a fermionic ground state and not to reproduce with high accuracy the numbers known analytically for these models. Thus we quote rather large uncertainties. This is because the computation typically stopped soon after it was clear that it converges to the correct energy. Precision of the MC method, as is known, may be increased by taking larger samples or longer runs.

It is important to emphasize that 1D examples, instructive and illustrative as they are, do not guarantee applicability of a method to many-dimensional problems. This is because of a very special case of 1D for fermion systems—namely, that the node in 1D is a point and not an extended object. In fact, previously known *ab initio* fermion sign prescriptions, including the above-mentioned cancellation schemes, work well in 1D while are impractical in higher dimensions because of escalating computational cost. For this reason we now turn our attention to applications of the ADMC method in higher dimensions.

In higher dimensions the special situation of 1D does not hold and a cancellation between “positive” and “negative” walkers is necessary. Generally, we found that the usual cancellation schemes, such as those reviewed in Sec. I, still fail with the ADMC method due to the low density of walkers.

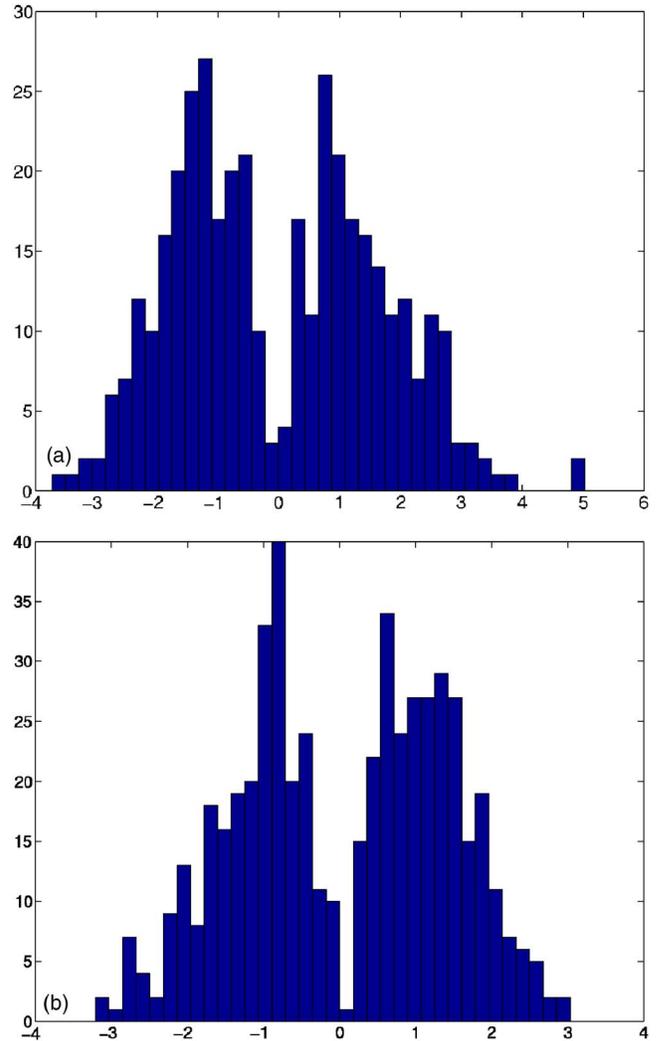


FIG. 1. (Color online) The distribution of walkers for a quantum harmonic oscillator (a) and linear potential (b) for an antisymmetric solution in 1D obtained with the antisymmetric diffusion process.

However, we found that a different type of cancellation similar in spirit to the fixed-node approximation is successful with the ADMC method and is able to achieve performance comparable to that of the fixed-node approximation with an exact node. In such cancellation schemes no nodal surface is assumed *a priori*, but it is established dynamically by the algorithm itself so as to balance the fluxes of positive and negative walkers. It has been known that with the regular DMC method such approaches are unstable because fluctuations in the population sizes of the ensembles of positive- and negative-sign walkers lead to an exponentially growing imbalance. This results in that one sample completely takes over while the other is extinguished and the entire system relaxes to the bosonic ground state. With the ADMC method this does not happen because the symmetry and balance between positive- and negative-sign ensembles are strictly enforced by antisymmetric diffusion or, in other words, because the stable ground state of the antisymmetric diffusion process is fermionic.

