An efficient sampling algorithm for variational Monte Carlo

Anthony Scemama,[a) Tony Lelièvre, Gabriel Stoltz, and Eric Cancès
CERMICS and INRIA Project Micmac, Ecole Nationale des Ponts et Chaussées, 6 et 8 Avenue Blaise Pascal, Cité Descartes-Champs sur Marne, 77455 Marne la Vallée Cedex 2, France

Michel Caffarel
Laboratoire de Chimie et Physique Quantiques, CNRS-UMR 5626, IRSAMC, Université Paul Sabatier, 118 route de Narbonne, 31062 Toulouse Cedex, France

We propose a new algorithm for sampling the N-body density $|\Psi(\mathbf{R})|^2/\int_{\mathbb{R}^{3N}}|\Psi|^2$ in the variational Monte Carlo framework. This algorithm is based upon a modified Ricci-Ciccotti discretization of the Langevin dynamics in the phase space $(\mathbf{R},\mathbf{P})$ improved by a Metropolis-Hastings accept/reject step. We show through some representative numerical examples (lithium, fluorine, and copper atoms and phenol molecule) that this algorithm is superior to the standard sampling algorithm based on the biased random walk (importance sampling). © *2006 American Institute of Physics.*
[DOI: 10.1063/1.2354490]

I. INTRODUCTION

Most quantities of interest in quantum physics and chemistry are expectation values of the form

$$\frac{\langle\Psi,\hat{O}\Psi\rangle}{\langle\Psi,\Psi\rangle}, \tag{1}$$

where $\hat{O}$ is the self-adjoint operator (the observable) associated with a physical quantity O and Ψ a given wave function.

For N-body systems in the position representation, Ψ is a function of $3N$ real variables and

$$\frac{\langle\Psi,\hat{O}\Psi\rangle}{\langle\Psi,\Psi\rangle} = \frac{\int_{\mathbb{R}^{3N}}[\hat{O}\Psi](\mathbf{R})\Psi(\mathbf{R})^* d\mathbf{R}}{\int_{\mathbb{R}^{3N}}|\Psi(\mathbf{R})|^2 d\mathbf{R}}. \tag{2}$$

High-dimensional integrals are very difficult to evaluate numerically by standard integration rules. For specific operators $\hat{O}$ and specific wave functions Ψ, e.g., for electronic Hamiltonians and Slater determinants built from Gaussian atomic orbitals, the above integrals can be calculated analytically. In some other special cases, (2) can be rewritten in terms of integrals on lower-dimensional spaces (typically $\mathbb{R}^3$ or $\mathbb{R}^6$).

In the general case, however, the only possible way to evaluate (2) is to resort to stochastic techniques. The variational Monte Carlo (VMC) method[1] consists in remarking that

$$\frac{\langle\Psi,\hat{O}\Psi\rangle}{\langle\Psi,\Psi\rangle} = \frac{\int_{\mathbb{R}^{3N}}O_L(\mathbf{R})|\Psi(\mathbf{R})|^2 d\mathbf{R}}{\int_{\mathbb{R}^{3N}}|\Psi(\mathbf{R})|^2 d\mathbf{R}}, \tag{3}$$

with $O_L(\mathbf{R})=[\hat{O}\Psi](\mathbf{R})/\Psi(\mathbf{R})$, hence that

$$\frac{\langle\Psi,\hat{O}\Psi\rangle}{\langle\Psi,\Psi\rangle} \simeq \frac{1}{L}\sum_{n=1}^{L} O_L(\mathbf{R}^n), \tag{4}$$

where $(\mathbf{R}^n)_{n\geqslant1}$ are points of $\mathbb{R}^{3N}$ drawn from the probability distribution $|\Psi(\mathbf{R})|^2/\int_{\mathbb{R}^{3N}}|\Psi|^2$.

The VMC algorithms described in the present article are generic, in the sense that they can be used to compute the expectation value of any observable, for any N-body system. In the numerical example, we will, however, focus on the important case of the calculation of electronic energies of molecular systems. In this particular case, the expectation value to be computed reads

$$\frac{\langle\Psi,\hat{H}\Psi\rangle}{\langle\Psi,\Psi\rangle} = \frac{\int_{\mathbb{R}^{3N}}E_L(\mathbf{R})|\Psi(\mathbf{R})|^2 d\mathbf{R}}{\int_{\mathbb{R}^{3N}}|\Psi(\mathbf{R})|^2 d\mathbf{R}}, \tag{5}$$

where the scalar field $E_L(\mathbf{R})=[\hat{H}\Psi](\mathbf{R})/\Psi(\mathbf{R})$ is called the *local energy*. Remark that if Ψ is an eigenfunction of $\hat{H}$ associated with the eigenvalue E, $E_L(\mathbf{R})=E$ for all $\mathbf{R}$. Most often, VMC calculations are performed with trial wave functions Ψ that are good approximations of some ground state wave function Ψ_0. Consequently, $E_L(\mathbf{R})$ usually is a function of low variance (with respect to the probability density $|\Psi(\mathbf{R})|^2/\int_{\mathbb{R}^{3N}}|\Psi|^2$). This is the reason why, in practice, the approximation formula

$$\frac{\langle\Psi,\hat{H}\Psi\rangle}{\langle\Psi,\Psi\rangle} \simeq \frac{1}{L}\sum_{n=1}^{L} E_L(\mathbf{R}^n) \tag{6}$$

is fairly accurate, even for relatively small values of L (in practical applications on realistic molecular systems L ranges typically between 10^6 and 10^9).

Of course, the quality of the above approximation formula depends on the way the points $(\mathbf{R}^n)_{n\geqslant1}$ are generated. In Sec. II B, we describe the standard sampling method currently used for VMC calculations. It consists in a biased (or importance sampled) random walk in the configuration space (also called position space) $\mathbb{R}^{3N}$ corrected by a Metropolis-

© 2006 American Institute of Physics

Hastings accept/reject procedure. In Sec. II C, we introduce a new sampling scheme in which the points $(\mathbf{R}^n)_{n \geqslant 1}$ are the projections on the configuration space of one realization of some Markov chain on the phase space (also called position-momentum space) $\mathbb{R}^{3N} \times \mathbb{R}^{3N}$. This Markov chain is obtained by a modified Langevin dynamics, corrected by a Metropolis-Hastings accept/reject procedure.

Finally, some numerical results are presented in Sec. III. Various sampling algorithms are compared and it is demonstrated on a bench of representative examples that the algorithm based on the modified Langevin dynamics is the most efficient one of the algorithms studied here (the mathematical criteria for measuring the efficiency will be made precise below).

Before turning to the technical details, let us briefly comment on the underlying motivations of our approach. The reason why we have introduced a (purely fictitious) Langevin dynamics in the VMC framework is twofold.

- First, sampling methods based on Langevin dynamics turn out to outperform those based on biased random walks in classical molecular dynamics (see Ref. 2 for a quantitative study on carbon chains).