Although we considered a variety of possible prescriptions for such a cancellation, here we mention only two and

in detail describe only one of them that we believe to have performed the best. In the first implementation the configuration space is partitioned into cells identified with a set of randomly chosen center points. A point of the configuration space is said to belong to the cell identified with the center in the set that is the closest to this point. This arrangement is also known as the Voronoi diagram. The wave function within each cell is assumed to have a constant sign. This divides the configuration space into pockets of positive and negative wave function signs so that the walkers of a given sign are not allowed to penetrate into the wave function pockets having the opposite sign. The signs of all cells are updated to track the average sign of all walkers entered each cell in the past. This allows the actual partition of the configuration space to be established dynamically.

In the second implementation, the sign of the wave function at the location of a particular walker is estimated based on the average sign of a number of nearest-neighbor walkers. E.g., if a particular walker \mathbf{R} has n positive neighbors and $K-n$ negative neighbors, the sign of the wave function at \mathbf{R} is assumed to be positive if $n > K-n$ and negative otherwise. Correspondingly, the walker is eliminated if it is found to reside in the wave function region with the wrong sign. The node of the wave function in this case is represented by the interface between the ensembles of walkers with positive and negative signs. No *a priori* information about the node position is needed, including any predefined parametrization of the nodal surface or of the configuration-space partition. The actual position of the node is established dynamically in a stochastic manner constrained to a degree by the initial sample distribution. E.g., our initial condition was typically given by a sample of walkers randomly distributed in the unit D -cube $[0, 1]^D$. Thus, the node typically was established as a plane nearly perpendicular to the vector $(1, 1, \dots, 1)$. Still, there was a substantial degree of noise in the position of the node due to stochastic nature of the process by which it was established.

In our simulations with quantum oscillators in 1D-20D we typically used $M=500-1000$ walkers with $\Delta\tau \approx 10^{-3}$ and simulation duration of $\approx 10^4$ Monte Carlo steps. With this choice of parameters the distance-based cancellation failed already in 5D with $E^- = 2.7 \pm 0.05$ (as is well known, for quantum oscillators, $E^+ = D/2 = 2.5$ in $D=5$ and $E^- = D/2 + 1 = 3.5$). On the other hand, fixed-node-like algorithms showed good performance in all our runs. In 2D the ADMC method with nearest-neighbor cancellation gave $E^- = 1.98 \pm 0.05$ vs $E_{fn}^- = 1.95 \pm 0.03$ of a fixed-node approximation (note that in this case the average distance between neighbor walkers in the sample is just $l \approx 0.05$ compared to $\sqrt{\Delta\tau} \approx 0.03$). In 5D we obtained $E^- = 3.53 \pm 0.06$ vs $E_{fn}^- = 3.44 \pm 0.06$ ($l \approx 0.3$); in 10D, $E^- = 6.03 \pm 0.1$ vs $E_{fn}^- = 5.95 \pm 0.1$ ($l \approx 0.6$); and in 15D, $E^- = 8.4 \pm 0.2$ vs $E_{fn}^- = 8.5 \pm 0.2$ ($l \approx 0.8$). The ADMC method with Voronoi partition of the configuration space in 5D gave $E^- = 3.63 \pm 0.1$ and in 20D $E^- = 11.65 \pm 0.3$ vs $E_{fn}^- = 11.2 \pm 0.2$ of the fixed-node approximation. Because of the rough representation of the nodal surface in this implementation, the energy estimate was typically worse here than that obtained with nearest-neighbor cancellation. For number of dimensions $D > 20$ we encountered growing bias in our simple

DMC realization which is known in the literature as population control bias [13]. Our computational resources did not allow us to make an attempt at elimination of this bias. Thus we had been limited to simulations with less than 20 dimensions. However, importance sampling, reweighing schemes, and runs with a larger number of walkers are known to improve the situation here [14].

In our numerical experiments the nearest-neighbor cancellation generally appeared as the best performer. This is an *ab initio* method that needs no *a priori* information about the nodal surface. Among other advantages is that this method can be easily generalized for use with importance sampling (as long as the guiding function is symmetric and nonzero) and that this scheme can be easily adopted for use with the conventional fixed-node DMC algorithm. In many dimensions we found that the effect of walkers' elimination in antisymmetric diffusion itself is deemphasized and can be successfully replaced by the cancellation routine. Thus, modifications to a regular fixed-node DMC method leading to the use of antisymmetric diffusion reduce to a redistribution of walkers between symmetry-related wave function pockets (following a regular diffusion move) and the calculation of the node position using a nearest-neighbor prescription. The DMC-move Green's function and elimination and move-rejection routines may be carried over without change from the regular fixed-node DMC algorithm.