- Second, a specific problem encountered in VMC calculations on fermionic systems is that the standard discretization of the biased random walk (Euler scheme) does not behave properly close to the nodal surface of the trial wave function Ψ. This is due to the fact that the drift term blows up as the inverse of the distance to the nodal surface: if a random walker gets close to the nodal surface, the drift term repulses it far apart in a single time step. As demonstrated in Refs. 3 and 4, it is possible to partially circumvent this difficulty by resorting to more clever discretization schemes. Another strategy consists in replacing the biased random walk by a Langevin dynamics: the walkers then have a mass (hence some inertia) and the singular drift does not directly act on the position variables (as it is the case for the biased random walk), but indirectly *via* the momentum variables. The undesirable effects of the singularities are thus expected to be damped down.

II. DESCRIPTION OF THE ALGORITHMS

A. Metropolis-Hastings algorithm

The Metropolis algorithm[5] was later generalized by Hastings[6] to provide a general purpose sampling method, which combines the simulation of a Markov chain with an accept/reject procedure.

In the present article, the underlying state space is either the configuration space $\mathbb{R}^{3N}$ or the phase space $\mathbb{R}^{3N} \times \mathbb{R}^{3N} \equiv \mathbb{R}^{6N}$. Recall that a homogeneous Markov chain on $\mathbb{R}^d$ is characterized by its transition kernel p. It is by definition the non-negative function of $\mathbb{R}^d \times \mathcal{B}(\mathbb{R}^d)$ [$\mathcal{B}(\mathbb{R}^d)$ is the set of all the Borel sets of $\mathbb{R}^d$] such that if $\mathbf{X} \in \mathbb{R}^d$ and $B \in \mathcal{B}(\mathbb{R}^d)$ the probability for the Markov chain to lay in B at step $n+1$ if it is at $\mathbf{X}$ at step n is $p(\mathbf{X}, B)$. The transition kernel has a den-

sity with respect to the Lebesgue measure if for any $\mathbf{X} \in \mathbb{R}^d$, there exists a non-negative function $f_{\mathbf{X}} \in L^1(\mathbb{R}^d)$ such that

$$p(\mathbf{X}, B) = \int_B f_{\mathbf{X}}(\mathbf{X}') d\mathbf{X}'. \tag{7}$$

The non-negative number $f_{\mathbf{X}}(\mathbf{X}')$ is often denoted by $T(\mathbf{X} \to \mathbf{X}')$ and the function $T: \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+$, is called the transition density.

Given a Markov chain on $\mathbb{R}^d$ with transition density T and a positive function $f \in L^1(\mathbb{R}^d)$, the Metropolis-Hastings algorithm consists in generating a sequence $(\mathbf{X}^n)_{n \in \mathbb{N}}$ of points in $\mathbb{R}^d$ starting from some point $\mathbf{X}^0 \in \mathbb{R}^d$ according to the following iterative procedure.

- Propose a move from $\mathbf{X}^n$ to $\widetilde{\mathbf{X}}^{n+1}$ according to the transition density $T(\mathbf{X}^n \to \widetilde{\mathbf{X}}^{n+1})$.

- Compute the acceptance rate,

$$A(\mathbf{X}^n \to \widetilde{\mathbf{X}}^{n+1}) = \min\left(\frac{f(\widetilde{\mathbf{X}}^{n+1}) T(\widetilde{\mathbf{X}}^{n+1} \to \mathbf{X}^n)}{f(\mathbf{X}^n) T(\mathbf{X}^n \to \widetilde{\mathbf{X}}^{n+1})}, 1 \right).$$

- Draw a random variable U^n uniformly distributed in $[0, 1]$, if $U^n \leqslant A(\mathbf{X}^n \to \widetilde{\mathbf{X}}^{n+1})$, accept the move: $\mathbf{X}^{n+1} = \widetilde{\mathbf{X}}^{n+1}$, and if $U^n > A(\mathbf{X}^n \to \widetilde{\mathbf{X}}^{n+1})$, reject the move: $\mathbf{X}^{n+1} = \mathbf{X}^n$.

It is not difficult to show (see Ref. 7 for instance) that for a very large class of transition densities T, the points $\mathbf{X}^n$ generated by the Metropolis-Hastings algorithm are asymptotically distributed according to the probability density $f(\mathbf{X})/\int_{\mathbb{R}^d} f$. On the other hand, the practical efficiency of the algorithm crucially depends on the choice of the transition density (i.e., of the Markov chain).

B. Random walks in the configuration space

In this section, the state space is the configuration space $\mathbb{R}^{3N}$ and $f = |\Psi|^2$, so that the Metropolis-Hastings algorithm actually samples the probability density $|\Psi(\mathbf{R})|^2 / \int_{\mathbb{R}^{3N}} |\Psi|^2$.

1. Simple random walk

In the original paper[5] of Metropolis *et al.*, the Markov chain is a simple random walk,

$$\widetilde{\mathbf{R}}^{n+1} = \mathbf{R}^n + \Delta R \mathbf{U}^n, \tag{8}$$

where ΔR is the step size and $\mathbf{U}^n$ are independent and identically distributed (iid) random vectors drawn uniformly in the $3N$-dimensional cube $K = [-1, 1]^{3N}$. The corresponding transition density is $T(\mathbf{R} \to \mathbf{R}') = (2\Delta R)^{-3N} \chi_K((\mathbf{R} - \mathbf{R}')/\Delta R)$ where χ_K is the characteristic function of the cube K (note that in this particular case, $T(\mathbf{R} \to \mathbf{R}') = T(\mathbf{R}' \to \mathbf{R})$).

2. Biased random walk

The simple random walk is far from being the optimal choice: it induces a high rejection rate, hence a large variance. A variance reduction technique usually referred to as

the importance sampling method, consists in considering the so-called biased random walk or overdamped Langevin dynamics,[8]

$$d\mathbf{R}(t) = \nabla[\ln|\Psi|](\mathbf{R}(t))dt + d\mathbf{W}(t), \tag{9}$$

where $\mathbf{W}(t)$ is a $3N$-dimensional Wiener process. Note that $|\Psi|^2$ is an invariant measure of the Markov process (9) and, better, that the dynamics (9) is in fact ergodic and satisfies a detailed balance property.[7] The qualifier ergodic means that for any function g, $\mathbb{R}^{3N} \rightarrow \mathbb{R}$, integrable with respect to $|\Psi(\mathbf{R})|^2 d\mathbf{R}$,

$$\lim_{T \rightarrow +\infty} \frac{1}{T} \int_0^T g(\mathbf{R}(t))dt = \frac{\int_{\mathbb{R}^{3N}} g(\mathbf{R})|\Psi(\mathbf{R})|^2 d\mathbf{R}}{\int_{\mathbb{R}^{3N}} |\Psi(\mathbf{R})|^2 d\mathbf{R}}, \tag{10}$$

the convergence being almost sure and in L^1. The detailed balance property reads

$$|\Psi(\mathbf{R})|^2 T_{\Delta t}(\mathbf{R} \rightarrow \mathbf{R}') = |\Psi(\mathbf{R}')|^2 T_{\Delta t}(\mathbf{R}' \rightarrow \mathbf{R}), \tag{11}$$

for any $\Delta t > 0$, where $T_{\Delta t}(\mathbf{R} \rightarrow \mathbf{R}')$ is the probability density that the Markov process (9) is at $\mathbf{R}'$ at time $t + \Delta t$ if it is at $\mathbf{R}$ at time t. These above results are classical for regular, positive functions Ψ, and have been recently proven for fermionic wave functions[9] (in the latter case, the dynamics is ergodic in each nodal pocket of the wave function Ψ).

Note that if one uses the Markov chain of density $T_{\Delta t}(\mathbf{R} \rightarrow \mathbf{R}')$ in the Metropolis-Hastings algorithm, the accept/reject step is useless, since due to the detailed balance property, the acceptance rate always equals 1.

The exact value of $T_{\Delta t}(\mathbf{R} \rightarrow \mathbf{R}')$ being not known, a discretization of Eq. (9) with Euler scheme, is generally used,

$$\mathbf{R}^{n+1} = \mathbf{R}^n + \Delta t \, \nabla[\ln|\Psi|](\mathbf{R}^n) + \Delta \mathbf{W}^n, \tag{12}$$

where $\Delta \mathbf{W}^n$ are iid Gaussian random vectors with zero mean and covariance matrix $\Delta t I_{3N}$ (I_{3N} is the identity matrix). The Euler scheme leads to the approximated transition density,

$$T_{\Delta t}^{\text{Euler}}(\mathbf{R} \rightarrow \mathbf{R}') = \frac{1}{(2\pi\Delta t)^{3N/2}}$$
$$\times \exp\left(-\frac{|\mathbf{R}' - \mathbf{R} - \Delta t \, \nabla[\ln|\Psi|](\mathbf{R})|^2}{2\Delta t}\right). \tag{13}$$

The time discretization introduces the so-called *time-step error*, whose consequence is that (12) samples $|\Psi(\mathbf{R})|^2 / \int_{\mathbb{R}^{3N}} |\Psi|^2$ *only approximately*. Note that the Metropolis-Hastings accept/reject procedure perfectly corrects the time-step error. In the limit $\Delta t \rightarrow 0$, the time-step error vanishes and the accept/reject procedure is useless.

This sampling method is much more efficient than the Metropolis-Hastings algorithm based on the simple random walk, since the Markov chain (12) does a large part of the work (for sufficiently small time-steps, it samples a good approximation of $|\Psi(\mathbf{R})|^2 / \int |\Psi^2|$), which is clearly not the case for the simple random walk.

The standard method in VMC computations currently is the Metropolis-Hastings algorithm based on the Markov chain defined by (12). For refinements of this method, we refer to (Refs. 10–12).

C. Random walks in the phase space

In this section, the state space is the phase space $\mathbb{R}^{3N} \times \mathbb{R}^{3N}$. Let us emphasize that the introduction of momentum variables is nothing but a numerical artifice. The phase space trajectories that will be dealt with in this section do not have any physical meaning.

1. Langevin dynamics

The Langevin dynamics of a system of N particles of mass m evolving in an external potential V reads

$$d\mathbf{R}(t) = (1/m)\mathbf{P}(t)dt,$$
$$d\mathbf{P}(t) = -\nabla V(\mathbf{R}(t))dt - \gamma \mathbf{P}(t)dt + \sigma d\mathbf{W}(t). \tag{14}$$

As above, $\mathbf{R}(t)$ is a $3N$-dimensional vector collecting the positions at time t of the N particles. The components of the $3N$-dimensional vector $\mathbf{P}(t)$ are the corresponding momenta and $\mathbf{W}(t)$ is a $3N$-dimensional Wiener process. The Langevin dynamics can be considered as a perturbation of the Newton dynamics (for which $\gamma = 0$ and $\sigma = 0$). The magnitudes σ and γ of the random forces $\sigma d\mathbf{W}(t)$ and of the drag term $-\gamma\mathbf{P}(t)dt$ are related through the fluctuation-dissipation formula,

$$\sigma^2 = \frac{2m\gamma}{\beta}, \tag{15}$$

where β is the reciprocal temperature of the system. Let us underline that in the present setting, β is a numerical parameter that is by no means related to the physical temperature of the system. It can be checked (at least for regular potentials V) that the canonical distribution

$$d\Pi(\mathbf{R}, \mathbf{P}) = Z^{-1} e^{-\beta H(\mathbf{R}, \mathbf{P})} d\mathbf{R} d\mathbf{P} \tag{16}$$

is an invariant probability measure for the system, Z being a normalization constant, and

$$H(\mathbf{P}, \mathbf{R}) = \frac{|\mathbf{P}|^2}{2m} + V(\mathbf{R}) \tag{17}$$

being the Hamiltonian of the underlying Newton dynamics. In addition, the Langevin dynamics is ergodic (under some assumptions on V). Thus, choosing

$$\beta = 1 \text{ and } V = -\ln|\Psi|^2, \tag{18}$$

the projection on the position space of the Langevin dynamics samples $|\Psi(\mathbf{R})|^2 / \int |\Psi|^2$. On the other hand, the Langevin dynamics does not satisfy the detailed balance property. We will come back to this important point in the forthcoming section.

In this context, the parameters m and γ [σ being then obtained through (15)] should be seen as numerical parameters to be optimized to get the best sampling. We now describe how to discretize and apply a Metropolis-Hastings algorithm to the Langevin dynamics (14) in the context of VMC.

TABLE I. Comparison of different discretization schemes for Langevin dynamics. The reference energy is −7.471 98(4) a.u..

Δt	BRW	BBK[a]	Force interpolation[b]	Splitting[c]	Ricci and Ciccotti[d]
0.05	−7.3758(316)	−7.4395(246)	−7.4386(188)	−7.4467(137)	−7.4576(07)
0.005	−7.4644(069)	−7.4698(015)	−7.4723(015)	−7.4723(015)	−7.4701(20)
0.001	−7.4740(007)	−7.4728(013)	−7.4708(017)	−7.4708(017)	−7.4696(17)
0.0005	−7.4732(010)	−7.4700(023)	−7.4709(022)	−7.4708(022)	−7.4755(26)

[a]Reference 13.
[b]Reference 14.
[c]Reference 15.
[d]Reference 16.

2. Time discretization of the Langevin dynamics

Many discretization schemes exist for Langevin dynamics. In order to choose which algorithm is best for VMC, we have tested four different schemes available in the literature,[13–16] with parameters $\beta=1$, $\gamma=1$, and $m=1$. Our benchmark system is a lithium atom, and a single determinantal wave function built upon Slater-type atomic orbitals, multiplied by a Jastrow factor. We turn off the accept/reject step in these preliminary tests, since our purpose is to compare the time-step errors for the various algorithms. From the results displayed in Table I, one can see that the Ricci-Ciccotti algorithm[16] is the method which generates the smallest time-step error. This algorithm reads

$$\mathbf{R}^{n+1} = \mathbf{R}^n + (\Delta t/m)\mathbf{P}^n e^{-\gamma\Delta t/2} + (\Delta t/2m)$$
$$\times [-\nabla V(\mathbf{R}^n)\Delta t + \mathbf{G}^n] e^{-\gamma\Delta t/4},$$

$$\mathbf{P}^{n+1} = \mathbf{P}^n e^{-\gamma\Delta t} - (\Delta t/2)[\nabla V(\mathbf{R}^n)$$
$$+ \nabla V(\mathbf{R}^{n+1})] e^{-\gamma\Delta t/2} + \mathbf{G}^n e^{-\gamma\Delta t/2}, \tag{19}$$

where $\mathbf{G}^n$ are iid. Gaussian random vectors with zero mean and variance $\sigma^2 I_{3N}$ with $\sigma^2 = (2\gamma m/\beta)\Delta t$.