Among the disadvantages of nearest-neighbor cancellation is higher fluctuations in M leading to larger statistical error and larger population control bias in E . Also, a simple mixed estimator can no longer be used to estimate the configuration's energy because it now misses the contribution from the flux of walkers across the node. The nearest-neighbor cancellation computational cost scales with the size of the ensemble as M^2 . This introduces substantial hardship if one attempts to counter population control bias or simply improve DMC precision by increasing the size of the sample. Furthermore, as any other cancellation scheme, this method has a " $N_p!$ problem" consisting in factorially growing a number of symmetry-related pockets (which all need to be properly sampled) when the number of particles in the simulation grows.

Specific to nearest-neighbor cancellation is the problem of finite node width. To get a better understanding of this, consider an arbitrary walker from the sample. Let s_V be a sphere of volume V_K such that there are K other walkers inside. All walkers inside s_V we will call nearest neighbors. Let s_i be a random variable representing the sign of a randomly picked nearest neighbor. Then the sign of the wave function at the position of the walker is represented by

$$S_K = \sum_{i=1}^{i=K} s_i. \quad (20)$$

According to Eq. (20), the partition of the configuration space into the pockets of positive and negative wave function signs is itself random. Even if a positive walker is drawn from the positive wave function sign pocket, it may be found to reside in the wrong region and may be eliminated with probability $P\{S_K < 0\}$. Thus, the nodal surface in nearest-

neighbor cancellation (i.e., the region where walkers are rapidly eliminated) is wide. The finite width of the node leads to a positive bias in the energy estimate in the ADMC method.

To estimate the width of the node consider the following. If $\langle \rho_+(\mathbf{R}) \rangle$ and $\langle \rho_-(\mathbf{R}) \rangle$ are s_V -averaged normalized (i.e., $\langle \rho_+ \rangle + \langle \rho_- \rangle = 1$) densities of positive and negative walkers in the neighborhood of point \mathbf{R} , respectively, and K is large, then the probability to conclude that the wave function at point \mathbf{R} has negative sign is given by

$$P\{S_K < 0\} = \text{erf}\left(\frac{\sqrt{K}\langle \rho_-(\mathbf{R}) \rangle - \langle \rho_+(\mathbf{R}) \rangle}{2\sqrt{\langle \rho_-(\mathbf{R}) \rangle \langle \rho_+(\mathbf{R}) \rangle}}\right) \quad (21)$$

(because in this case S_K may be treated as normally distributed random variable with certain mean and dispersion). Thus, if $\langle \rho_-(\mathbf{x}) \rangle$ changes from 0 distance d away from the node to 1 distance d on the other side of the node, we get

$$P\{S_K < 0\} \approx \text{erf}\left(-\frac{x}{\sqrt{(d^2 - x^2)/K}}\right), \quad (22)$$

where x is the distance from point \mathbf{R} to the node. d scales as the average distance between walkers in the sample $d \sim l \sim L/M^{1/D}$, where L is the extent of the wave function and M is the size of the sample. Thus, the width of the node in nearest-neighbor cancellation is approximately given by

$$d_{\text{node}} \approx \frac{L/M^{1/D}}{\sqrt{1+K}}. \quad (23)$$

When D is large, the width of the node is large. The good news is that it can be efficiently reduced as $1/\sqrt{K}$. In fact, we are able to clearly see this effect in our numerical simulations (see Fig. 2). On the other hand, since to thin the node we effectively integrate the wave function over some volume s_V [i.e., $S_K \approx \int_{s_V} d\mathbf{R} \Psi(\mathbf{R})$], our node is additionally smoothed in the process so that its features smaller than the smoothening volume s_V are lost. This is of limited concern because of very sparse volume sampling in the DMC method that already to a large degree suppresses the effect of such features. Also, note that the linear size of the node features that are smoothed away in nearest-neighbor cancellation practically does not depend on K for large D (i.e., $\sim K^{1/D}$).

Finally, in $D \approx 20-40$ we found that nearest-neighbor cancellation failed to provide an accurate result with the energy appearing to relax to the bosonic ground state. At this point we also started to see exceedingly large population control bias in the DMC subroutine; thus, a connection between the two may be possible. Implementation with the configuration-space partition via the Voronoi diagram showed no signs of such a breakdown; i.e., the energy estimate always stayed significantly above E^+ . At this time we cannot claim complete understanding of the reasons behind this failure. Our numerical experimentation yielded the following. This failure could not be reproduced in 2D with as few as seven walkers; thus, it is not due to a low walker density. The ensembles of positive- and negative-sign walkers appeared to be clearly separated in the configuration space and did not mix. When antisymmetric diffusion resampling was restricted from full D -dimensional reflection to a

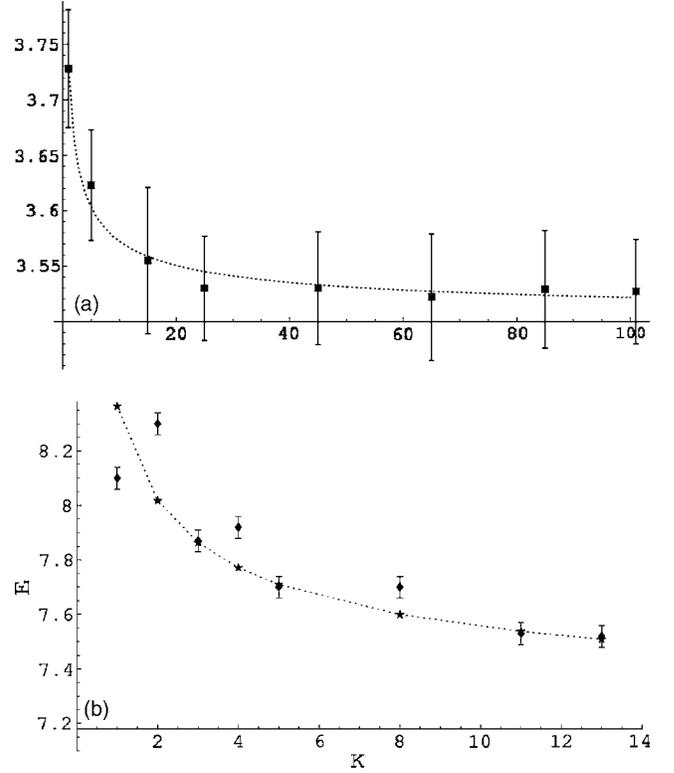


FIG. 2. Finite node width bias in the ADMC method with nearest-neighbor cancellation for a quantum oscillator in 5D (a) and three fermions in 3D (b). Dotted lines are $E = a + b/\sqrt{K}$ fits. The best fits are $E = 3.498 + 0.245/\sqrt{K}$ for the harmonic oscillator and $E = 7.18 + 1.18/\sqrt{K}$ for three fermions.

change in only one or two coordinates, the effect completely disappeared (i.e., the result of a fixed-node approximation was recovered in all dimensions). Finally, the interface between the ensembles of positive and negative walkers appeared to rotate. It may be possible that a breakdown of the ADMC method with nearest-neighbor cancellation is because the process enters some dynamical regime different from the one considered above and because some high-dimensional mode develops that is able to lower the energy of the sample while keeping it antisymmetric.

However, we believe that the observed effect is mainly due to a clustering of the sample into dense statistically dependent groups of walkers, known to be an issue in higher-dimensional DMC simulations. This proposition is consistent with all of the above made observations: the absence of the effect in lower dimensions or when symmetry reflection is reduced to one or two dimensions because such dense clusters effectively annihilate with themselves when they are reflected to a relatively nearby position, interface rotation because of penetration of such clusters (and their mirror images in the opposite direction) through the nearest-neighbor node. If this proposition is correct, the use of importance sampling will greatly reduce this condition. Also, it may be possible that in simulations with few relatively low-dimensional fermions (2D–3D) this issue may be absent to begin with given our above-mentioned experiments with restricted symmetry reflection.