It can be seen from Table I that the Ricci-Ciccotti algorithm also outperforms the biased random walk (12), as far as sampling issues are concerned. In the following, we shall therefore use the Ricci-Ciccotti algorithm.

3. Metropolized Langevin dynamics

The discretized Langevin dynamics does not exactly sample the target distribution Π, but rather from some approximation $\Pi_{\Delta t}$ of Π. It is therefore tempting to introduce a Metropolis-Hastings accept/reject step to further improve the quality of the sampling. Unfortunately, this idea cannot be straightforwardly implemented for two reasons.

- First, this is not technically feasible, since the Markov chain defined by (19) does not have a transition density. Indeed, as the same Gaussian random vectors $\mathbf{G}^n$ are used to update both the positions and the momenta, the measure $p((\mathbf{R}^n, \mathbf{P}^n), \cdot)$ is supported on a $3N$-dimensional submanifold of the phase space $\mathbb{R}^{3N} \times \mathbb{R}^{3N}$.

- Second, leaving apart the above mentioned technical difficulty, which is specific to the Ricci-Ciccotti scheme, the Langevin dynamics is *a priori* not an efficient Markov chain for the Metropolis-Hastings algorithm because it does not satisfy the detailed balance property.

Let us now explain how to tackle these two issues, starting with the first one.

To make it compatible with the Metropolis-Hastings framework, one needs to slightly modify the Ricci-Ciccotti algorithm. Following (Refs. 14 and 17), we thus introduce iid *correlated* Gaussian vectors $(G_{1,i}^n, G_{2,i}^n)$ $(1 \leq i \leq 3N)$ such that

$$\langle (G_{1,i}^n)^2 \rangle = \sigma_1^2 = \frac{\Delta t}{\beta m \gamma}\left(2 - \frac{3 - 4e^{-\gamma\Delta t} + e^{-2\gamma\Delta t}}{\gamma\Delta t}\right), \tag{20a}$$

$$\langle (G_{2,i}^n)^2 \rangle = \sigma_2^2 = \frac{m}{\beta}(1 - e^{-2\gamma\Delta t}), \tag{20b}$$

$$\frac{\langle G_{1,i}^n G_{2,i}^n \rangle}{\sigma_1 \sigma_2} = c_{12} = \frac{(1 - e^{-\gamma\Delta t})^2}{\beta\gamma\sigma_1\sigma_2}. \tag{20c}$$

Setting $\mathbf{G}_1^n = (G_{1,i}^n)_{1 \leq i \leq 3N}$ and $\mathbf{G}_2^n = (G_{2,i}^n)_{1 \leq i \leq 3N}$, the modified Ricci-Ciccotti algorithm reads

$$\mathbf{R}^{n+1} = \mathbf{R}^n + (\Delta t/m)\mathbf{P}^n e^{-\gamma\Delta t/2}$$
$$- (\Delta t^2/2m)\nabla V(\mathbf{R}^n)e^{-\gamma\Delta t/4} + \mathbf{G}_1^n,$$

$$\mathbf{P}^{n+1} = \mathbf{P}^n e^{-\gamma\Delta t} - (\Delta t/2)[\nabla V(\mathbf{R}^n)$$
$$+ \nabla V(\mathbf{R}^{n+1})]e^{-\gamma\Delta t/2} + \mathbf{G}_2^n. \tag{21}$$

The above scheme is a consistent discretization of (14) and the corresponding Markov chain does have a transition density, which reads (see Appendix)

$$T_{\Delta t}^{\text{MRC}}((\mathbf{R}^n, \mathbf{P}^n) \to (\mathbf{R}^{n+1}, \mathbf{P}^{n+1}))$$
$$= Z^{-1} \exp\left[-\frac{1}{2(1 - c_{12}^2)}\left(\left(\frac{|\mathbf{d}_1|}{\sigma_1}\right)^2\right.\right.$$
$$\left.\left. + \left(\frac{|\mathbf{d}_2|}{\sigma_2}\right)^2 - 2c_{12}\frac{\mathbf{d}_1}{\sigma_1}\cdot\frac{\mathbf{d}_2}{\sigma_2}\right)\right], \tag{22a}$$

with

$$\mathbf{d}_1 = \mathbf{R}^{n+1} - \mathbf{R}^n - \Delta t\frac{\mathbf{P}^n}{m}e^{-\gamma\Delta t/2} + \frac{\Delta t^2}{2m}\nabla V(\mathbf{R}^n)e^{-\gamma\Delta t/4}, \tag{22b}$$

$$\mathbf{d}_2 = \mathbf{P}^{n+1} - \mathbf{P}^n e^{-\gamma\Delta t} + \tfrac{1}{2}\Delta t[\nabla V(\mathbf{R}^n) + \nabla V(\mathbf{R}^{n+1})]e^{-\gamma\Delta t/2}. \tag{22c}$$

Unfortunately, inserting directly the transition density (22) in the Metropolis-Hastings algorithm leads to a high rejection rate. Indeed, if $(\mathbf{R}^n, \mathbf{P}^n)$ and $(\mathbf{R}^{n+1}, \mathbf{P}^{n+1})$ are related through (21), $T_{\Delta t}^{\mathrm{MRC}}((\mathbf{R}^n, \mathbf{P}^n) \to (\mathbf{R}^{n+1}, \mathbf{P}^{n+1}))$ usually is much greater than $T_{\Delta t}^{\mathrm{MRC}}((\mathbf{R}^{n+1}, \mathbf{P}^{n+1}) \to (\mathbf{R}^n, \mathbf{P}^n))$, since the probability that the random forces are strong enough to make the particle go back in one step from where it comes is very low in general. This is related to the fact that the Langevin dynamics does not satisfy the detailed balance relation.

It is, however, possible to further modify the overall algorithm by ensuring some microscopic reversibility in order to finally obtain low rejection rates. For this purpose, we introduce momentum reversions. Denoting by $T_{\Delta t}^{\mathrm{Langevin}}$ the transition density of the Markov chain obtained by integrating (14) *exactly* on the time interval $[t, t+\Delta t]$, it is indeed not difficult to check (under convenient assumptions on $V = -\ln|\Psi|^2$) that the Markov chain defined by the transition density

$$\widetilde{T}_{\Delta t}^{\mathrm{Langevin}}((\mathbf{R}, \mathbf{P}) \to (\mathbf{R}', \mathbf{P}'))$$
$$= T_{\Delta t}^{\mathrm{Langevin}}((\mathbf{R}, \mathbf{P}) \to (\mathbf{R}', -\mathbf{P}')) \tag{23}$$

is ergodic with respect to Π and satisfies the detailed balance property,

$$\Pi(\mathbf{R}, \mathbf{P})\widetilde{T}_{\Delta t}^{\mathrm{Langevin}}((\mathbf{R}, \mathbf{P}) \to (\mathbf{R}', \mathbf{P}'))$$
$$= \Pi(\mathbf{R}', \mathbf{P}')\widetilde{T}_{\Delta t}^{\mathrm{Langevin}}((\mathbf{R}', \mathbf{P}')$$
$$\to (\mathbf{R}, \mathbf{P})). \tag{24}$$

Replacing the exact transition density $T_{\Delta t}^{\mathrm{Langevin}}$ by the approximation $T_{\Delta t}^{\mathrm{MRC}}$, we now consider the transition density

$$\widetilde{T}_{\Delta t}^{\mathrm{MRC}}((\mathbf{R}, \mathbf{P}) \to (\mathbf{R}', \mathbf{P}')) = T_{\Delta t}^{\mathrm{MRC}}((\mathbf{R}, \mathbf{P}) \to (\mathbf{R}', -\mathbf{P}')). \tag{25}$$

The new sampling algorithm that we propose can be stated as follows.

- Propose a move from $(\mathbf{R}^n, \mathbf{P}^n)$ to $(\widetilde{\mathbf{R}}^{n+1}, \widetilde{\mathbf{P}}^{n+1})$ using the transition density $\widetilde{T}_{\Delta t}^{\mathrm{MRC}}$. In other words, perform one step of the modified Ricci-Ciccotti algorithm,

$$\mathbf{R}_*^{n+1} = \mathbf{R}^n + (\Delta t/m)\mathbf{P}^n e^{-\gamma\Delta t/2}$$
$$- (\Delta t^2/2m)\,\nabla\,V(\mathbf{R}^n)e^{-\gamma\Delta t/4} + \mathbf{G}_1^n,$$
$$\mathbf{P}_*^{n+1} = \mathbf{P}^n e^{-\gamma\Delta t} - (\Delta t/2)[\nabla V(\mathbf{R}^n)$$
$$+ \nabla V(\mathbf{R}^{n+1})]e^{-\gamma\Delta t/2} + \mathbf{G}_2^n, \tag{26}$$

and set $(\widetilde{\mathbf{R}}^{n+1}, \widetilde{\mathbf{P}}^{n+1}) = (\mathbf{R}_*^{n+1}, -\mathbf{P}_*^{n+1})$.

- Compute the acceptance rate,

$$A((\mathbf{R}^n, \mathbf{P}^n) \to (\widetilde{\mathbf{R}}^{n+1}, \widetilde{\mathbf{P}}^{n+1})) = \min\left(\frac{\Pi(\widetilde{\mathbf{R}}^{n+1}, \widetilde{\mathbf{P}}^{n+1})\widetilde{T}_{\Delta t}^{\mathrm{MRC}}((\widetilde{\mathbf{R}}^{n+1}, \widetilde{\mathbf{P}}^{n+1}) \to (\mathbf{R}^n, \mathbf{P}^n))}{\Pi(\mathbf{R}^n, \mathbf{P}^n)\widetilde{T}_{\Delta t}^{\mathrm{MRC}}((\mathbf{R}^n, \mathbf{P}^n) \to (\widetilde{\mathbf{R}}^{n+1}, \widetilde{\mathbf{P}}^{n+1}))}, 1 \right).$$

- Draw a random variable U^n uniformly distributed in $(0, 1)$ and if $U^n \leq A((\mathbf{R}^n, \mathbf{P}^n) \to (\widetilde{\mathbf{R}}^{n+1}, \widetilde{\mathbf{P}}^{n+1}))$, accept the proposal: $(\overline{\mathbf{R}}^{n+1}, \overline{\mathbf{P}}^{n+1}) = (\widetilde{\mathbf{R}}^{n+1}, \widetilde{\mathbf{P}}^{n+1})$, and if $U^n > A((\mathbf{R}^n, \mathbf{P}^n) \to (\widetilde{\mathbf{R}}^{n+1}, \widetilde{\mathbf{P}}^{n+1}))$, reject the proposal, and set $(\overline{\mathbf{R}}^{n+1}, \overline{\mathbf{P}}^{n+1}) = (\mathbf{R}^n, \mathbf{P}^n)$.

- Reverse the momenta,

$$(\mathbf{R}^{n+1}, \mathbf{P}^{n+1}) = (\overline{\mathbf{R}}^{n+1}, -\overline{\mathbf{P}}^{n+1}). \tag{27}$$

Note that a momentum reversion is systematically performed just after the Metropolis-Hastings step. As the invariant measure Π is left unchanged by this operation, the global algorithm (Metropolis-Hastings step based on the transition density $\widetilde{T}_{\Delta t}^{\mathrm{MRC}}$ plus momentum reversion) actually samples Π. The role of the final momentum reversion is to preserve the underlying Langevin dynamics: while the proposals are accepted, the above algorithm generates Langevin trajectories, that are known to efficiently sample an approximation of the target density Π. Numerical tests seem to show that, in addition, the momentum reversion also plays a role when the

proposal is rejected: it seems to increase the acceptance rate of the next step, preventing the walkers from being trapped in the vicinity of the nodal surface $\Psi^{-1}(0)$.

As the points $(\mathbf{R}^n, \mathbf{P}^n)$ of the phase space generated by the above algorithm form a sampling of Π, the positions $(\mathbf{R}^n)$ sample $|\Psi(\mathbf{R})|^2 / \int_{\mathbb{R}^{3N}} |\Psi|^2$ and can therefore be used for VMC calculations.

III. NUMERICAL EXPERIMENTS AND APPLICATIONS

A. Measuring the efficiency

A major drawback of samplers based on Markov processes is that they generate sequentially correlated data. For a trajectory of L steps, the effective number of independent observations is in fact $L_{\mathrm{eff}} = L/N_{\mathrm{corr}}$, where N_{corr} is the *correlation length*, namely, the number of successive correlated moves.

In the following applications, we provide estimators for the correlation length N_{corr} and for the so-called inefficiency η (see below), which are relevant indicators of the quality of the sampling. In this section, following Stedman *et al.*,[18] we describe the way these quantities are defined and computed.