Finally, we discuss the applications of the ADMC method to a toy model of three fermions on harmonic strings bound to a common center. These represent a case closer to realistic in terms of the complexity of the nodal surface. Recall that in harmonic oscillators even in 2D the node is a simple plane. As was mentioned, a difficulty with the larger number of particles is factorially growing the number of symmetry-related pockets that all need to be considered for proper cancellation. E.g., in the case of three particles there are $3!=6$ wave function pockets which need to be properly sampled. Thus, our simulations performed with 300 walkers are equivalent to a fixed-node simulation with only $300/3!=50$ walkers per wave function pocket. With such a low number of walkers the proper representation of the nodal surface becomes an issue. In particular, the application of the ADMC method with $M=300$ walkers and $K \approx 10$ nearest neighbors to three fermions in 3D yields $E^- = 7.53 \pm 0.03$ vs $E^- = 6.56 \pm 0.03$ for a fixed-node approximation with $M=50$. Compare this with the fermion ground-state energy $E^- = 6.5$ (the ground state for three fermions in 2D and 3D may be obtained as a Slater determinant of quantum oscillator wave functions with quantum numbers 00, 10, and 01, and, thus, has energy $E^- = N_p D/2 + 2$). The ADMC method with $M=900$ and $K=17$ yields $E^- = 7.06 \pm 0.02$. The reason for such a large bias is twofold. On the one side, there is a finite-node-width bias decaying as $1/\sqrt{K}$ (see Fig. 2). On the other side, there is bias due to the nodal surface being represented only by some $M=50$ points in nine-dimensional space. A further increase in M , which would decrease these biases, leads to M^2 rising computational cost.

A simple improvement may be made, however, that alleviates these problems by recognizing that the central ingredient here is to better represent the node and not “intrinsically” to have a larger sample size. If we had a better representation of the node, even $M=300$ walkers would suffice to get a good answer. A better representation of the node may be achieved by using walkers from the previous generations to deduce the position of the node. With such a modification one can simulate $M=300$ walkers and the node represented by almost $10 \times 300/3!$ points (which we call the codebook) at the same computational cost as simulating $M=900$ walkers. This also allows us to use a larger number of nearest neighbors K and, thus, further improve the accuracy of the node representation. The Appendix presents the implementation of the algorithm that uses a nearest-neighbor search in the codebook. After such a modification we find for three fermions in 3D with $M=300$ a codebook with nine pages (i.e., nine previous generations sampled) and $K \approx 60$, $E^- = 6.63 \pm 0.01$. We find, with $K \approx 15$, $E^- = 6.74 \pm 0.04$ and, with $K \approx 5$, $E^- = 6.93 \pm 0.02$. These exhibit perfect $1/\sqrt{K}$ scaling with the best fit $E^-(K) = 6.5 + 0.18/\sqrt{K}$. For three fermions in 2D with $M=300$ and $K \approx 60$ we obtain $E^- = 5.02 \pm 0.01$ vs $E^- = 5.0 \pm 0.03$ fixed-node results with $M=50$. Accordingly, the result for the simple nearest-neighbor algorithm without codebook was $E^- = 5.8 \pm 0.02$ with $M=300$ and $E^- = 5.27 \pm 0.01$ with $M=900$.

With four fermions the situation is further worsened because now there are $4!=24$ symmetry related pockets instead of $3!=6$. With $M=300$ we have just a little more than ten walkers to sample each pocket. Still, using the same param-

eters as above we find $E^- = 10.4 \pm 0.05$. This is to be compared with the exact answer $E^- = 9.0$. Respectively, the fixed-node approximation with $M=15$ gives $E^- = 10.32 \pm 0.04$, but with $M=300$ it gives $E^- = 8.81 \pm 0.03$. We may try to increase the size of the codebook by the factor $4!/3!=4$ to account for the increased number of symmetry duplicates to be considered. Doing so we find with $M=300$, $E^- = 9.6 \pm 0.1$. This, indeed, lowered the energy estimate but not sufficiently to recover the exact answer.

It is impossible to extend our calculations to five fermions using brute force because of the escalating $N_p!$ problem. However, we believe this situation may be eventually resolved by accounting for $N_p!$ permutations implicitly and, keeping only the walkers from the single wave function pocket, account for $N_p!$ implicit mirror copies using the procedure with algebraically growing cost. For example, a search on the matrix of all interparticle distances for two configurations $(\mathbf{r}_1, \dots, \mathbf{r}_{N_p})$ and $(\mathbf{r}'_1, \dots, \mathbf{r}'_{N_p})$ may allow one to efficiently locate the nearest neighbors out of $N_p!$ copies.

IV. CONCLUSIONS

In this work we present an approach to the fermion sign problem which is based on the suppression of the symmetric modes at the level of the Hamiltonian. This makes an antisymmetric solution the true ground state and suppresses the exponential signal-to-noise ratio problem. This is achieved by introducing a nonlocal projection operator $\mathcal{A}(1 + \hat{\pi})$ in the Hamiltonian which shifts the energy of all “bosonic” states by a large positive constant $2A$. Such an alteration, formally, destroys the locality of the Hamiltonian; however, it can be fully incorporated in the redefinition of the stochastic diffusion process. After such a redefinition, the approach can be applied within the framework of the regular DMC method. We tested and illustrated this approach on an example of a few potential models in 1D, quantum oscillator in 1D–20D, and a toy model of three and four fermions on harmonic strings in 2D and 3D. We found that the method performs well in 1D as well as in many-dimensional oscillators and for three and four fermions. With cancellation algorithms similar to the fixed-node approximation but using the dynamic nodal surface the exact fermion DMC method showed a performance comparable to that of the fixed-node approximation with an exact node. Thus, our simulations show that with improvements provided by antisymmetric diffusion and a simple prescription for dynamical determination of the constant wave function sign regions, the exact fermion DMC method may be efficiently applied even in higher dimensions.

One of the advantages of our method is its portability to already existing algorithms. Essentially, two modifications are necessary to use antisymmetric diffusion in an existing fixed-node algorithm. First, after regular diffusion moves are performed, the walkers should be additionally redistributed between the symmetry-related configuration-space regions followed by corresponding sign flips (i.e., if the permutation leading to the region was negative, the sign of the walker should be flipped). Then, the position of the nodal surface should be evaluated given current and previous sample con-

figurations. The thus determined nodal surface can be used in a fixed-node elimination and move-rejection prescription in place of the usual node-crossing condition. Our algorithm also allows straightforward generalization for use with importance sampling as long as the guiding function is symmetric and everywhere nonzero.

In this paper a nearest-neighbor search was introduced in order to facilitate the determination of the nodal surface from the sample configuration. It is simple and has good performance but also showed a breakdown in higher dimensions. Further advances are possible within this framework. The representation of the nodal surface may be improved by accumulating sample configurations from previous generations. In principle, if configurations from all previous times are considered in such a manner, the ADMC method should show performance identical to that of the fixed-node approximation with exact node, and the above-mentioned breakdown should be removed since the node will be stabilized in the limit of large simulation times. To facilitate a nearest-neighbor search in an increasingly large codebook improved algorithms should be used to yield a sign estimate without an exhaustive codebook search. Furthermore, the $N_p!$ problem may be alleviated by considering the set of walkers within a single wave function pocket and accounting for $N_p!$ mirror copies implicitly. In particular, we believe an efficient algorithm may exist yielding for two vectors $\mathbf{R}=(\mathbf{r}_1 \dots \mathbf{r}_N)$ and $\mathbf{R}'=(\mathbf{r}'_1 \dots \mathbf{r}'_N)$ the permutation $\sigma=i_1, \dots, i_N$ minimizing $(\mathbf{R}-\sigma\mathbf{R}')^2$. Using this thus-defined distance function $\rho_\sigma(\mathbf{R}, \mathbf{R}')^2 = \min_{\sigma} (\mathbf{R}-\sigma\mathbf{R}')^2$, the cost of the search over $N_p!$ copies in the sample of walkers may be greatly reduced.

Finally, it may be advantageous to view the above discussion in a wider context. The partition of the configuration space into wave function pockets of constant sign does not have to be restricted to a nearest-neighbor prescription. Instead, any many-dimensional classifier trained on the sequence of sample configurations may be used to define regions of different wave function signs. A nearest-neighbor search is only one example of such a classifier. At the same time a significant body of literature on high-dimensional classifiers already exists in image processing and pattern recognition where one usually relies on neural networks. One may hope to incorporate the conditions of the problem's symmetry in the structure of such neural network and, thus, remove the need for tracking of $N_p!$ copies of each walker. In principle, with K units such a network can provide accuracy for the wave function sign representation $\sim 1/\sqrt{K}$ [15]. Antisymmetric diffusion will provide the stability necessary for the training of the classifier to be successful. Significant advances along this direction seem plausible.

APPENDIX: MATLAB SOURCE CODE FOR IMPLEMENTATION OF THE ADMC METHOD WITH NEAREST-NEIGHBOR CANCELLATION USING A CODEBOOK SEARCH

Here we present the source code for our implementation of the ADMC method with nearest-neighbor cancellation. A codebook over $N_{cdb}=9$ snapshots of previous sample con-

figurations is used to define the nodal hypersurface. Snapshots are taken with a large interval of time $t_{cdb}=0.75$, close to the energy autocorrelation time, to avoid having statistically correlated sample configurations in the codebook.