The sequence of samples is split into N_B blocks of L_B steps, where the number L_B is chosen such that it is a few orders of magnitude higher than N_{corr}. The mean energy is $\langle E_L \rangle_{|\Psi|^2}$ and the variance is $\sigma^2 = \langle (E_L - \langle E_L \rangle_{|\Psi|^2})^2 \rangle_{|\Psi|^2}$. These quantities are defined independently on the VMC algorithm used. The empirical mean of the local energy reads

$$\langle E_L \rangle_{|\Psi|^2}^{N_B, L_B} = \frac{1}{N_B L_B} \sum_{i=1}^{N_B L_B} E_L(\mathbf{R}^i). \tag{28}$$

The empirical variance over all the individual steps is given by

$$[\sigma^{N_B, L_B}]^2 = \frac{1}{N_B L_B} \sum_{i=1}^{N_B L_B} (E_L(\mathbf{R}^i) - \langle E_L \rangle_{|\Psi|^2}^{N_B, L_B})^2, \tag{29}$$

and the empirical variance over the blocks by

$$[\sigma_B^{N_B, L_B}]^2 = \frac{1}{N_B} \sum_{i=1}^{N_B} (E_{B,i} - \langle E_L \rangle_{|\Psi|^2}^{N_B, L_B})^2, \tag{30}$$

where $E_{B,i}$ is the average energy over block i,

$$E_{B,i} = \frac{1}{L_B} \sum_{j=(i-1)L_B+1}^{iL_B} E_L(\mathbf{R}^j). \tag{31}$$

Following Ref. 18, we define the correlation length as

$$N_{corr} = \lim_{N_B \to \infty} \lim_{L_B \to \infty} L_B \frac{[\sigma_B^{N_B, L_B}]^2}{[\sigma^{N_B, L_B}]^2}, \tag{32}$$

and the inefficiency η of the run as

$$\eta = \lim_{N_B \to \infty} \lim_{L_B \to \infty} L_B [\sigma_B^{N_B, L_B}]^2. \tag{33}$$

On the numerical examples presented below, the relative fluctuations of the quantities $L_B[\sigma_B^{N_B,L_B}]^2/[\sigma^{N_B,L_B}]^2$ and $L_B[\sigma_B^{N_B,L_B}]^2$ become small for $L_B > 50$ and $N_B > 50$.

The definition of these two quantities can be understood as follows. Since $L_B \gg N_{corr}$ and only L_B/N_{corr} are independent samples among the samples in the block, the central limit theorem yields $E_{B,i} \simeq \langle E_L \rangle_{|\Psi|^2} + \sigma G^i / \sqrt{L_B/N_{corr}}$, where G^i are iid normal random variables. Thus, in the limit $N_B \to \infty$ and $L_B \to \infty$, we obtain that $(\sigma_B^{N_B,L_B})^2 = \sigma^2/(L_B/N_{corr})$. Since $\lim_{N_B \to \infty} \lim_{L_B \to \infty} [\sigma^{N_B,L_B}]^2 = \sigma^2$, we obtain (32). The inefficiency η is thus equal to $N_{corr}\sigma^2$ and is large if the variance is large, or if the number of correlated steps is large.

Using this measure of efficiency, we can now compare the sampling algorithms (the simple random walk, the biased random walk, and the Langevin algorithm) for various systems. In any case, a Metropolis-Hastings accept/reject step is used. We found empirically from several tests that convenient values for the parameters of the Langevin algorithm are $\gamma = 1$ and $m = Z^{3/2}$, where Z is the highest nuclear charge among all the nuclei. For each algorithm, we compare the efficiency for various values of the step length, namely, the increment ΔR in the case of the simple random walk, and the time step Δt for the other two schemes. For a given algorithm, simple arguments corroborated by numerical tests show that there exists an optimal value of this increment: for smaller (respectively for larger) increments, the correlation

TABLE II. The lithium atom: comparison of the simple random walk, the biased random walk, and the proposed Langevin algorithm. The runs were carried out with 100 walkers, each realizing 50 blocks of 1000 steps. The reference energy is −7.471 98(4) a.u.

ΔR	$\langle E_L \rangle$	N_{corr}	η	$\langle A \rangle$	
		Simple random walk			
0.05	−7.471 26(183)	94.5±3.3	11.72(42)	0.91	
0.10	−7.472 39(97)	35.2±1.2	4.8(14)	0.82	
0.15	−7.471 89(75)	20.5(5)	2.30(06)	0.74	
0.20	−7.471 57(56)	14.3(4)	1.62(04)	0.66	
0.25	−7.471 82(56)	12.1(3)	1.40(05)	0.59	
0.30	−7.471 89(56)	11.4(3)	1.57(17)	0.52	
0.35	−7.472 75(59)	12.4(3)	1.57(17)	0.46	
0.40	−7.471 30(63)	14.4(5)	1.93(22)	0.40	
Δt	$\langle E_L \rangle$	N_{corr}	η	$\langle |\Delta \mathbf{R}| \rangle$	$\langle A \rangle$
		Biased random walk			
0.01	−7.471 98(53)	10.31(29)	1.23(3)	0.284(09)	0.98
0.03	−7.471 56(39)	5.26(14)	0.73(7)	0.444(21)	0.92
0.04	−7.471 95(35)	4.82(12)	0.57(3)	0.486(26)	0.88
0.05	−7.472 19(32)	4.74(11)	0.55(2)	0.514(31)	0.85
0.06	−7.472 04(38)	4.95(11)	0.58(3)	0.533(36)	0.81
0.07	−7.472 51(32)	5.39(14)	0.61(3)	0.546(40)	0.78
0.10	−7.472 49(42)	7.56(25)	0.87(5)	0.555(50)	0.68
		Langevin			
0.20	−7.472 33(34)	5.07(10)	0.60(1)	0.236(08)	0.97
0.30	−7.472 07(34)	4.14(09)	0.47(1)	0.328(15)	0.93
0.35	−7.471 80(31)	3.96(08)	0.45(1)	0.366(18)	0.91
0.40	−7.471 85(29)	3.75(08)	0.44(2)	0.399(22)	0.89
0.45	−7.472 64(29)	3.88(08)	0.45(2)	0.426(25)	0.86
0.50	−7.471 91(29)	4.07(14)	0.46(2)	0.426(25)	0.84
0.60	−7.472 58(32)	4.78(16)	0.52(2)	0.481(36)	0.78

between two successive positions increases since the displacement of the particle is small (respectively since many moves are rejected), and this increases the number of correlated steps N_{corr}.

One can notice on the results (see Tables II–V) that a large error bar corresponds to large values for N_{corr} and η. The quantities N_{corr} and η are a way to refine the measure of efficiency, since the same length of error bar may be obtained for different values of the numerical parameters.

Let us now present some numerical tests. We compare the algorithms and parameters at a fixed computational cost. The reference values are obtained by ten times longer VMC simulations. The error bars given in parentheses are 68% confidence intervals. We also provide the mean acceptance rate (denoted by $\langle A \rangle$ in the tables) and, when it is relevant, the mean of the length of the increment $\mathbf{R}^{n+1} - \mathbf{R}^n$ over one time step (denoted by $\langle |\Delta \mathbf{R}| \rangle$ in the tables) for the biased random walk and the Langevin dynamics. These tests were performed using the QMC=Chem program,[19] and the wave functions are available upon request.

B. Atoms

Lithium. The lithium atom was chosen as a first simple example. The wave function is the same as for the benchmark system used for the comparison of the various Langevin schemes, namely, a single Slater determinant of Slater-

TABLE III. The fluorine atom: comparison of the simple random walk, the biased random walk, and the proposed Langevin algorithm. The runs were carried out with 100 walkers, each realizing 100 blocks of 100 steps. The reference energy is −99.397(2) a.u.