Also, we noticed that in print some symbols in the code could have changed, such as "''" which is used for MATLAB matrix transposition and string definitions.

```
% ADMC algorithm (MATLAB)
clear all
% -----
% PARAMETERS
D=3; % dimensionality
N=3; % number of fermions
M0=300; % target pop size

dt=0.01; % sim time step
ttot=60; % total sim time
tau=1; % pop size reset time

ineighb=3; % min nearest neighbors
fneighb=751; % max nearest neighbors
pneighb=0.04; % percent nearest neighbors

K0=9; % codebook history depth
tcdb=0.75; % codebook update time

% -----
% AUXILIARY
itr=ceil(ttot/dt);
dK=ceil(tcdb/dt);
dt2=sqrt(dt);
egN=20;
E=0;

% all permutations of N fermions
pP=uint8(perms(N:-1:1)');
% permutations signs of N fermions
sP=perm_sign(pP);

% -----
% INTIALIZING
% statistics array
ss=zeros(4,itr);
% E-smoothing array
mEG= repmat(E, [1, egN]);
% array of walkers coordinates
walkers=rand(D,N,M0);
% array of "walker sign > 0"
signs=false(M0,1);
% codebook for walkers & signs
mK=ceil(1.5*M0);
wkcdb=cell(K0,1);
sgncdb=cell(K0,1);
for k=1:K0
    wkcdb{k}=zeros(D,N,mK);
    sgncdb{k}=zeros(mK,1);
end
% codebook pages age tracker
```

```

agecdb=repmat(K0+2,[1,K0]);
% -----
% CYCLE
for i=1:itr
% current population size
M=size(walkers,3); M1=M;

% -----
% REGULAR DIFFUSION STEP
% compute potential before
v0=squeeze(sum(sum(walkers.^2,1),2))/2;
% perform diffusion step
walkers=walkers+dt2*randn(D,N,M);
% compute potential after
v1=squeeze(sum(sum(walkers.^2,1),2))/2;
% compute multiplicity factor
mult=floor(exp(-dt*(v0+v1)/2)*...
exp(dt*E)+rand(M,1));

% repopulate walkers w/r to multiplicity
idx=zeros(max(mult),M);
for k=1:M idx(1:mult(k),k)=k; end
idx=idx(idx>0); M=length(idx);
walkers=walkers(:, :, idx);
signs=signs(idx);
if (ineighb >=M) error('M is low!'); end

% -----
% A-DIFFUSION RESAMPLING
% choose symmetry pocket to which
% the walkers will be translated
idx=1+floor(size(pP,2)*rand(M,1));
P=uint16(pP(:, idx));
% rearrange N-fermions according to P:
% some tricks for "faster" matlab
Z=uint16(repmat(1:M,[N,1]));
for k=1:D
X=uint16(repmat(k,N,M));
nidx=sub2ind([D,N,M],X(:,P),Z(:));
walkers(k, :, :)=reshape(walkers(nidx),N,M);
end
% sign flip in resampling( "sign > 0")
% 1 x 1 - > 1, 1 x 0 - > 0, 0 x 1 - > 0, 0 x 0 - > 1
% :: operation "=="
signs=(signs==sP(idx));
% -----
% CANCELLATION
% update codebook
if (rem(i,dK)==0)
% find oldest codepage out there
[junk,idx]=max(agecdb); idx=idx(1);
agecdb=agecdb+1;
agecdb(idx)=0;
% fill in target codepage
wlkcdb{idx}=walkers;
sgncdb{idx}=signs;
end
% reshape to N*D coords

wlkdst=reshape(walkers,D*N,M);
% how many pages in cdbook?
idx=find(agecdb < K0+1);
cdbdst=[ ];
cdbsgn=[ ];
for k=idx
mK=size(wlkcdb{k},3);
cdbdst=[cdbdst, ...
reshape(wlkcdb{k},D*N,mK)];
cdbsgn=[cdbsgn;sgncdb{k}];
end
% include current configuration
% of walkers into codebook
if (rem(i,dK)~=0)
cdbdst=[cdbdst, wlkdst];
cdbsgn=[cdbsgn; signs];
end
curK=size(cdbdst,2);
% percentage ## of nearest neighbors
neighb=max(ineighb, ...
min(fneighb, round(pneighb*curK)));
% computes distance all-to-all
a=repmat(sum(wlkdst.^2,1),[1,curK]);
b=repmat(sum(cdbdst.^2,1),[M,1]);
dists=a+b-2*(wlkdst'*cdbdst);
% find nearest neighbours
[junk,nnidx]=sort(dists,2);
% find signs of nearest neighbours
nnsigns=sign(cdbsgn(nnidx(:,1:neighb))-0.5);
% find wave function sign estimates
signsX=sum(nnsigns,2)>0;
idx=find(signsX(:)==signs(:));
M=length(idx);
% clear wrong walkers
walkers=walkers(:, :, idx);
signs=signs(idx);

% -----
% ADJUST ENERGY OFFSET
% mixed energy estimator
EV=mean(sum(sum(walkers.^2,1),2))/2;
% growth estimator smoothed
if (i==1) mEG(:)=EV; end
mEG(1:end-1)=mEG(2:end);
mEG(end)=E-log(M/M1)/dt;
EG=mean(mEG);
% adjust energy offset
E=EG+log(M0/M)/tau;
% save statistics to array ss
ss(:,i)=[M;E;EV;EG];
end

% MEANS AND VARIATIONS
% cut-off initial relaxation
itr0=ceil(max(5,K0*tcdb)/dt);
% mean and STD for E
ssR=mean(ss(:,itr0:end),2);