ΔR	$\langle E_L \rangle$	N_{corr}	η	$\langle A \rangle$	
		Simple random walk			
0.02	−99.398(72)	38.9(7)	823(31)	0.87	
0.05	−99.426(39)	20.3(4)	405(11)	0.69	
0.08	−99.406(28)	15.6(4)	326(17)	0.53	
0.10	−99.437(23)	15.8(3)	282(07)	0.44	
0.12	−99.402(24)	16.6(4)	341(24)	0.36	
0.15	−99.398(25)	19.4(5)	412(41)	0.27	

Δt	$\langle E_L \rangle$	N_{corr}	η	$\langle |\Delta \mathbf{R}| \rangle$	$\langle A \rangle$
		Biased random walk			
0.002	−99.411(21)	9.9(2)	206(04)	0.211(08)	0.94
0.003	−99.424(17)	8.8(2)	173(04)	0.242(11)	0.90
0.004	−99.430(15)	7.6(2)	147(03)	0.263(16)	0.86
0.005	−99.399(14)	7.3(2)	142(03)	0.275(17)	0.82
0.006	−99.406(14)	7.4(1)	137(03)	0.282(19)	0.79
0.007	−99.430(14)	7.4(2)	142(08)	0.286(21)	0.75
0.008	−99.421(13)	7.6(2)	141(05)	0.287(23)	0.71
0.009	−99.406(13)	7.8(2)	177(19)	0.285(25)	0.67
0.010	−99.419(15)	7.8(2)	162(10)	0.281(27)	0.64
0.011	−99.416(14)	8.3(2)	147(05)	0.276(28)	0.60
0.012	−99.420(15)	9.1(3)	205(34)	0.270(29)	0.57
0.013	−99.425(17)	10.2(4)	224(38)	0.263(30)	0.54
		Langevin			
0.10	−99.402(16)	8.9(2)	199(04)	0.095(02)	0.98
0.20	−99.403(12)	6.0(1)	123(02)	0.174(06)	0.94
0.25	−99.402(12)	5.4(1)	108(02)	0.204(09)	0.91
0.30	−99.395(11)	5.3(1)	104(02)	0.228(10)	0.87
0.35	−99.409(12)	5.4(1)	108(06)	0.245(15)	0.83
0.40	−99.402(11)	5.5(1)	102(03)	0.256(18)	0.78
0.45	−99.406(11)	5.9(1)	114(06)	0.261(21)	0.73
0.50	−99.408(12)	6.6(2)	124(07)	0.262(24)	0.68
0.55	−99.407(14)	7.9(4)	149(10)	0.257(26)	0.62
0.60	−99.405(15)	9.2(4)	178(13)	0.250(42)	0.56

type basis functions improved by a Jastrow factor to take account of the electron correlation. The reference energy associated with this wave function is −7.471 98(4) a.u., and the comparison of the algorithms is given in Table II. The runs were made of 100 random walks composed of 50 blocks of 1000 steps. For the simple random walk, the lowest values of the correlation length and of the inefficiency are, respectively, 11.4 and 1.40. The biased random walk is much more efficient, since the optimal correlation length and inefficiency are more than twice smaller, i.e., 4.74 and 0.55. The proposed algorithm is even more efficient: the optimal correlation length is 3.75 and the optimal inefficiency is 0.44.

Fluorine. The fluorine atom was chosen for its relatively "high" nuclear charge ($Z=9$), leading to a time scale separation of the core and valence electrons. The wave function is a Slater determinant with Gaussian-type basis functions where the $1s$ orbital was substituted by a Slater-type orbital, with a reference energy of −99.397(2) a.u. The runs were made of 100 random walks composed of 100 blocks of 100 steps. The results are given in Table III. For the simple

random walk, the lowest values of the correlation length and of the inefficiency are, respectively, 15.6 and 282. The biased random walk, for which the optimal correlation length and inefficiency are 7.4 and 137 respectively, is again twice more efficient than the simple random walk. The Langevin algorithm is more efficient than the biased random walk: the optimal correlation length is 5.3 and the optimal inefficiency is 102.

Copper. We can go even further in the time scale separation and take the copper atom ($Z=29$) as an example. The wave function is a Slater determinant with a basis of Slater-type atomic orbitals, improved by a Jastrow factor to take account of the electron correlation. The reference energy is −1639.2539(24). The runs were made of 40 random walks composed of 500 blocks of 500 steps. From Table IV, one can remark that the Langevin algorithm is again more efficient than the biased random walk, since the optimal correlation length and inefficiency are, respectively, 28.7 and 4027, whereas using the biased random walk, these values are 51.0 and 5953.

TABLE IV. The copper atom: comparison of the biased random walk with the proposed Langevin algorithm. The runs were carried out with 40 walkers, each realizing 500 blocks of 500 steps. The reference energy is $-1639.2539(24)$ a.u.

Δt	$\langle E_L \rangle$	N_{corr}	η	$\langle |\Delta \mathbf{R}| \rangle$	$\langle A \rangle$
		Biased random walk			
0.000 3	$-1639.2679(78)$	79.1 ± 2.7	10 682(420)	0.1311(108)	0.86
0.000 4	$-1639.2681(98)$	70.4 ± 1.3	8 682(204)	0.1385(137)	0.81
0.000 5	$-1639.2499(96)$	61.3 ± 2.5	7 770(297)	0.114(162)	0.75
0.000 6	$-1639.2629(96)$	56.0 ± 1.2	6 834(88)	0.1414(183)	0.70
0.000 7	$-1639.2575(73)$	53.8 ± 0.8	6 420(81)	0.1393(201)	0.65
0.000 75	$-1639.2518(85)$	53.1 ± 0.9	6 330(91)	0.1377(209)	0.62
0.000 8	$-1639.2370(86)$	55.7 ± 3.6	6 612(405)	0.1357(216)	0.60
0.001 05	$-1639.2694(85)$	51.0 ± 0.8	5 953(90)	0.1228(241)	0.48
0.001 1	$-1639.2563(110)$	54.3 ± 1.8	6 513(221)	0.1198(245)	0.46
0.001 2	$-1639.2523(72)$	59.9 ± 5.5	7 266(658)	0.1136(251)	0.43
		Langevin			
0.05	$-1639.2553(92)$	61.3 ± 1.7	8 256(89)	0.371(1)	0.99
0.10	$-1639.2583(76)$	40.6 ± 3.1	5 319(383)	0.0705(30)	0.97
0.15	$-1639.2496(65)$	30.1 ± 0.8	4 042(103)	0.0978(60)	0.93
0.20	$-1639.2521(71)$	28.7 ± 0.9	4 027(403)	0.1173(96)	0.87
0.30	$-1639.2510(67)$	35.2 ± 2.5	4 157(291)	0.1326(170)	0.71
0.40	$-1639.2524(78)$	50.5 ± 3.7	5 922(455)	0.1210(225)	0.52

C. The phenol molecule

The phenol molecule was chosen to test the proposed algorithm because it contains three different types of atoms (H, C, and O). The wave function here is a single Slater determinant with Gaussian-type basis functions. The core molecular orbitals of the oxygen and carbon atoms were substituted by the corresponding atomic $1s$ orbitals. The comparison of the biased random walk with the Langevin algorithm is given in Table V. The optimal correlation length using the biased random walk is 10.17, whereas it is 8.23 with our Langevin algorithm. The optimal inefficiency is again lower with the Langevin algorithm (544) than with the biased random walk (653).

TABLE V. The phenol molecule: comparison of the biased random walk with the proposed Langevin algorithm. the runs were carried out with 100 walkers, each realizing 100 blocks of 100 steps. The reference energy is $-305.647(2)$ a.u.

Δt	$\langle E_L \rangle$	N_{corr}	η	$\langle |\Delta \mathbf{R}| \rangle$	$\langle A \rangle$
		Biased random walk			
0.003	$-305.6308(83)$	18.71(24)	1368(12)	0.522(29)	0.85
0.004	$-305.6471(78)$	16.00(28)	1193(30)	0.547(36)	0.80
0.005	$-305.6457(65)$	15.29(20)	1077(14)	0.555(43)	0.74
0.006	$-305.6412(79)$	15.00(17)	1018(11)	0.552(48)	0.69
0.007	$-305.6391(67)$	14.52(26)	1051(53)	0.540(52)	0.63
0.008	$-305.6530(65)$	14.72(19)	980(10)	0.523(56)	0.58
0.009	$-305.6555(82)$	15.28(28)	1272(163)	0.502(59)	0.54
		Langevin			
0.05	$-305.6417(101)$	23.13(41)	1932(41)	0.126(02)	0.99
0.1	$-305.6416(68)$	13.97(22)	1189(23)	0.240(06)	0.97
0.2	$-305.6496(57)$	9.70(13)	812(12)	0.408(20)	0.89
0.3	$-305.6493(56)$	9.36(16)	817(36)	0.487(36)	0.78
0.4	$-305.6473(58)$	12.21(22)	834(20)	0.485(50)	0.61
0.5	$-305.6497(80)$	17.51(44)	1237(52)	0.425(58)	0.43

D. Discussion of the results

We observe that on our numerical tests, the Langevin dynamics is always more efficient than the biased random walk. Indeed, we notice the following.

- The error bar (or N_{corr}, or η) obtained with the Langevin dynamics for an optimal set of numerical parameters is always smaller than the error bar obtained with other algorithms (for which we also optimize the numerical parameters).

- The size of the error bar does not seem to be as sensitive to the choice of the numerical parameters as for other methods. In particular, we observe on our numerical tests that the value $\Delta t = 0.2$ seems to be convenient to obtain good results with the Langevin dynamics, whatever the atom or molecule.

ACKNOWLEDGMENT

APPENDIX: DERIVATION OF THE TRANSITION PROBABILITY (22)

The random vector $(\mathbf{d}_1, \mathbf{d}_2)$ [defined by (22b) and (22c)] is a Gaussian random vector and therefore admits a density with respect to the Lebesgue measure in $\mathbb{R}^{6N}$. If, for $1 \leq i \leq 3N$, we denote by $d_{1,i}$ and $d_{2,i}$ the components of $\mathbf{d}_1$ and $\mathbf{d}_2$, we observe that the Gaussian random vectors $(d_{1,i}, d_{2,i})$ are iid. Therefore, the transition probability $T((\mathbf{R}^n, \mathbf{P}^n) \rightarrow (\mathbf{R}^{n+1}, \mathbf{P}^{n+1}))$ reads

$$T((\mathbf{R}^n, \mathbf{P}^n) \rightarrow (\mathbf{R}^{n+1}, \mathbf{P}^{n+1})) = Z^{-1}(p(d_{1,i}, d_{2,i}))^{3N}, \qquad (A1)$$

where Z is a normalization constant and p denotes the density (in $\mathbb{R}^2$) of the Gaussian random vectors $(d_{1,i}, d_{2,i})$.

From Eq. (21), one can see that

$$d_{1,i} = R_i^{n+1} - R_i^n - \Delta t \frac{P_i^n}{m} e^{-\gamma \Delta t/2} + \frac{\Delta t^2}{2m} \nabla_i V(\mathbf{R}^n) e^{-\gamma \Delta t/4},$$

$$d_{2,i} = P_i^{n+1} - P_i^n e^{-\gamma \Delta t} + \frac{1}{2} \Delta t [\nabla_i V(\mathbf{R}^n) + \nabla_i V(\mathbf{R}^{n+1})] e^{-\gamma \Delta t/2}$$

is a Gaussian random vector with covariance matrix

$$\Gamma = \begin{bmatrix} \sigma_1^2 & c_{12}\sigma_1\sigma_2 \\ c_{12}\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}.$$

Thus

$$p(d_1, d_2) = (2\pi\sqrt{\det \Gamma})^{-1} \exp\left(-\frac{1}{2}(d_1, d_2)\Gamma^{-1}(d_1, d_2)^T\right).$$
$$(A2)$$

Since

$$\Gamma^{-1} = \frac{1}{(1 - c_{12}^2)} \begin{bmatrix} 1/\sigma_1^2 & -c_{12}/\sigma_1\sigma_2 \\ -c_{12}/\sigma_1\sigma_2 & 1/\sigma_2^2 \end{bmatrix},$$

(22c) is easily obtained from (A1) and (A2).

[1] D. Bressanini and P. J. Reynolds, *Advances in Chemical Physics* (Wiley, New York, 1999), Vol. 105, p. 37.

[2] E. Cancès, F. Legoll, and G. Stoltz, IMA Report No. 2040, 2005 (unpublished); http://www.ima.umn.edu/preprints/apr2005/2040.pdf

[3] M. Caffarel and P. Claverie, J. Chem. Phys. **88**, 1100 (1988).

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

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

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

[7] S. P. Meyn and R. L. Tweedie, *Markov Chains and Stochastic Stability* (Springer, Berlin, 1993).

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

[9] E. Cancès, B. Jourdain, and T. Lelièvre, Math. Models Meth. Appl. Sci. **16**, 1403 (2006).

[10] C. J. Umrigar, Phys. Rev. Lett. **71**, 408 (1993).

[11] Z. Sun, M. M. Soto, and W. A. Lester, Jr., J. Chem. Phys. **100**, 1278 (1994).

[12] D. Bressanini and P. J. Reynolds, J. Chem. Phys. **111**, 6180 (1999).

[13] B. Brünger, C. L. Brooks, and M. Karplus, Chem. Phys. Lett. **105**, 495 (1984).

[14] M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids* (Oxford University Press, Oxford, 1989) and references in Sec. 9.3.

[15] J. A. Izaguirre, D. P. Catarello, J. M. Wozniak, and R. D. Skeel, J. Chem. Phys. **114**, 2090 (2001).

[16] A. Ricci and G. Ciccotti, Mol. Phys. **101**, 1927 (2003).

[17] S. Chandrasekhar, Rev. Mod. Phys. **15**, 1 (1943).

[18] N. L. Stedman, W. M. C. Foulkes, and M. Nekovee, J. Chem. Phys. **109**, 2630 (1998).

[19] M. Caffarel, QMC=Chem is a quantum Monte Carlo program, IRSAMC, Université Paul Sabatier-CNRS, Toulouse, France.