```

```

ssD=std(ss(:,itr0:end),0,2);
fprintf('E: %g + / - %g; M %g + / - %g\n', ...
    ssR(2), ssD(2), ssR(1), ssD(1));

% function sP=perm_sign(pP)
% %copy this function
% % to file perm_sign.m
% % ...obtains permutations signs
% nPerm=size(pP,2);
% sP=false(nPerm,1);
%
% for k=1:nPerm
% perm=pP(:,k);
% flips=1;

% i=1;
% while(~isempty(i))
% i=find(perm(1:end-1)>perm(2:end));
% if(~isempty(i))
% i=i(1);
% x=perm(i);
% perm(i)=perm(i+1);
% perm(i+1)=x;
% flips=-flips;
% end
% end
% sP(k)=(flips>0);
% end

```

-
- [1] W. Foulkes, L. Mitás, R. Needs, and G. Rajagopal, *Rev. Mod. Phys.* **73**, 33 (2001).
- [2] I. Kosztin, B. Faber, and K. Schulten, *Am. J. Phys.* **64**, 633 (1996).
- [3] S. Karlin and H. Taylor, *A Second Course in Stochastic Processes* (Academic, New York, 1981).
- [4] P. Reynolds, D. Ceperley, B. Alder, and W. Lester, *J. Chem. Phys.* **77**, 5593 (1982).
- [5] J. Anderson, *J. Chem. Phys.* **63**, 1499 (1975).
- [6] J. Moskowitz, K. Schmidt, M. Lee, and M. Kalos, *J. Chem. Phys.* **77**, 349 (1982).
- [7] D. M. Ceperley and B. J. Alder, *Phys. Rev. Lett.* **45**, 566 (1980).
- [8] D. Arnow, M. Kalos, M. Lee, and K. Schmidt, *J. Chem. Phys.* **77**, 5562 (1982).
- [9] M. Kalos and F. Pederive, in *Quantum Monte Carlo Methods in Physics and Chemistry*, edited by M. Nightingale and C. Umrigar, Vol. 525 of *NATO Advanced Study Institute, Series, C: Mathematical and Physical Sciences* (Kluwer Academic, Boston, 1999), p. 263.
- [10] B. Hammond, W. Lesler, Jr., and P. Reynolds, *Monte Carlo Methods in Ab Initio Quantum Chemistry* (World Scientific, Singapore, 1994).
- [11] D. Diedrich and J. Anderson, *Science* **258**, 786 (1992).
- [12] J. Anderson and C. Traynor, *J. Chem. Phys.* **95**, 7418 (1991).
- [13] C. J. Umrigar, M. P. Nightingale, and K. J. Runge, *J. Chem. Phys.* **99**, 2865 (1993); D. M. Ceperley and M. H. Kalos, in *Monte Carlo Methods in Statistical Physics*, 2nd ed., edited by K. Binder (Springer, Berlin, 1979); J. H. Hetherington, *Phys. Rev. A* **30**, 2713 (1984).
- [14] P. J. Reynolds, D. M. Ceperley, B. J. Alder, and W. A. Lester, *J. Chem. Phys.* **77**, 5593 (1982); R. C. Grimm and R. G. Storer, *J. Comput. Phys.* **7**, 134 (1971).
- [15] A. R. Barron, *IEEE Trans. Inf. Theory* **39**, 930 (1993).